A data-driven systematic, consistent and non-exhaustive approach to Model Selection

Diego Ribeiro Marcondes

Thesis presented to the
Institute of Mathematics and Statistics
of the University of São Paulo
in partial fulfillment
of the requirements
for the degree of
Doctor of Science

Program: Applied Mathematics
Advisor: Prof. Dr. Claudia Monteiro Peixoto

During the development of this work, the author received financial support from the National Council for Scientific and Technological Development (CNPq)

São Paulo
July, 2022

A data-driven systematic, consistent and non-exhaustive approach to Model Selection

Diego Ribeiro Marcondes

This version of the thesis includes the corrections and modifications suggested by the Examining Committee during the defense of the original version of the work, which took place on July 14, 2022.

A copy of the original version is available at the Institute of Mathematics and Statistics of the University of São Paulo.

Examining Committee:

Prof. Dr. Claudia Monteiro Peixoto (Advisor) – IME-USP
Prof. Dr. Junior Barrera – IME-USP
Prof. Dr. Claudio Landim – IMPA
Prof. Dr. Marcelo S. Reis – Unicamp
Prof. Dr. Ulisses M. Braga-Neto – Texas A&M University

The content of this work is published under the CC BY 4.0 license
(Creative Commons Attribution 4.0 International License)

Ficha catalográfica elaborada com dados inseridos pelo(a) autor(a)
Biblioteca Carlos Benjamin de Lyra
Instituto de Matemática e Estatística
Universidade de São Paulo

Marcondes, Diego
A data-driven systematic, consistent and non-exhaustive approach to Model Selection / Diego Marcondes; orientadora, Claudia Peixoto. - São Paulo, 2022.
176 p.: il.

Tese (Doutorado) - Programa de Pós-Graduação em Matemática Aplicada / Instituto de Matemática e Estatística / Universidade de São Paulo.
Bibliografia
Versão corrigida

1. Model Selection. 2. Statistical Learning. 3. U-curve algorithms. 4. VC theory. 5. PAC learning. I. Peixoto, Claudia. II. Título.

Bibliotecárias do Serviço de Informação e Biblioteca
Carlos Benjamin de Lyra do IME-USP, responsáveis pela
estrutura de catalogação da publicação de acordo com a AACR2:
Maria Lúcia Ribeiro CRB-8/2766; Stela do Nascimento Madruga CRB 8/7534.

Para a minha família

Agradecimentos

Se eu vi mais longe, foi por estar sobre ombros de gigantes
— Bernardo de Chartres

Primeiramente agradeço a Deus por dar saúde para mim e minha família, e possibilitar a realização deste trabalho.

Agradeço a toda a minha família por todo o apoio que sempre me deram, em especial aos meus pais Lucia e Nilton, meu irmão Thiago, minhas avós Elzi e Neide, e meus tios Angelina e Tobias.

Agradeço a minha orientadora Profa. Claudia por todo o apoio, desde a Graduação até este Doutorado, e pelo grande empenho e dedicação em me ajudar neste e em muitos outros trabalhos.

Agradeço ao Prof. Adilson pelas inúmeras discussões na pracinha do IME que me ajudaram a avançar o meu conhecimento. Ele e a Claudia muito me ensinaram nesses meus anos no IME, não só nos estudos e na pesquisa, mas para a vida.

Agradeço ao Prof. Junior, que me apresentou ao problema tratado nesta tese e cujas intuições foram o ponto de partida para desenvolver a teoria aqui apresentada.

Por fim, agradeço a Laryssa, a minha companheira para toda a vida, por todo o amor, apoio e companheirismo, e por ter me aturado e tido paciência comigo nos últimos anos.

Resumo

Diego Ribeiro Marcondes. **Uma abordagem sistemática, consistente e não-exaustiva para Seleção de Modelos baseada em dados**. Tese (Doutorado). Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2022.

A ciência moderna consiste em desenvolver um conjunto de hipóteses para explicar um fenômeno observável, confrontá-las com a realidade, e manter como possíveis explicações hipóteses que ainda não foram falsificadas. Esse conjunto de hipóteses é chamado de modelo, logo um passo importante do método científico é selecionar um modelo. Em métodos de Aprendizado Estatístico, isso consiste em selecionar um modelo dentre candidatos baseando-se em evidências quantitativas, e então aprender hipóteses nele pela minimização de uma função de risco empírica. A necessidade de selecionar um modelo, ao invés de considerar a união dos candidatos como as hipóteses possíveis, é a suscetibilidade a *overfitting*, a partir da qual emerge um *trade-off* entre complexidade e viés. Se escolhermos um modelo altamente complexo, então teremos nele hipóteses que explicam o fenômeno muito bem, mas também poderá haver hipóteses que explicam os dados empíricos muito bem, e não é claro como separamos essas hipóteses, logo ocorre *overfitting*. Se escolhermos um modelo mais simples, pode ocorrer que as hipóteses que se encaixam bem nos dados empíricos são as mesmas que melhor explicam o fenômeno, mas podem não explicá-lo muito bem, já que podem haver hipóteses que não estão no modelo que o explicam melhor, logo há um viés no aprendizado nesse modelo. Assim, escolher adequadamente o modelo é uma parte importante da solução de problemas de aprendizado, o que é feito por meio de Seleção de Modelos. Esta tese propõe uma abordagem baseada em dados sistemática, consistente e não-exaustiva para Seleção de Modelos. O principal conceito da abordagem são as coleções de modelos candidatos, que chamamos Espaços de Aprendizado, que, quando vistas como conjuntos parcialmente ordenados por inclusão, podem ter uma estrutura rica que aumenta a qualidade do aprendizado. A abordagem é baseada em dados, pois apenas o Espaço de Aprendizado e função de risco são escolhidas, e o restante da abordagem é baseado em dados. Ela é sistemática, pois é constituída de um sistema formal com dois passos: selecionar um modelo do Espaço de Aprendizado e aprender hipóteses nele. Do ponto de vista estatístico, há um modelo-alvo dentre os candidatos, que é aquele com menor viés e complexidade, e a abordagem é consistente, pois, quando o tamanho da amostra aumenta, o modelo selecionado converge para o modelo-alvo com probabilidade um, e os erros de estimação relacionados com o aprendizado de hipóteses nele convergem em probabilidade para zero. Desenvolvemos propriedades *U-curve* dos Espaços de Aprendizado que implicam a existência de algoritmos *U-curve* que podem estimar de forma ótima o modelo-alvo sem realizar uma busca exaustiva, e que podem também ser implementados eficientemente para obter soluções sub-ótimas. A principal implicação da abordagem são situações em que a falta de dados pode ser mitigada por alto poder computacional, uma propriedade que pode estar por trás dos métodos de aprendizado modernos de alta performance que demandam altos recursos computacionais. Ilustramos a abordagem em dados reais e simulados para aprender no importante Espaço de Aprendizado das Partições, para prever sequencias binárias geradas por cadeias de Markov, para aprender W-operadores multicamadas, e para filtrar imagens binárias através do aprendizado de funções Booleanas intervalares.

Palavras-chave: Aprendizado Estatístico. Seleção de Modelos. Algoritmos U-curve. Teoria VC. Aprendizado PAC. W-operadores. Reticulado das partições. Validação cruzada. Busca de arquiteturas de redes neurais.

Abstract

Diego Ribeiro Marcondes. **A data-driven systematic, consistent and non-exhaustive approach to Model Selection**. Thesis (Doctorate). Institute of Mathematics and Statistics, University of São Paulo, São Paulo, 2022.

Modern science consists on conceiving a set of hypotheses to explain observable phenomena, confronting them with reality, and keeping as possible explanations only hypotheses which have not yet been falsified. Such a set of hypotheses is called a model, hence an important step of the scientific method is to select a model. Under a Statistical Learning framework, this consists on selecting a model among candidates based on quantitative evidence, and then learning hypotheses on it by the minimization of an empirical risk function. The need to select a model, rather than considering the union of the candidates as the possible hypotheses, is the liability to *overfitting*, from which arises a complexity-bias trade-off. If we choose a highly complex model, then we may have in it hypotheses which explain the underlying process very well, but there may also be hypotheses which explain the empirical data very well, and it is not clear how to separate them, so we overfit the data. If we choose a simpler model, it may happen that the hypotheses which well fit empirical data are the same that better explain the process, but may not explain it very well, as there may be hypotheses not in the model which are better, so there is a bias when learning on this model. Therefore, properly choosing the model is an important part of the solution of a learning problem, and is performed via Model Selection. This thesis proposes a data-driven systematic, consistent and non-exhaustive approach to Model Selection. The main feature of the approach are the collections of candidate models, which we call Learning Spaces, that, when seen as a set partially ordered by inclusion, may have a rich structure which enhance the quality of learning. The approach is data-driven since the only features which are chosen are the Learning Space and risk function, so all other features are based on data. It is systematic since it is constituted of a formal system of two steps: select a model from the Learning Space and then learn hypotheses on it. From a statistical point of view, there is a target model among the candidates, which is that with the lowest bias and complexity, and the approach is consistent since, as the sample size increases, the selected model converges to the target with probability one, and the estimation errors related to the learning of hypotheses on it converge in probability to zero. We establish U-curve properties of the Learning Spaces which imply the existence of U-curve algorithms that can optimally estimate the target model without an exhaustive search, which can also be efficiently implemented to obtain suboptimal solutions. The main implication of the approach are instances in which the lack of data may be mitigated by high computational power, a property which may be behind the high-performance computing demanding modern learning methods. We illustrate the approach on simulated and real data to learn on the important Partition Lattice Learning Space, to forecast binary sequences under a Markov Chain framework, to learn multilayer W-operators, and to filter binary images via the learning of interval Boolean functions.

Keywords: Statistical Learning. Model Selection. U-curve algorithms. VC theory. PAC learning. W-operators. Partition lattice. Cross validation. Neural architecture search.

List of Abbreviations

ASFFS	Adaptative Floating Search
DNN	Deep neural network
ERM	Empirical Risk Minimization
GAMLSS	Generalized Additive Models for Location Scale and Shape
GGCP	Generalized Glivenko-Cantelli Problem
GLM	Generalized Linear Models
GPU	Graphical Processing Units
MDE	Maximum Discrimination Error
MNIST	Modified National Institute of Standards and Technology
PAC	Probably Approximately Correct
ROBDD	Reduced Ordered Binary Decision Diagram
SBS	Sequential Backward Selection
SFFS	Sequential Forward Floating Selection
SFS	Sequential Forward Selection
SRM	Structured Risk Minimization
SVM	Support Vector Machine
VC	Vapnik-Chervonenkis

List of Symbols

$\mathbb{Z}$	Integer numbers
$\mathbb{Z}_+$	Positive integer numbers
$\mathbb{R}$	Real numbers
$\mathbb{R}_+$	Positive real numbers
$\mathbb{1}\{\cdot\}$	Indicator function
ℓ	Loss function
d_{VC}	Vapnik-Chervonenkis dimension
Ω	Sample space of a probability space
$\mathcal{S}$	σ-algebra of a probability space
$\int_{\mathcal{Z}} f(z)\, dP(z)$	Lebesgue–Stieltjes integral of f in domain $\mathcal{Z}$ under distribution P
$\mathbb{P}$	Probability measure
$\mathbb{E}$	Expectation under probability measure $\mathbb{P}$
$\lvert A\rvert$	Cardinality of set A

List of Figures

1 A data set of 10 points in the plane obtained from an underlying process of observed phenomena. The dashed line represents the interpolation polynomial that completely explains the points, but does not really represent the pattern of the underlying process, which is better represented by a degree 2 polynomial (solid curve). This is a special case of a regression problem. 2

1.1 Classical framework of Machine Learning. 6

1.2 Example of a linear classifier learned by SVM. The points represent the training sample $\mathcal{D}_N$, and their shape is related to their observed value in the output variable. The yellow and orange regions are, respectively, above and below the learned classifier, and represent the points classified as 1 and 0. The wrongly classified points are in red, and the in-sample error of the estimated classifier is 0.11. 8

1.3 Solution to the Model Selection problem, in which $\hat{h}_i^{\mathcal{D}_N}$, $i = 1, \dots, n$, are the minimizers of $L_{\mathcal{D}_N}$ in each candidate model, and $\hat{L}$ is an estimator of the out-of-sample error, given by the expectation of the loss function under the validation sample $\mathcal{D} \setminus \mathcal{D}_N$. The model with the least validation error $\mathcal{M}_{m^\star}$ is chosen, and the hypotheses estimated is $\hat{h}_{m^\star}^{\mathcal{D}}$, the minimizer of $L_{\mathcal{D}}$ in $\mathcal{M}_{m^\star}$. See [1, Chapter 4] and [32] for more details of this framework. This diagram was adapted from [1] and [32]. 9

1.4 Learning framework via Learning Spaces. 11

1.5 Illustration of a DNN hypothesis, following architecture $\mathcal{A}$, as defined in (1.7). The rectangle's height is proportional to the dimension of the input variable of the respective layer, i.e., $d_1, \dots, d_m$, which can change from layer to layer. 19

1.6 Examples of (a) linear regression and polynomial fitting ((b) degree 2 and (c) degree 3). The points represent the training sample $\mathcal{D}_N$, and the curve is the minimizer of the mean quadratic error on the sample. These are the least square polynomials. 22

1.7 Examples of functions in $\mathcal{H}_1$ and $\mathcal{H}_2$, indexed by parameters p and p_1, p_2, respectively. On the one hand, given any three points $x_1 < x_2 < x_3$, the sequence $h(x_1), h(x_2), h(x_3)$ can change values at most once if $h \in \mathcal{H}_1$, and such a change occurs when $\min\{x_1, x_2, x_3\} < p < \max\{x_1, x_2, x_3\}$, while it can take any value in $\{0, 1\}^3$ when considering that $h \in \mathcal{H}_2$. On the other hand, given any four points $x_1 < x_2 < x_3 < x_4$, the sequence $h(x_1), h(x_2), h(x_3), h(x_4)$ can change values at most twice if $h \in \mathcal{H}_2$, and such changes occur when there exists $i \in \{2, 3\}$ such that $x_1 < p_1 < x_i < p_2 < x_4$. In this instance, the sequence $0, 1, 0, 1$ cannot be generated by a function in $\mathcal{H}_2$, since the value of $h(x_i)$ changes three times. Another way to see that it is not possible, is to note that the zeros and ones in this sequence should appear in clusters, and there should be at most two clusters of one value (zero or one), and at most one cluster of another. These clusters are represented by the regions $x < p_1, p_1 < x < p_2$ and $p_2 < x$. In the sequence $0, 1, 0, 1$ we have four clusters, two of each value, what is not possible. . . 24

1.8 Example of an indicator function $I(\cdot; h, \beta)$ in $\mathcal{G}_{\mathcal{H},\ell}$. 25

1.9 Example of binary function $I(x, y; h, \beta), x, y \in \mathbb{R}$, when $\mathcal{H} = \{h(x) = ax + b : a, b \in \mathbb{R}\}$ contains the linear functions of one variable, and $\ell((x, y), h) = [y - h(x)]^2$ is the quadratic loss function. The vertical lines represent a distance $\sqrt{\beta}$ from the respective point in the direction of $h(x)$, illustrating that points outside the pink region are at a distance greater than that from the line. 26

1.10 Examples of five point dichotomies which can be generated by functions in $\mathcal{G}_{\mathcal{H},\ell}$ defined in (1.10). The dashed lines are the lines parallel to h, but with a slope differing on $\pm\sqrt{\beta}$, hence the region between them represents the points classified as zero. Set $\mathcal{G}_{\mathcal{H},\ell}$ shatters these five points. 26

1.11 Examples of six point dichotomies which cannot be generated by functions in $\mathcal{G}_{\mathcal{H},\ell}$ defined in (1.10), when the points form (a) convex and (b) non-convex polygons, illustrating that $\mathcal{G}_{\mathcal{H},\ell}$ cannot shatter six points. 27

1.12 Parametric lattice for variable selection when $d = 4$. 33

1.13 Partition Lattice for Linear Classifiers with $d = 4$ or for $\mathcal{X} = \{1, 2, 3, 4\}$. The tables present the hypotheses in selected models $\mathcal{M}_1, \mathcal{M}_2$ of the Partition Lattice Learning Space for $\mathcal{X} = \{1, 2, 3, 4\}$. The orange nodes represent the Boolean lattice of variable selection when $\mathcal{X} = \{0, 1\}^2$, so its points are $1 = (0, 0), 2 = (0, 1), 3 = (1, 0)$ and $4 = (1, 1)$. The dashed nodes are the ones in $\mathbb{L}(\mathcal{H}) \cap \mathcal{M}$, in which $\mathcal{M}$ is composed by the non-decreasing hypotheses. We present an example of joint empirical frequencies observed in a training and validation sample. The number in each node represents its estimated error calculated as (1.11), by first estimating a hypothesis via ERM with the training sample, and then calculating its error on the validation sample. The bold hypotheses in each table represent the ERM hypothesis of the respective model. When there is more than one ERM hypothesis in a model, we consider the minimum validation error among them as its estimated error. 37

1.14 Examples of a continuous chain (orange) and a chain that is not continuous (blue) within a Boolean Learning Space. Observe that there is no model in the Learning Space between two subsequent models of the orange chain, while in the blue chain there are two models between the second and third (from bottom to top) models of it. 40

1.15 Example of a (a) strong local minimum, that is a local minimum of all chains which contain it, and (b) sup-strong local minimum, that is a model with error lesser or equal to all models at a distance one from it that are greater. Observe that (b) is also a weak local minimum of four chains that pass through it. 41

1.16 Decomposition of $\mathcal{H}$ by a $\mathbb{L}(\mathcal{H})$. We omitted some models for a better visualization, since $\mathbb{L}(\mathcal{H})$ should cover $\mathcal{H}$. 42

1.17 Types II, III, and IV estimation errors when learning on $\hat{\mathcal{M}}$, in which $\hat{h}_{\hat{\mathcal{M}}} \equiv \hat{h}^{\mathrm{A}}_{\hat{\mathcal{M}}}$. 43

1.18 Type IV estimation error and type II estimation error of learning on $\mathcal{H}$ via ERM with sample $\mathcal{D}_N$. 44

1.19 Two frameworks for learning hypotheses via Learning Spaces. (a) A sample of size $N + M$ is split into two, one of size N that is used to estimate $\hat{\mathcal{M}}$ by minimization of $\hat{L}$ on $\mathbb{L}(\mathcal{H})$, and another of size M used to learn a hypothesis on $\hat{\mathcal{M}}$ by ERM. (b) The whole sample of size N is used for estimating $\hat{\mathcal{M}}$ by the minimization of $\hat{L}$ on $\mathbb{L}(\mathcal{H})$, and to estimate hypotheses on $\hat{\mathcal{M}}$ via ERM. 47

2.1 The errors of the equivalence classes (cf. (1.15)) of $\mathbb{L}(\mathcal{H})$ in ascending order. The MDE $\epsilon^\star$ is the difference between the error of the target class $\mathcal{M}^\star$, and the second to best $\mathcal{M}_2$. The colored intervals represent a distance of $\epsilon^\star/2$ from the real error of each model, and the colored estimated errors $\hat{L}$ illustrate a case such that the estimated error is within $\epsilon^\star/2$ of the real error for all models. The class $\mathcal{M}_1$ has the same error as $\mathcal{M}^\star$, but has a smaller estimated error, and, by the definition of $\mathcal{M}^\star$, greater VC dimension. Note from the representation that, if one can estimate $\hat{L}$ within a margin of error of $\epsilon^\star/2$, then $\hat{\mathcal{M}}$ will be a model with the same error as $\mathcal{M}^\star$, in this case $\mathcal{M}_1$ (cf. Proposition 2.6). 53

2.2 Sample size N needed to have bounds (2.24) and (2.25) equal to 0.05 as a function of $d_{VC}(\mathcal{H})$, for distinct values of $\epsilon^\star$ (columns) and ϵ (lines), and $c = 0.2$. The curves of type II bound (2.24) are in red, and the ones of type IV bound (2.25) are in green. When the red curve is below the green one, we have a tighter bound for type II estimation error when learning directly on $\mathcal{H}$ with a sample of size $2N$, while when the green curve is below the red one, we have a tighter bound for type IV estimation error when learning with independent sample on $\mathbb{L}_2(\mathcal{H})$, with a training sample of size $0.8N$, a validation sample of size $0.2N$, and an independent sample of size N. To aid in the visualization, we painted the space between the two curves in green when the bound of type IV estimation error (2.25) is tighter, and in red when the bound of type II estimation error (2.24) is tighter. 66

2.3 Sample size N needed to have bounds (2.24) and (2.26) equal to 0.05 as a function of $d_{VC}(\mathcal{H})$, for distinct values of $\epsilon^\star$ (columns) and ϵ (lines), and $c = 0.2$. The curves of type II bound (2.24) are in red, and the ones of type IV bound (2.26) are in green. When the red curve is below the green one, we have a tighter bound for type II estimation error when learning directly on $\mathcal{H}$ with a sample of size $2N$, while when the green curve is below the red one, we have a tighter bound for type IV estimation error when learning with independent sample on $\mathbb{L}(\mathcal{H})$, the Partition Lattice Learning Space, with a training sample of size $0.8N$, a validation sample of size $0.2N$ and an independent sample of size N. To aid in the visualization, we painted the space between the two curves in green when the bound of type IV estimation error (2.26) is tighter, and in red when the bound of type II estimation error (2.24) is tighter. 67

3.1 Illustration of the U-curve phenomenon, instantiated to a chain of nested models with increasing complexity. This is the typical behavior of $\hat{L}$ on continuous chains of a Learning Space that satisfies the strong U-curve property (cf. Definition 3.1). 82
3.2 Example of a lattice satisfying the weak U-curve property. The number inside each node $\mathcal{M}$ is $\hat{L}(\mathcal{M})$. The strong local minimums are in green, the weak local minimums are in orange, the inf-strong local minimums are dashed and the sup-weak local minimums are dotted. All strong local minimums are global minimums of all continuous chains which contain them, so this is an example of a weak U-curve property configuration. The inclusion relation $\subset$ is from the bottom to the top. 85
3.3 Illustration of (a) strong and (b) sup-strong local minimums. 85
3.4 A Learning Space isomorphic to a Boolean lattice, so it is U-curve compatible. The orange nodes represent the lattice $C^-(\mathcal{M})$, and the blue nodes the lattice $C^+(\mathcal{M})$, for a given $\mathcal{M}$. The orange dashed nodes are in $N^-(\mathcal{M}_i)$, and the blue dashed nodes are in $N^+(\mathcal{M}_j)$. The green nodes are an example of a pair $\mathcal{M}_1, \mathcal{M}_2$ for which the condition (3.11) of Theorem 3.4 should be satisfied. 90

4.1 Percentage of the simulations in which the error of the ERM hypothesis in $\mathcal{H}$ were lesser, equal, or greater than the error of the hypothesis learned on $\hat{\mathcal{M}}$, i.e., via Learning Spaces, with the independent sample for each example, sample size and algorithm. 104
4.2 (A) Daily bitcoin value in US Dollars from April 30th 2013 to April 6th 2022, which are the days considered in the training, validation and test samples according to the colors. (B) The balance of two accounts which started with $1,000$ US Dollars of bitcoin in February 1st 2021, and which followed, respectively, the strategy of staying on the market every day (red), and staying on the market only on days in which the learned hypothesis, for the respective value of d, predicts as positive days (green). 117
4.3 Matrix representation of black and white handwritten digits in the MNIST data set [89]. The gray pixels (value greater than zero) were considered as black (value one). The zero values are omitted for a better visualization. . 118

4.4 Example of a W-operator filter ψ which recognizes the boundary of a digit. The window W is a subset of $\{0,1\}^{5\times5}$ and the W-operator equals zero if all considered neighbors of a pixel are equal, and one otherwise. The window W is centered at every possible pixel of $x \in \mathcal{X}$, that are all but the ones at the first and last two rows and columns, and the W-operator is calculated for this pixel. Going through every pixel of the image, results in the image on the right-hand side. The zero values are omitted for a better visualization. 120
4.5 The windows of the multilayer W-operator estimated to predict the zero digit in the MNIST data set. 128
4.6 Black and white images of size 300×300 of dogs, such that $x \leq w \leq y$. Image x has $49,044$, image w has $65,774$ and image y has $75,427$ black pixels. Hence, there are $2^{26,383}$ images in $[x, y]$. 129

5.1 The approximation error of hypotheses spaces. 140

List of Tables

3.1 First to 30th Bell number. 86

3.2 Counterexample of empirical training and validation joint frequencies under which a strong local minimum of the Partition Lattice Learning is not a global minimum of all chains that pass through it. The strong local minimum is $\pi = \{\{1\}, \{2,4\}, \{3,5\}, \{6\}\}$, with $\hat{L}(\mathcal{H}|_\pi) = 0.25$, but $\hat{L}(\mathcal{H}|_{\pi'}) = 0.2$ with $\pi' = \{\{1,2,3,4,5,6\}\}$, and $\pi' \leq \pi$. There are more strong local minimums which are not global minimums under these empirical joint frequencies. 93

3.3 Empirical distributions of a training and validation samples when $\mathcal{X} = \{a_1, a_2, a_3, b\}$ is a set with four points, and estimated hypothesis for partitions $\pi_1 \wedge \pi_2 = \{\{a_1, a_2, a_3\}, b\}$, $\pi_1 = \{\{a_1, a_2\}, a_3, b\}$, $\pi_2 = \{\{a_1, a_3\}, a_2, b\}$ and $\pi_1 \vee \pi_2 = \{a_1, a_2, a_3, b\}$. 94

4.1 Joint distributions considered in each example. All of them have a same $L(h^\star)$, but are, from Example 1 to 4, of increasing Conditional Entropy and $\epsilon^\star$, except for Example 4 which has the same $\epsilon^\star$ as Example 2. 102

4.2 The percentage of simulated samples in which the hypothesis returned by the suboptimal algorithm was better, worse and as good as the hypothesis returned by the optimal algorithm, for each example and sample size. When more than one hypothesis is returned, we consider the real error of the hypothesis with the least error when comparing the algorithms. 105

4.3 Results of the simulations for each example, sample size and type of algorithm (optimal or suboptimal). We present the number of models exhausted; the estimated error $\hat{L}(\hat{\mathcal{M}})$ of $\hat{\mathcal{M}}$; the real error $L(\hat{\mathcal{M}})$ of $\hat{\mathcal{M}}$; the real error $L(\hat{h})$ of $\hat{h}$, the ERM hypothesis in $\mathcal{H}$ of the whole sample (union of training, validation and independent sample); the real error $L(\hat{h}^{\tilde{D}_M}_{\hat{\mathcal{M}}})$ of the hypothesis estimated from $\hat{\mathcal{M}}$ with the independent sample; and the execution time of the algorithm in minutes. For each quantity, we present the median, and within parentheses the percentiles 2.5% and 97.5%, of the 100 samples. . . 107

4.4 (A) Conditional distribution of an order 3 Markov chain, which actually represents a variable order Markov chain with contexts and conditional probabilities in (B). (C) Represents the context tree of the variable order Markov chain with conditional distribution (B). 109
4.5 Results of the models estimated via Algorithm 5 to forecast the variation of bitcoin. For each model, it is presented the maximum VC dimension d considered, the time in minutes it took to run Algorithm 5, the maximum order of its contexts, the VC dimension of $\hat{\mathcal{M}}$, the classification error on the validation and test sample, and the minimum, maximum and final spread obtained in the test period by applying the strategy based on the learned classifier. The training, validation and test sample sizes of all models are, respectively, 2774, 31 and 430. 116
4.6 Estimated hypothesis for $d = 8$. 118
4.7 Confusion matrix of the multilayer W-operator learned to predict the zero digit in the MNIST data set. The test error is 0.0219. 127

Contents

Introduction 1

1 Model Selection via Learning Spaces 5

1.1 Motivation: Model Selection in Machine Learning . . . 6

1.1.1 Classical Machine Learning framework . . . 6

1.1.2 Model Selection in Machine Learning . . . 8

1.1.3 Model Selection via Learning Spaces . . . 11

1.1.4 Estimation errors under Model Selection . . . 12

1.1.5 Computational aspects of Model Selection via Learning Spaces . 13

1.2 Framework for the learning of hypotheses . . . 14

1.2.1 Hypotheses spaces and loss functions . . . 14

1.2.2 Examples of hypotheses spaces and loss functions . . . 16

1.2.3 Target hypotheses . . . 19

1.2.4 VC dimension . . . 22

1.2.5 Model error estimation . . . 27

1.3 Learning Spaces . . . 29

1.3.1 Building Learning Spaces . . . 31

1.3.2 Examples of Learning Spaces . . . 32

1.3.3 Minimums of Learning Spaces . . . 40

1.4 Target model and main objective . . . 41

1.5 The learning of hypotheses via Learning Spaces . . . 45

1.5.1 Learning model $\hat{\mathcal{M}}$ . . . 45

1.5.2 Learning hypotheses on $\hat{\mathcal{M}}$ . . . 46

1.6 Next steps . . . 48

2 Consistency of Model Selection via Learning Spaces 49

2.1 VC theory and PAC-learnability . . . 50

2.2 Convergence to the target model . . . 52

2.3 Convergence of estimation errors on $\hat{\mathcal{M}}$ . . . 59

2.3.1 Learning with independent sample . . . 59
2.3.2 Learning by reusing . . . 68
2.4 Unbounded loss functions . . . 70
2.4.1 Convergence to the target model . . . 72
2.4.2 Convergence of estimation errors on $\hat{\mathcal{M}}$ . . . 76
2.4.3 Learning with independent sample . . . 76
2.4.4 Learning by reusing . . . 79
2.5 Next steps . . . 80

3 U-curve: properties and algorithms 81
3.1 Occam's razor and peaking phenomenon are facets of U-curve . . . 81
3.2 U-curve properties . . . 83
3.3 U-curve on the Partition Lattice Learning Space . . . 86
3.4 Sufficient condition for the weak U-curve property . . . 89
3.5 A generic U-curve algorithm . . . 94
3.6 Improving the U-curve algorithm . . . 97
3.7 Next steps . . . 99

4 Applications 101
4.1 Learning via the Partition Lattice Learning Space . . . 102
4.2 Forecasting variable order Markov chains . . . 108
4.2.1 Main ideas and definitions . . . 108
4.2.2 Suboptimal algorithm . . . 111
4.2.3 Investment strategy for bitcoin . . . 114
4.3 Multilayer W-operator . . . 117
4.3.1 Main ideas . . . 118
4.3.2 Notation and definitions . . . 122
4.3.3 MNIST results . . . 126
4.4 Interval Boolean functions . . . 127
4.4.1 Main ideas . . . 127
4.4.2 Notation and definitions . . . 131
4.4.3 U-curve property . . . 133

5 Discussion 135
5.1 Main results and implications . . . 135
5.2 Learning Spaces and penalized loss functions . . . 138
5.3 Decreasing the approximation error . . . 139
5.4 Perspectives in neural networks . . . 140

5.5 Limitations . . . 141
5.6 Topics for future researches . . . 141

Appendixes

A Vapnik-Chervonenkis theory **143**
A.1 Generalized Glivenko-Cantelli Problems . . . 143
A.2 Convergence to zero of type I estimation error . . . 145
A.2.1 Binary loss functions . . . 145
A.2.2 Bounded loss functions . . . 150
A.2.3 Unbounded loss functions . . . 151
A.3 Convergence to zero of type II estimation error . . . 158
A.4 Finite VC dimension is sufficient and necessary for consistency . . . 160

B Useful Mathematical concepts **161**
B.1 Lattice theory . . . 161
B.2 Directed acyclic graph . . . 162
B.3 Hoeffding's Inequality . . . 163
B.4 Borel-Cantelli Lemma . . . 163

References **165**

Introduction

Modern science consists on conceiving a set of hypotheses to explain observable phenomena, confronting them with reality, and keeping as possible explanations only hypotheses which have not yet been falsified [119]. Such a set of hypotheses is called a model, hence an important step of the scientific method is to select a model, to only then confront its hypotheses with reality and keep only hypotheses not falsified as possible explanations to phenomena.

A model may be built by reasoning from observation, or may be selected from a collection of candidate models[1]. In this thesis, we are concerned with the latter, which, under a Statistical Learning framework [149, 150], consists on selecting a model among candidates based on quantitative empirical evidence, and then *learning*[2] hypotheses on it by the minimization, over the model, of a risk function based on empirical data. The *risk* of a hypothesis is a measure of how much it *explains* the phenomena characterized by the empirical data, and the hypotheses which minimize it are the *best* hypotheses in the model.

The need to select a model among candidates, rather than considering the union of the candidate models as the possible hypotheses, is the liability to *overfitting* [114], which occurs when the learned hypotheses explain very well the available empirical data, but fail to explain new data characterizing the phenomena. As an elementary example, assume the empirical data is formed by points in the plane, and the hypotheses are curves in it that seek to explain the *pattern* of the points, what is a special case of a regression problem. A possible collection of candidate models are polynomials with a certain degree, varying from zero to the data set size N minus one, which is a collection of N candidate models.

If one considered as model the union of the candidates, that is formed by all polynomials with degree at most $N - 1$, and minimized over this model a risk function related to the distance from the observed data points to each hypothesis, he would learn a hypothesis that interpolates the data points. This is the dashed curve in Figure 1, which consists of a data set with 10 points. Although this hypothesis completely explains the observed points, it is hardly an explanation for the pattern of the data, which is much more reasonable

[1] In this instance, reasoning is employed to choose the candidate models.

[2] Learning hypotheses in a model has the same meaning as selecting hypotheses, or estimating hypotheses, from a model, and is the usual expression employed in Statistical Learning. The learned hypotheses are to be understood as the ones which have not been *falsified* by the empirical evidence, and not as hypotheses which *for sure* explain the phenomena, what would be outside the scientific method.

to be explained by a degree 2 polynomial, represented by the solid curve in Figure 1. Hence, the model must be carefully selected, since choosing an arbitrarily complex model does not necessarily lead to a better learning of hypotheses, which is measured as how good the learned hypotheses perform on new data from the same process. Complexity in this regression example is understood as the degree of the polynomials, so the degree 2 polynomial that better represents the data pattern is in a model less complex than that of the interpolating polynomial.

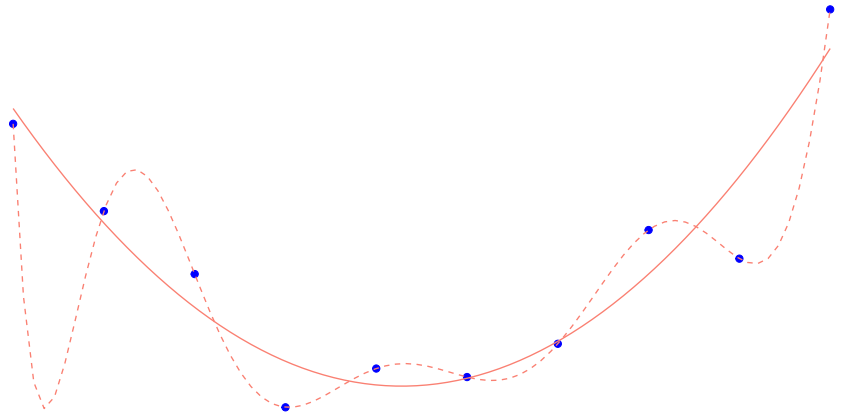

Figure 1: *A data set of* 10 *points in the plane obtained from an underlying process of observed phenomena. The dashed line represents the interpolation polynomial that completely explains the points, but does not really represent the pattern of the underlying process, which is better represented by a degree* 2 *polynomial (solid curve). This is a special case of a regression problem.*

From the overfitting arises a complexity-bias trade-off when selecting a model. On the one hand, if we choose a highly complex model, then we may have in it hypotheses which explain the underlying process very well, but there may also be hypotheses which explain the empirical data very well, and it is not clear how to separate these two kinds of hypotheses when they do not coincide, so we overfit the data. On the other hand, if we choose a simpler model, it may happen that the hypotheses which well fit empirical data are the same, in the model, that *better* explain the process, but may not explain it very well, as there may be hypotheses not in the model which are much better, so even though we avoid overfitting, there is a bias when learning on this model, also called underfitting. Therefore, properly choosing the model is an important part of the solution of a learning problem, and one manner of performing it is by selecting a model from candidates based on quantitative empirical data, in what is called Model Selection.

This thesis proposes a data-driven systematic, consistent and non-exhaustive approach to Model Selection. The main feature of the approach are the collections of candidate models, which we call Learning Spaces, that are collections of subsets of a model $\mathcal{H}$, called hypotheses space, which contains all hypotheses one is willing to consider as explanation for the phenomena. These collections, denoted by $\mathbb{L}(\mathcal{H})$, cover $\mathcal{H}$, i.e., the union of the subsets in $\mathbb{L}(\mathcal{H})$ equal $\mathcal{H}$, and, when seen as partially ordered by inclusion, may have a rich structure which enhance the quality of learning by first selecting a model in it, and then learning hypotheses on such a model. Another feature of the approach is that, due to

the structure of $\mathbb{L}(\mathcal{H})$, a model may be properly selected without exhaustively searching it, what is usually a bottleneck of Model Selection, which prevents considering a high number of candidate models.

The main characteristics of the approach are:

- **Data-driven**: the only features of the learning process which are chosen *a priori* are the candidate models $\mathbb{L}(\mathcal{H})$ and the risk function to be minimized, so all other features are based on data, without the need for any assumption about the distribution which generated it, so it is a distribution-free framework. The approach also does not depend on hyper-parameters to regulate the search on $\mathbb{L}(\mathcal{H})$, or any kind of penalization to the risk function, what is common in Model Selection methods [99].
- **Systematic**: from properties of $\mathbb{L}(\mathcal{H})$, we conceive a systematic method to learn hypotheses, which is constituted of two steps: select a model from $\mathbb{L}(\mathcal{H})$ and then learn hypotheses on it. The first step is a combinatorial problem in which we choose a model based on data, where we then solve a combinatorial or continuous optimization problem to learn hypotheses. This two-step systematic approach differs from some classical learning frameworks, in which either the model is fixed a priori, or the selection of a model is not completely data-driven and systematic, but rather a heuristic or dependent on hyper-parameters and penalization [1, Chapter 4].
- **Consistent**: in the proposed approach there is a *target* model, which is that among the candidates with the lowest bias and complexity, so the approach is consistent in the sense that, as the sample size increases, the selected model converges to the target with probability one, and the estimation errors related to the learning of hypotheses on it converge in probability to zero. This is an extension of PAC-learnability [148] to the learning of hypotheses via Model Selection, guaranteeing that, if the sample size is great enough, then, with high probability, one will be learning on the *best* model there is to learn in $\mathbb{L}(\mathcal{H})$, with low estimation error.
- **Non-exhaustive**: we establish properties of $\mathbb{L}(\mathcal{H})$ which imply the existence of an algorithm that can optimally estimate the target model without exhaustively searching $\mathbb{L}(\mathcal{H})$, whose cardinality may increase exponentially with the number of parameters that represent the hypotheses. This algorithm solves a U-curve problem [8, 32, 131, 133] and, although is NP-hard [128], can be applied to solve real problems [130]. Moreover, the method may not only be employed to obtain optimal, but also suitable suboptimal solutions to the Model Selection problem, which can be computed efficiently.

Model Selection via Learning Spaces is a systematic method based on empirical data to select a model with nice asymptotic properties, that can be applied to solve real problems, having the characteristics one expects a learning approach to have: a solid theoretical foundation with applicability. The main implication of the method, which is thoroughly analyzed in this thesis, is that the lack of data may be mitigated by high computational power. We expect with this work to present an instance where this principle, which we believe may be behind modern learning techniques, such as deep neural networks, holds from both a theoretical and empirical perspective.

We start Chapter 1 with a motivation of the proposed approach, concerning Model

Selection in Machine Learning, which is followed by a presentation of the main concepts and notation related to learning hypotheses in a Statistical Learning framework. Then, we define the Learning Spaces and present examples of them, some of which are already employed to Model Selection in the literature. At this point, we have the tools necessary to formally define the main objective of this thesis, that is presented in Section 1.4, and revolves around the important concept of target model. To end the chapter, we finally define in Section 1.5 the systematic data-driven approach to Model Selection, the object of this thesis.

In Chapter 2, we show that the proposed approach is consistent. We start by defining consistency, and then treat the case of bounded loss (risk) functions, by showing first that the selected model converges with probability one to the target model, and then that the estimation errors of learning converge in probability to zero. In order to do so, we employ classical tools of Vapnik-Chervonenkis theory, which we recall in Appendix A. We end this chapter studying the consistency for unbounded loss functions, which requires novel methods and technical results.

The non-exhaustiveness of the method is established in Chapter 3. We start by defining the U-curve properties, which are a formalization of heuristics in Model Selection related to Occam's razor, the peaking phenomenon and/or the curse of dimensionality, and then show that such a property is satisfied on a specific Learning Space that is suitable to solve many learning problems. Next, we present a sufficient condition for a U-curve property that draws a parallel with convexity, culminating on the definition of lattice convexity, when poset $(\mathbb{L}(\mathcal{H}), \subset)$ has a lattice structure, which is formally defined in Appendix B. From the U-curve properties, we derive generic non-exhaustive U-curve algorithms that search $\mathbb{L}(\mathcal{H})$ to obtain optimal solutions, when a U-curve property is satisfied, or suitable suboptimal when this is not the case. We outline that a detailed study of U-curve algorithms and their implementation is out of the scope of this thesis and is left as a topic for future research.

After attesting that the systematic data-driven approach for Model Selection is consistent and non-exhaustive, we illustrate its features with applications in Chapter 4. We start by learning with simulated data in a particular Learning Space to illustrate its theoretical properties regarding consistency, and optimality of U-curve algorithms. Then, we instantiate the method to forecast sequences of binary values which are generated by variable order Markov chains, and apply it to obtain a successful investment strategy for bitcoin. The third application is an attempt of defining a discrete neural network given by the composition of W-operators [15], which we call multilayer W-operator, that is applied to recognize the handwritten digit zero in the MNIST data set [89]. We end this chapter defining a Learning Space specially suitable for filtering binary images, and showing that a U-curve property is satisfied by it.

In Chapter 5, we discuss the main results, implications, and limitations of this thesis. In special, we discuss how penalized Model Selection methods are an important special case of our approach, although are not treated in this thesis, and discuss the perspectives of the approach regarding neural networks. Throughout the thesis, we present a myriad of topics for future researches, which are condensed in the last section of the discussion. We end the thesis with one final remark.

Chapter 1

Model Selection via Learning Spaces

The first step of Model Selection is to fix *all possible models* one is willing to consider as an explanation to a phenomenon. A model is a set of hypotheses about reality which may or may not be satisfied, to a certain degree, by the phenomenon. We denote these models, which are not necessarily disjoint sets, by $\mathcal{M}_1, \dots, \mathcal{M}_n$, $n \geq 1$, and its hypotheses by $h \in \mathcal{M}_i$. We assume there is a function $\mathscr{C}$, from the space of models to $\mathbb{Z}_+$, which represents the *complexity*, in some sense, of each model, so two models $\mathcal{M}_i, \mathcal{M}_j$ are as complex if

$$\mathscr{C}(\mathcal{M}_i) = \mathscr{C}(\mathcal{M}_j),$$

and $\mathscr{C}^{-1}(\mathbb{Z}_+)$ is a partition of $\{\mathcal{M}_1, \dots, \mathcal{M}_n\}$, with each block containing the models with a same complexity.

Such a sequence of models generates a hypotheses space $\mathcal{H}$, which contains all hypotheses one is willing to consider:

$$\mathcal{H} = \bigcup_{i=1}^{n} \mathcal{M}_i.$$

Although a hypotheses space $\mathcal{H}$ may be viewed as a union of models, one could also depart from $\mathcal{H}$, and then choose a collection of models

$$\{\mathcal{M}_1, \dots, \mathcal{M}_n\} \subset \mathcal{P}(\mathcal{H}), \tag{1.1}$$

in the powerset of $\mathcal{H}$, that covers[1] $\mathcal{H}$, on which to search for a suitable model to express reality. There are as many collections (1.1) as the number of covers of[2] $\mathcal{H}$, and one of the main problems in Model Selection is to properly choose one.

[1] It is also common for one to choose a collection which do not cover $\mathcal{H}$. In this case, the set of hypotheses would be a subset of $\mathcal{H}$, so we regard this as a special case of the abstract theory, in which the effective hypotheses space is the subset of $\mathcal{H}$. See [1, Section 4.3] for an example of such an approach.

[2] See [95] for this number when $\mathcal{H}$ is a set with finite cardinality.

After a collection is chosen, one then applies a procedure to select a model from it and, among the hypotheses in such model, select the one that *best* express reality. In this framework, the selection, also called learning, of hypotheses is performed in two, not necessarily disjoint, steps: select a model from a collection of models, and learn hypotheses on such model.

The main objective of this chapter is to develop a data-driven systematic approach to Model Selection that will be showed to be consistent and non-exhaustive in later chapters. In order to achieve this task, we define the Learning Spaces of a hypotheses space $\mathcal{H}$ as collections of candidate models. The main feature of such approach is that both the selection of a model and the learning of hypotheses are performed, once the Learning Space is chosen, systematically based solely on data.

In Section 1.2, we present the main elements related to hypotheses spaces and the learning of hypotheses from data. In Section 1.3, we define the Learning Spaces $\mathbb{L}(\mathcal{H})$ and introduce some examples, while in Section 1.4, we formally present the main objectives of this thesis. In Section 1.5, we present an approach to Model Selection based on a Learning Space. But first, as a motivation, we present the Model Selection problem, and the main characteristics of its solution via Learning Spaces, from the perspective of Machine Learning problems.

1.1 Motivation: Model Selection in Machine Learning

1.1.1 Classical Machine Learning framework

The classical framework of Machine Learning is a triple $(\mathcal{H}, \mathbb{A}, \mathcal{D}_N)$, composed by a set $\mathcal{H}$ of hypotheses h, which are functions from $\mathcal{X} \subset \mathbb{R}^d$, $d \geq 1$, to $\mathcal{Y} \subset \mathbb{R}$, called hypotheses space, and a learning algorithm $\mathbb{A}(\mathcal{H}, \mathcal{D}_N)$, which searches $\mathcal{H}$ seeking to minimize an error measure that assesses how good each $h \in \mathcal{H}$ predicts the values of Y from instances of X. This error is based on a training sample $\mathcal{D}_N = \{(X_1, Y_1), \ldots, (X_N, Y_N)\}$ of a random vector (X, Y), with range $\mathcal{X} \times \mathcal{Y}$ and unknown joint probability distribution P. See Figure 1.1 for an illustration of this framework.

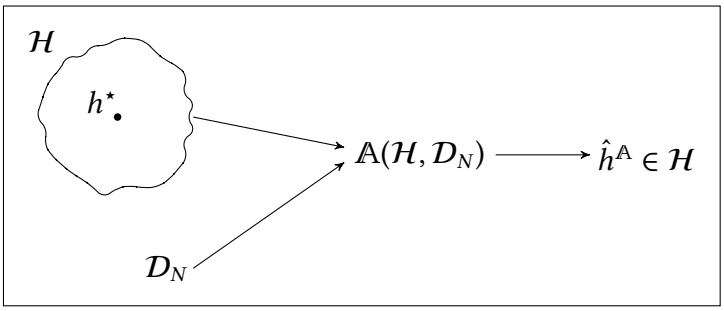

Figure 1.1: *Classical framework of Machine Learning.*

Let $\ell : \mathcal{Y} \times \mathcal{Y} \mapsto \mathbb{R}_+$ be a loss function. The error, or risk, of a hypothesis $h \in \mathcal{H}$ is an expected value of the local measures $\ell(h(x), y), (x, y) \in \mathcal{X} \times \mathcal{Y}$. If the expectation is

the sample mean of $\ell(h(x), y)$ under $\mathcal{D}_N$, we have the in-sample error $L_{\mathcal{D}_N}(h)$, while if the expectation of $\ell(h(X), Y)$ is under the joint distribution P, we then have the out-of-sample error $L(h)$.

Common loss functions are the simple loss function $\ell(y_1, y_2) = \mathbb{1}\{y_1 \neq y_2\}$, when $\mathcal{Y}$ has a finite number of elements, what characterizes a classification problem, or the quadratic loss function $\ell(y_1, y_2) = (y_1 - y_2)^2$, when $\mathcal{Y}$ has infinite elements, what characterizes a regression problem. In this context, a target hypothesis $h^\star \in \mathcal{H}$ is such that its out-of-sample error is minimum in $\mathcal{H}$, i.e., $L(h^\star) \leq L(h), \forall h \in \mathcal{H}$, while an Empirical Risk Minimization (ERM) hypothesis $\hat{h}$ is such that its in-sample error is minimum, i.e., $L_{\mathcal{D}_N}(\hat{h}) \leq L_{\mathcal{D}_N}(h), \forall h \in \mathcal{H}$.

The algorithm $\mathbb{A}$ returns a $\hat{h}^{\mathbb{A}} \in \mathcal{H}$ seeking to approximate a target hypothesis $h^\star \in \mathcal{H}$. The returned hypothesis can be, for example, an ERM hypothesis, but this is not necessary. In any case, whatever is the algorithm $\mathbb{A}$, such learning framework has an important parameter that is problem-specific: the hypotheses space $\mathcal{H}$, which has a strong impact on the generalization quality of the estimated hypothesis $\hat{h}^{\mathbb{A}}$, that is characterized by a small out-of-sample error.

The fundamental result in Machine Learning is the Vapnik-Chervonenkis (VC) theory [149, 150, 151, 152, 153, 154], which implies that a hypotheses space $\mathcal{H}$ is PAC-learnable [148] if, and only if, it has finite VC dimension ($d_{VC}(\mathcal{H}) < \infty$) [139, Theorem 6.7]. This means that, for any data generating joint distribution P, $L(\hat{h}^{\mathbb{A}})$ is close to $L(h^\star)$ with great confidence, if N is sufficiently large. Therefore, it is possible to learn hypotheses with a finite sample, with precision and confidence dependent on the training sample size N and the VC dimension (complexity) of $\mathcal{H}$.

Concrete example: linear classifiers

In order to aid the understanding of this section by the non-initiated reader, we will exemplify some concepts discussed here using the linear classifiers, as follows. Consider the hypotheses spaces given by

$$\mathcal{H} = \left\{ h_a(x_1, x_2) = \mathbb{1}\{a_0 + a_1 x_1 + a_2 x_2 >= 0\} \ : \ a = (a_1, a_1, a_2) \in \mathbb{R}^3 \right\},$$

that is formed by the functions from $\mathbb{R}^2$ to $\{0, 1\}$ represented by lines with parameters a_0, a_1, a_2, which classify the points above and below the line in one and zero, respectively. This is the hypotheses space of the dimension two linear classifiers, also known as perceptrons, and was one of the first hypotheses spaces employed to solve classification problems [135].

Considering the simple loss function, the empirical error $L_{\mathcal{D}_N}(h_a)$ represents the proportion of the sample points $\mathcal{D}_N$ that is miss-classified by h_a, while the out-of-sample error $L(h_a)$ represents the expected proportion of the points miss-classified by h_a according to the data generating distribution P. The ERM lines are those with minimum empirical classification error, and there will be usually infinite such lines. This can be seen in Figure 1.2, where slightly changing the value of the intercept (a_0) of the displayed line does not change its empirical error, hence the line with changed intercept is also an ERM hypothesis.

Another manner of having a suitable and more computing-efficient solution is, instead of considering the ERM hypotheses $\hat{h}$, consider the hypotheses $\hat{h}^{\mathbb{A}}$ returned by the Support Vector Machine (SVM) algorithm [37], which are minimizers over $\mathbb{R}^3$ of the following empirical error:

$$L_{\mathcal{D}_N}(a) = \lambda \left(a_0^2 + a_1^2 + a_3^2\right) + \left[\frac{1}{N}\sum_{i=1}^{N} \max\left(0, 1 - Y_i(a_0 + a_1 x_1 + a_2 x_2)\right)\right], \qquad a \in \mathbb{R}^3, \tag{1.2}$$

in which $\lambda > 0$ is a free hyper-parameter which has to be fixed. Since (1.2) can be written as a convex optimization problem, it can be computed more efficiently than minimizing the empirical classification error, hence is usually the algorithm considered in practice to learn linear classifiers. Figure 1.2 presents a classifier learned by SVM.

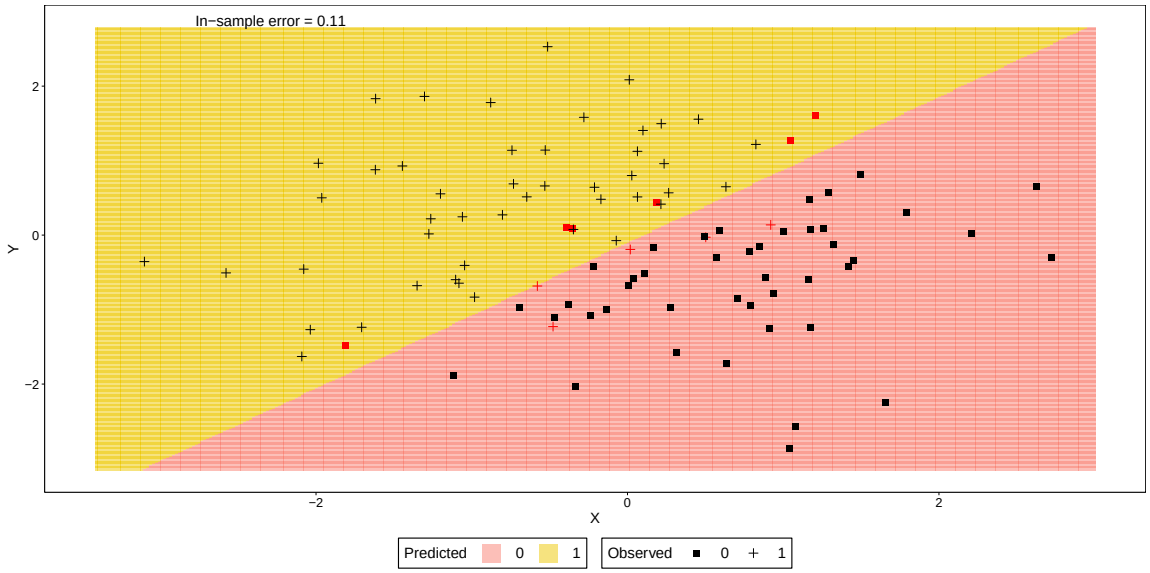

Figure 1.2: *Example of a linear classifier learned by SVM. The points represent the training sample $\mathcal{D}_N$, and their shape is related to their observed value in the output variable. The yellow and orange regions are, respectively, above and below the learned classifier, and represent the points classified as* 1 *and* 0*. The wrongly classified points are in red, and the in-sample error of the estimated classifier is* 0.11.

1.1.2 Model Selection in Machine Learning

The VC theory is general, has a structural importance to the field, and is a useful guide for modeling practical problems. However, since $N(d_{VC}(\mathcal{H}), \epsilon, \delta)$, the least N, given a hypotheses space $\mathcal{H}$, a margin or error ϵ, and a confidence δ, under VC theory bounds, such that

$$\mathbb{P}\left(\sup_{h\in\mathcal{H}} \left|L(h) - L_{\mathcal{D}_N}(h)\right| > \epsilon\right) \leq \delta, \tag{1.3}$$

is not a tight bound,[3] it is usually a meaningless quantity in real application problems.

[3] This is the case because VC bounds are distribution-free, which means they should hold for any distribution P. This translates into holding in the worst-case scenario among the possible data generating distributions.

In fact, the sample size N depends on data availability, which may be conditioned upon several factors, such as technical difficulties and costs. Thus, parameters (N, ϵ, δ) are usually predetermined, so the only free component to be adjusted on VC theory's bounds is the hypotheses space or, more precisely, its VC dimension. This adjustment is usually performed via a data-driven selection of the hypotheses space, known in the literature of the field as Model Selection (see [44, 61, 99, 125] for a review of Model Selection techniques). From now on, we use the words *model*, *hypotheses space* and *hypotheses subspace* (of a hypotheses space) as synonyms.

In order to select the hypotheses space based on data, one might apply a combinatorial algorithm, which searches a family of candidate models seeking to minimize an estimator of the out-of-sample error of the *best* hypothesis of each one. In other words, given a family $\{\mathcal{M}_1, \dots, \mathcal{M}_n\}$ of models, a sample of size N, and a consistent estimator $\hat{L}$ of the out-of-sample error, often given by an independent validation sample or cross-validation, such algorithm returns a model $\mathcal{M}$, whose estimated optimal hypothesis is somewhat the best estimator for a target hypothesis. This framework of Model Selection under a validation sample is depicted in Figure 1.3

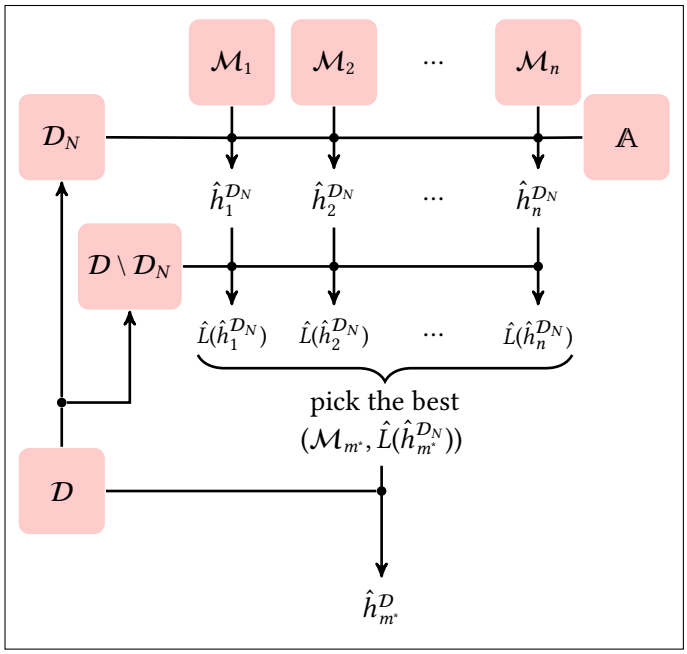

Figure 1.3: *Solution to the Model Selection problem, in which $\hat{h}_i^{\mathcal{D}_N}$, $i = 1, \dots, n$, are the minimizers of $L_{\mathcal{D}_N}$ in each candidate model, and $\hat{L}$ is an estimator of the out-of-sample error, given by the expectation of the loss function under the validation sample $\mathcal{D} \setminus \mathcal{D}_N$. The model with the least validation error $\mathcal{M}_{m^*}$ is chosen, and the hypotheses estimated is $\hat{h}_{m^*}^{\mathcal{D}}$, the minimizer of $L_{\mathcal{D}}$ in $\mathcal{M}_{m^*}$. See [1, Chapter 4] and [32] for more details of this framework. This diagram was adapted from [1] and [32].*

If the candidate models are nested, i.e., $\mathcal{M}_1 \subset \cdots \subset \mathcal{M}_n$, then a method based on the Structured Risk Minimization (SRM) Inductive Principle may be applied to solve this problem (see [150, Chapter 4] for more details, and [5] for an example). In this case, the loss of each model is penalized by a function of its VC dimension to establish a trade-off

See Appendix A for more details.

between low in-sample error and high complexity, establishing a stopping criterion when searching for a model, from simplest to most complex, among the nested candidates.

The SRM methods are actually special cases of a general method to Model Selection, characterized by a penalization of the in-sample error according to the complexity of each model, in both nested and non-nested frameworks (see [99] for an in-depth presentation of Model Selection by penalization, and [6, 16, 82, 83] for results in more specific learning frameworks). Moreover, the classical problem of variable, or feature, selection [60, 77] constitutes another framework for Model Selection, in which a family of partially ordered candidate models is generated through elimination of variables.

A limitation of variable selection is that the family of candidate models is too constrained, so it may not be sharp enough for some problems of interest, while the limitation of other common methods are the restriction to a nested family of candidate models, or the dependence on a choice of penalization.

In this thesis, we propose a family of candidate models, called Learning Spaces, which can be designed with adequate constraints for each class of problems, and has properties which one can take advantage of to implement Model Selection algorithms more efficient than an exhaustive search of the candidate models. In this context, variable selection, SRM and penalization methods, become particular cases.

Concrete example: linear classifiers

The classical solution for the linear classifiers Model Selection problem, under the scheme in Figure 1.3, would be to consider the candidate models

$$\mathcal{M}_1 = \{h_a \in \mathcal{H} \,:\, a_1 = a_2 = 0\} \qquad \mathcal{M}_2 = \{h_a \in \mathcal{H} \,:\, a_2 = 0\}$$
$$\mathcal{M}_3 = \{h_a \in \mathcal{H} \,:\, a_1 = 0\} \qquad \mathcal{M}_4 = \mathcal{H}$$

containing the constant hypotheses ($\mathcal{M}_1$), the hypotheses that do not depend on the second coordinate of X ($\mathcal{M}_2$), the hypotheses that do not depend on the first coordinate of X ($\mathcal{M}_3$) and all hypotheses $\mathcal{H}$ ($\mathcal{M}_4$), which contain the hypotheses that depend on both coordinates of X.

With the notation of Figure 1.3, $\hat{h}_1^{\mathcal{D}_N}, \hat{h}_2^{\mathcal{D}_N}, \hat{h}_3^{\mathcal{D}_N}$ and $\hat{h}_4^{\mathcal{D}_N}$ would be ERM hypotheses of the respective models, and $\hat{L}(\hat{h}_1^{\mathcal{D}_N}), \hat{L}(\hat{h}_2^{\mathcal{D}_N}), \hat{L}(\hat{h}_3^{\mathcal{D}_N})$ and $\hat{L}(\hat{h}_4^{\mathcal{D}_N})$ would be the empirical error of the respective hypotheses in a validation sample $\mathcal{D}/\mathcal{D}_N$, that is a sequence of independent random variables with distribution P that are independent of $\mathcal{D}_N$. We note that, instead of considering ERM hypotheses as the representative hypotheses of each model, one could have considered hypotheses learned by SVM, which could then be evaluated by the empirical classification error under the validation sample.

Due to the independence between $\mathcal{D}_N$ and the validation sample, the minimizer of the validation error $\hat{L}$ among the four ERM hypotheses will not necessarily be $\hat{h}_4^{\mathcal{D}_N}$, which is clearly the minimizer of $L_{\mathcal{D}_N}$ among these hypotheses, as it minimizes the empirical error in $\mathcal{H}$ which contain all four models. Hence, the selected model is not necessarily $\mathcal{M}_4$, and may be a proper subset of $\mathcal{H}$.

We note that this is a special case of variable selection, in which there are two variables,

represented by the coordinates of X, and Model Selection in this instance means selecting on which variables the target hypotheses depend on: none ($\mathcal{M}_1$), only first variable ($\mathcal{M}_2$), only second variable ($\mathcal{M}_3$) or both variables ($\mathcal{M}_4$).

On the one hand, if the target hypotheses do not depend on all variables, then selecting a model this way could enhance the quality of the learning, since one would learn in a proper subset of $\mathcal{H}$, which is less complex, hence the learning would be more efficient. On the other hand, if the target hypotheses depend on all variables, then selecting a model this way could not be efficient, since one might not select a model at all and learn on the whole hypotheses space $\mathcal{H}$ ($\mathcal{M}_4$). In this instance, the Model Selection via variable selection would be too constrained to solve the problem at hand.

1.1.3 Model Selection via Learning Spaces

We propose an extension of the classical learning framework, defining $(\mathcal{H}, \mathbb{L}(\mathcal{H}), \mathbb{A}, \mathcal{D}_N)$ composed by a hypotheses space $\mathcal{H}$; a Learning Space $\mathbb{L}(\mathcal{H})$, which is a poset of subspaces of $\mathcal{H}$, that covers $\mathcal{H}$, and satisfies a property regarding the VC dimension of related subspaces; and a learning algorithm $\mathbb{A}(\mathbb{L}(\mathcal{H}), \mathcal{D}_N)$, which processes $\mathbb{L}(\mathcal{H})$ and a training sample $\mathcal{D}_N$, and returns $\hat{\mathcal{M}} \in \mathbb{L}(\mathcal{H})$, a subspace of $\mathcal{H}$ with nice properties, and $\hat{h}^{\mathbb{A}}_{\hat{\mathcal{M}}} \in \hat{\mathcal{M}}$, a hypothesis that seeks to approximate the target $h^\star$ of $\mathcal{H}$. Under this framework, the learning of hypotheses is performed in two consecutive steps: one first learns a model $\hat{\mathcal{M}}$ among the candidates in $\mathbb{L}(\mathcal{H})$, and then learns a hypothesis $\hat{h}^{\mathbb{A}}_{\hat{\mathcal{M}}} \in \hat{\mathcal{M}}$ in it. This framework is depicted in Figure 1.4.

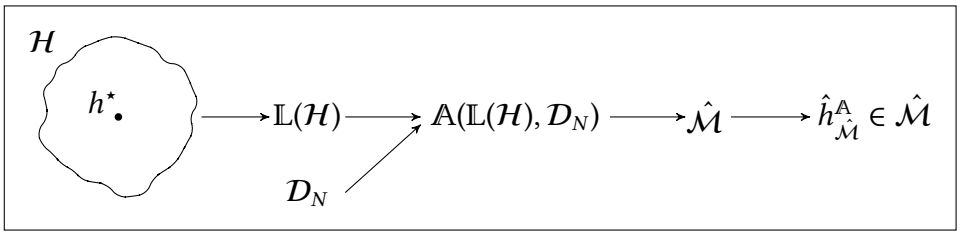

Figure 1.4: *Learning framework via Learning Spaces.*

An interesting feature of Model Selection via Learning Spaces is an implicit regularization [147]. When we consider candidate models with distinct complexities, and optimize over them an error measure based, for example, on a validation method, we have an implicit regularization, in the sense of avoiding selecting models too complex, that would lead to overfitting, and favoring simpler models, which may better capture the patterns of the data, hence better generalize (have small out-of-sample error).

In this sense, Model Selection via Learning Spaces may be considered an *implicit complexity regularizer*, where the regularization is due to considering a great family of candidate models with distinct complexities, and optimizing an error measure which tries to protect against overfitting. This differs from typical regularization procedures which often penalize an error measure by the complexity of each hypothesis, and optimize the penalized error over $\mathcal{H}$, usually learning directly a hypothesis without searching for a subspace of $\mathcal{H}$ first [24, 99, 102, 112, 113].

As opposed to methods for Model Selection based on penalization of loss functions, the spirit of the framework proposed here is to *not* penalize the loss function, but rather obtain a general and consistent framework for Model Selection based on resampling techniques such as cross-validation. Hence, in this scenario, the obtained regularization is implicit, since it is not explicitly considered in the loss function by penalizing it. However, although not in the spirit of the paper, penalized methods also fit into the framework. We discuss why this is the case in Section 5.2.

1.1.4 Estimation errors under Model Selection

When selecting a model by any approach, one should mind the estimation errors of learning on a given space. At principle, when one learns on $\mathcal{H}$, disregarding any hypothesis not in it, he commits two types of errors

$$\sup_{h\in\mathcal{H}}\left|L_{\mathcal{D}_N}(h) - L(h)\right| \qquad \text{and} \qquad L(\hat{h}^{\mathbb{A}}) - L(h^\star), \tag{1.4}$$

which we call type I and type II estimation error, respectively. If type I estimation error is small, then we can estimate the out-of-sample error of any hypothesis in $\mathcal{H}$ by the in-sample error with great precision. If type II estimation error is small, then the hypothesis $\hat{h}^{\mathbb{A}}$, estimated by the algorithm $\mathbb{A}$, well approximates a target $h^\star$. Since, fixed the margin of error ϵ and the sample size N, the VC bounds for the tail probabilities (cf. (1.3)) of both errors in (1.4) are increasing functions of VC dimension (see [42, 149] and Appendix A), the smaller the hypotheses space is, in the VC dimension sense, the lesser are the estimation errors on it, with high probability, so that we may regulate the VC dimension of the hypotheses space to better estimate.

This may be accomplished by selecting a proper subset $\mathcal{M} \subset \mathcal{H}$ on which to learn. However, when we restrict the learning to such subspace, we commit another two types of errors, which we call types III and IV estimation errors, that are, respectively,

$$L(h^\star_{\mathcal{M}}) - L(h^\star) \qquad \text{and} \qquad L(\hat{h}^{\mathbb{A}}_{\mathcal{M}}) - L(h^\star),$$

in which $h^\star_{\mathcal{M}}$ is a minimizer of L in $\mathcal{M}$ and $\hat{h}^{\mathbb{A}}_{\mathcal{M}}$ is the hypothesis in $\mathcal{M}$ estimated by the algorithm $\mathbb{A}$. If type III estimation error is small, then a target hypothesis $h^\star_{\mathcal{M}}$ of $\mathcal{M}$ well approximates a target hypothesis $h^\star$ of $\mathcal{H}$. If type IV estimation error is small, then the estimated hypothesis $\hat{h}^{\mathbb{A}}_{\mathcal{M}}$ of $\mathcal{M}$ well approximates a target of $\mathcal{H}$. If both types III and IV estimation errors are small, then it is possible to learn on $\mathcal{M}$ without adding a great bias to the learning process.

If there is no prior information about the target $h^\star$ which allow us to consider a subset of $\mathcal{H}$ such that $\mathcal{M} \ni h^\star$, so type III estimation error is zero and type IV reduces to type II[4], it may not be possible to restrict $\mathcal{H}$ beforehand and still estimate a good hypothesis relatively to $h^\star$. However, we may learn on a random subset $\hat{\mathcal{M}} \subset \mathcal{H}$ in a manner such that all four estimation errors are asymptotically zero, i.e., tend in probability to zero as the sample size increases. Such a subset is random, for it depends on sample $\mathcal{D}_N$: $\hat{\mathcal{M}}$ is

[4] In this case, type II estimation error is $L(\hat{h}^{\mathbb{A}}_{\mathcal{M}}) - L(h^\star_{\mathcal{M}})$, which is equal to type IV if $h^\star \in \mathcal{M}$.

learned from data.

The framework for Model Selection based on Learning Spaces selects a model $\hat{\mathcal{M}} \in \mathbb{L}(\mathcal{H})$ which is such that types I and II estimation errors tend to be smaller than on $\mathcal{H}$, and types III and IV estimation errors are asymptotically zero. In the proposed approach, all estimation errors converge to zero when the sample size tends to infinity, and $\hat{\mathcal{M}}$ converges with probability one to the target subspace $\mathcal{M}^\star$ of $\mathcal{H}$, which is the model in $\mathbb{L}(\mathcal{H})$ with the least VC dimension that contains a target hypothesis $h^\star$ (cf. Figure 1.16). We say that $\hat{\mathcal{M}}$ is statistically consistent if it satisfies these two properties. Our approach does not demand the specification of a hypotheses space $\mathcal{M}$ *a priori*, but rather introduces the learning of a hypotheses space from data among those in $\mathbb{L}(\mathcal{H})$ as a mean to better learn hypotheses, so prior information is all embedded in $\mathbb{L}(\mathcal{H})$.

The target model is central in our approach. As is the case in all optimization problems, there must be an optimal solution to the Model Selection problem, which satisfies certain desired conditions. In the proposed framework, the optimal solution is the model in $\mathbb{L}(\mathcal{H})$ with the least VC dimension which contains a target hypothesis. From the perspective of estimation errors, this model provides the best circumstances in $\mathbb{L}(\mathcal{H})$ to learn hypotheses with a fixed sample of size N. On the one hand, type III estimation error is zero and type IV reduces to type II. On the other hand, the VC dimension is minimal under these constraints, so the bounds for the tail probabilities of types I and II estimation errors are tightest.

1.1.5 Computational aspects of Model Selection via Learning Spaces

The concept of target model brings a new learning paradigm, under which one estimates it seeking to better estimate a target hypothesis, with a fixed sample size. This paradigm, represented in Figure 1.4, is in contrast with the classical Machine Learning framework, presented in Figure 1.1. Throughout this thesis, we present the main results from the perspective of this paradigm, offering a guide on how one can, theoretically, better estimate, without having to increase the sample size, by incorporating prior knowledge about the problem at hand into $\mathbb{L}(\mathcal{H})$, and employing high computational power.

Indeed, apart from the statistical consistency of the method, it is important to consider the computational aspects of learning hypotheses via Learning Spaces. At principle, to select a model from $\mathbb{L}(\mathcal{H})$, one would have to apply a combinatorial algorithm that performs an exhaustive search of it, looking for the model which minimizes some error measure. If the cardinality of $\mathbb{L}(\mathcal{H})$ is too great, which will be often the case, this exhaustive search cannot be performed, so computing $\hat{\mathcal{M}}$ is not possible. Nevertheless, due to the structure of some Learning Spaces, there may exist non-exhaustive algorithms to compute $\hat{\mathcal{M}}$, so, even if these algorithms are highly complex, they may still be employed to solve practical problems when high computational power is available.

In this thesis, we define the U-curve properties that, when satisfied, allow a non-exhaustive calculation of $\hat{\mathcal{M}}$ via a U-curve algorithm. The properties are rigorous mathematical definitions of a phenomenon intuitively related to the bias-variance trade-off, Occam's razor, peaking phenomenon and the curse of dimensionality [20, 45, 58, 126, 162], in which the estimated error of a model decreases with its complexity up to a point when

there is an inflection point, and the error starts increasing with the complexity, forming a U-shaped curve. This heuristic behavior, which is supported by empirical evidence, but often does not have a mathematical proof, is rigorously defined here for models organized in a lattice, and proved to be satisfied in certain instances.

We show that a specific Learning Space satisfies a U-curve property, and establish a sufficient condition for it that is closely related to convexity, but under a lattice algebra. This is, to our knowledge, one of the few rigorous result in the literature asserting general conditions under which a non-exhaustive combinatorial search of candidate models for the purpose of Model Selection returns an optimal solution. We then briefly discuss how one may take advantage of this property to develop U-curve algorithms that return $\hat{\mathcal{M}}$, if a U-curve property is satisfied, but may also be employed efficiently to obtain suboptimal solutions when a property is not satisfied, as has been done for variable selection, where the candidate models form a Boolean lattice (see [8, 55, 130, 131, 133] for more details and Section B.1 for the definition of a Boolean lattice).

We now formally define, in a more general manner, the ideas discussed here from the point of view of Machine Learning.

1.2 Framework for the learning of hypotheses

1.2.1 Hypotheses spaces and loss functions

Let Z be a random vector defined on a probability space $(\Omega, \mathcal{S}, \mathbb{P})$, with range $\mathcal{Z} \subset \mathbb{R}^d$, $d \geq 1$. Denote $P(z) := \mathbb{P}(Z \leq z)$ as the probability distribution of Z at point $z \in \mathcal{Z}$, which we assume unknown, but fixed throughout this thesis. Define a sample $\mathcal{D}_N = \{Z_1, \ldots, Z_N\}$ as a sequence of independent and identically distributed random vectors, defined on $(\Omega, \mathcal{S}, \mathbb{P})$, with distribution P.

Let $\mathcal{H}$ be a general set, whose typical element we denote by h, which we call hypotheses space. We denote subsets of $\mathcal{H}$ by $\mathcal{M}_i$, indexed by the positive integers, i.e., $i \in \mathbb{Z}_+$. We may also denote a subset of $\mathcal{H}$ by $\mathcal{M}$ to ease notation. Throughout this thesis, we consider *model* and *subset of* $\mathcal{H}$ as synonyms.

For each hypothesis in $\mathcal{H}$, we assign a value indicating the loss incurred by the use of such hypothesis as an explanation of a feature of Z (phenomenon). Let $\ell : \mathcal{Z} \times \mathcal{H} \mapsto \mathbb{R}_+$ be a loss function, which represents the loss $\ell(z, h)$ that incurs when one applies hypothesis $h \in \mathcal{H}$ to *explain* a feature of point $z \in \mathcal{Z}$. Denoting $\ell_h(z) := \ell(z, h)$ for $z \in \mathcal{Z}$, we assume that, for each $h \in \mathcal{H}$, the composite function $\ell_h \circ Z$ is $(\Omega, \mathcal{S})$-measurable. Sometimes we assume that ℓ is bounded, but it is not necessary for all theory developed, only where specified. For examples of hypotheses spaces and loss functions, see Section 1.2.2.

The out-of-sample error, also known in the literature as risk or loss, of a hypothesis $h \in \mathcal{H}$ is defined as

$$L(h) := \mathbb{E}[\ell_h(Z)] = \int_{\mathcal{Z}} \ell(z, h)\, dP(z),$$

in which $\mathbb{E}$ means expectation under $\mathbb{P}$. This is to be interpreted as the *mean loss* incurred when hypothesis h is used to explain a given feature of Z.

The out-of-sample error is fixed, but unknown, as is P. Therefore, to assess the out-of-sample error of a hypothesis, one needs to estimate it. One possible estimator is the in-sample error, or empirical error, of a hypothesis h, defined as

$$L_{\mathcal{D}_N}(h) := \frac{1}{N}\sum_{i=1}^{N} \ell(Z_i, h),$$

that is the empirical mean of $\ell_h(Z)$ on sample $\mathcal{D}_N$.

There may be equivalent representations for a set of hypotheses and the error associated to them. Moreover, different hypotheses spaces may have common features which make them equivalent, even though they may be of distinct nature. We define the equivalence of hypotheses spaces, which is a useful concept when developing Model Selection methods, since one can implement a method that works for a class of equivalent hypotheses spaces.

Definition 1.1. ***(Equivalence of hypotheses spaces)***

(a) Two hypotheses spaces $\mathcal{H}_1$ and $\mathcal{H}_2$ are equivalent if, and only if, there exists a bijective transformation $T : \mathcal{H}_1 \mapsto \mathcal{H}_2$. We denote $\mathcal{H}_1 \sim \mathcal{H}_2$.

(b) Let $\ell_1 : \mathcal{Z}\times\mathcal{H}_1 \mapsto \mathbb{R}_+$ and $\ell_2 : \mathcal{Z}\times\mathcal{H}_2 \mapsto \mathbb{R}_+$ be loss functions associated to hypotheses spaces $\mathcal{H}_1$ and $\mathcal{H}_2$, respectively. We say that $(\mathcal{H}_1, \ell_1)$ is equivalent to $(\mathcal{H}_2, \ell_2)$, and denote $(\mathcal{H}_1, \ell_1) \sim (\mathcal{H}_2, \ell_2)$, if, and only if, there exists a bijective transformation $T : \mathcal{H}_1 \mapsto \mathcal{H}_2$ such that

$$\ell_1(z, h_1) \leq \ell_1(z, h_2) \iff \ell_2(z, T(h_1)) \leq \ell_2(z, T(h_2)),$$

for $h_1, h_2 \in \mathcal{H}_1$ and all $z \in \mathcal{Z}$.

Assume there is a bijective transformation $T : \mathcal{H}_1 \mapsto \mathcal{K}$ from $\mathcal{H}_1$ into an arbitrary set $\mathcal{K}$, which we call a *representation* for the hypotheses in $\mathcal{H}_1$. Since the equivalence relation is transitive, if $\mathcal{H}_1 \sim \mathcal{K}$ and $\mathcal{H}_1 \sim \mathcal{H}_2$, then $\mathcal{H}_2 \sim \mathcal{K}$, hence, given a representation $\mathcal{K}$, there is a class of hypotheses spaces which are represented by it.

Common representations are such that $\mathcal{K} \subset \mathbb{R}^d$, so hypotheses in $\mathcal{H}_1$ are represented by d-dimensional vectors of parameters, or $\mathcal{K} = \{h : \mathcal{X} \mapsto \mathcal{Y}\}$, so the hypotheses in $\mathcal{H}_1$ are represented by functions from a set $\mathcal{X}$ to a set $\mathcal{Y}$. Indeed, an important class of hypotheses spaces are those whose hypotheses can be represented by functional relations. We call them functional hypotheses spaces, as follows.

Definition 1.2 (Functional hypotheses spaces). *A hypotheses space $\mathcal{H}$ is said functional, if there exists a bijective transformation $T : \mathcal{H} \mapsto \mathcal{K}$, in which $\mathcal{K} = \{h : \mathcal{X} \mapsto \mathcal{Y}\}$ is a set of functions, with $\mathcal{Z} = \mathcal{X} \times \mathcal{Y} \subset \mathbb{R}^{d_X + d_Y}$, $d_X, d_Y \geq 1$.*

In functional hypotheses spaces, we may decompose $Z = (X, Y)$, in which X and Y are random vectors, defined on $(\Omega, \mathcal{S}, \mathbb{P})$, with ranges $\mathcal{X} \subset \mathbb{R}^{d_X}$ and $\mathcal{Y} \subset \mathbb{R}^{d_Y}$, respectively, and the hypotheses may be expressed as functional relations $h : \mathcal{X} \mapsto \mathcal{Y}$ between the random vectors X and Y. By properly choosing loss functions, the loss incurred by applying a hypothesis $h \in \mathcal{H}$ will be that of *predicting* Y by $h(X)$.

1.2.2 Examples of hypotheses spaces and loss functions

We present some examples of hypotheses spaces, and suitable loss functions for them.

Example 1.1 (Maximum Likelihood Methods)**.** Let $\mathcal{H} \subset \mathbb{R}^d$, $d \geq 1$, $\mathcal{Z} \subset \mathbb{R}^{d_Z}$, $d_Z \geq 1$, and $f(\cdot|\theta) : \mathcal{Z} \mapsto \mathbb{R}_+$ be a probability function, or probability density function, for each[5] $\theta \in \mathcal{H}$. We consider the loss function

$$\ell(z, \theta) = -\log f(z|\theta) \qquad\qquad z \in \mathcal{Z},\ \theta \in \mathcal{H},$$

so, given $\mathcal{D}_N$, the in-sample error

$$L_{\mathcal{D}_N}(\theta) = -\frac{1}{N}\sum_{i=1}^{N} \log f(Z_i|\theta) \tag{1.5}$$

actually represents minus the log-likelihood function of the parameter θ. Hence, Maximum Likelihood Methods [136] may be expressed under the hypotheses space framework (see Example 1.7 for more details). ■

Example 1.2 (Regression)**.** Let $\mathcal{H} \subset \mathbb{R}^d$, $d \geq 1$, and $\mathcal{Z} = \mathcal{X} \times \mathcal{Y} \subset \mathbb{R}^{d_X+1}$, $d_X \geq 1$. Let $f(\cdot|\theta) : \mathcal{Y} \mapsto \mathbb{R}_+$ be a probability function, or probability density function, for each $\theta \in \mathbb{R}^{d_\theta}$, $d_\theta \geq 1$, and $g_h : \mathbb{R}^{d_X} \mapsto \mathbb{R}^{d_\theta}$ be smooth functions indexed by d-dimensional vectors $h \in \mathcal{H}$, called link functions. In regression models based on the Maximum Likelihood Principle, one considers the loss function

$$\ell((x, y), h) = -\log f(y|g_h(x)) \qquad\qquad (x, y) \in \mathcal{Z},\ h \in \mathcal{H},$$

so the in-sample error

$$L_{\mathcal{D}_N}(h) = -\frac{1}{N}\sum_{i=1}^{N} \log f(Y_i|g_h(X_i))$$

represents minus the log-likelihood function of link function parameter h, given a sample $\mathcal{D}_N$.

Although there are regression frameworks which are slightly different, the Generalized Linear Models (GLM) [108], and the more comprehensive Generalized Additive Models for Location Scale and Shape (GAMLSS) [132], are important examples of such framework for regression, in which the link functions are of the form

$$g_h(x) = (g \circ h)(x) = g(hx),$$

in which h is actually a matrix of dimension $d = d' \times d_X$, $d' \geq 1$, and $g : \mathbb{R}^{d'} \mapsto \mathbb{R}^{d_\theta}$ is a smooth fixed function. In this case, the link function is defined by applying a smooth function to a linear transformation of the input variable x, and its parameters are the coefficients of such linear transformation.

A special case is that of linear regression, when $f(\cdot|\theta)$ is the probability density function

[5] We use notation θ instead of h, as it is the usual notation for Maximum Likelihood Methods [136].

of a Normal distribution, with mean θ and a fixed, but unknown, variance σ^2, and the link function is given by

$$g_h(x) = hx,$$

with $d' = 1$. In this case, the loss function equals

$$\ell((x, y), h) = c_1[hx - y]^2 + c_2,$$

in which $c_1, c_2 \in \mathbb{R}$, $c_1 > 0$, are constant terms dependent on $\mathcal{D}_N$ and σ^2, but not on h, so this hypotheses space with the likelihood loss is equivalent to the hypotheses space of linear functions with the quadratic loss function (see Example 1.3). ■

Remark 1.3. *In Examples 1.1 and 1.2, in order for $\ell(z, \theta) \geq 0$, we assume there exists a constant $C \in \mathbb{R}_+$ such that $\ell(z, \theta) + C \geq 0$, so we consider ℓ plus this constant as the loss function. Observe that by summing this constant, the qualitative behavior of L and $L_{\mathcal{D}_N}$ does not change, since the minimizers of them remain the same, and the hypotheses space framework still represents Maximum Likelihood Methods.*

Example 1.3 (Functional real-valued hypotheses space)**.** Let $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, with $\mathcal{X} \subset \mathbb{R}^{d_X}$, $\mathcal{Y} \subset \mathbb{R}$, $d_X \geq 1$, and consider $\mathcal{H} = \{h : \mathcal{X} \mapsto \mathcal{Y}\}$ to be a subset of the space of all functions from $\mathcal{X}$ to $\mathcal{Y}$. Two important loss functions in this case are, respectively, when the cardinality of $\mathcal{Y}$ is infinite or finite,

$$\ell_1((x, y), h) = [h(x) - y]^2$$
$$\ell_2((x, y), h) = \mathbb{1}\{h(x) \neq y\} := \begin{cases} 1, & \text{if } h(x) \neq y \\ 0, & \text{if } h(x) = y \end{cases}$$

called, respectively, the quadratic loss function and simple loss function. One then obtains as out-of sample error, respectively, the mean quadratic error and the classification error:

$$L_1(h) = \mathbb{E}\left([h(X) - Y]^2\right) \qquad L_2(h) = \mathbb{P}\left(h(X) \neq Y\right).$$

Analogously, the in-sample error is the empirical mean quadratic error and classification error on sample $\mathcal{D}_N$.

Functional real-valued hypotheses spaces are the most common hypotheses spaces. They may be employed when modeling classification problems, which are quite important in Machine Learning theory, or to solve regression problems via linear regression, as linear regression is equivalent to the functional hypotheses space of linear functions under the quadratic loss. ■

Example 1.4 (Boolean hypotheses space)**.** A special case of functional hypotheses space is that of the Boolean functions, that is

$$\mathcal{H} = \{h : \{0, 1\}^d \mapsto \{0, 1\}\},$$

for a $d \geq 1$. This is an important hypotheses space with applications in many areas, such as image recognition [11, 49], genetics [14] and cryptography [38, 156]. The usual loss function in this case is the simple loss function. ■

Example 1.5 (Linear classifiers)**.** Another important functional hypotheses space is that of the linear classifiers, in which the input of the hypotheses is a vector in $\mathbb{R}^d$, $d \geq 1$, and the output is binary. This hypotheses space may be defined as

$$\mathcal{H} = \left\{ h_a(x) = \frac{1}{2}\operatorname{sgn}\left\{ a_0 + \sum_{i=1}^{d} a_i x_i \right\} + \frac{1}{2} \ : \ a_0, a_i \in \mathbb{R} \right\},$$

in which $x = (x_1, \dots, x_d) \in \mathbb{R}^d$, and h_a is the function indexed by its parameters $a = (a_0, \dots, a_d) \in \mathbb{R}^{d+1}$.

Each linear classifier is represented by a hyperplane which divides $\mathbb{R}^d$ into two subspaces, containing the points classified as zero and one, respectively. Although linear classifiers are quite simple, they are a very good introductory example of classifier, and was one of the first hypotheses applied in classification problems [135]. ■

Example 1.6 (Deep neural network)**.** For each $\theta := (\theta_0, \dots, \theta_{m+1}) \in \Theta \subset \mathbb{R}^{t_0} \times \cdots \times \mathbb{R}^{t_{m+1}}$, let $f_0^{\theta_0}, \dots, f_{m+1}^{\theta_{m+1}}$, $m \geq 2$, be a sequence of functions

$$f_0^{\theta_0} \ : \ \mathcal{X} \mapsto \mathbb{R}^{d_1} \qquad \cdots \qquad f_k^{\theta_k} \ : \ \mathbb{R}^{d_k} \mapsto \mathbb{R}^{d_{k+1}} \qquad \cdots \qquad f_{m+1}^{\theta_{m+1}} \ : \ \mathbb{R}^{d_{m+1}} \mapsto \mathcal{Y} \tag{1.6}$$

for $1 \leq k \leq m$, that are completely determined by parameters θ, in which $1 \leq d_k, t_{k'} < \infty$ for all $1 \leq k \leq m+1$ and $0 \leq k' \leq m+1$.

Assume that $m \geq 2$ is fixed, and a class

$$\mathcal{A} = \left\{ \{ f_0^{\theta_0}, \dots, f_{m+1}^{\theta_{m+1}} \} \ : \ \theta \in \Theta \right\}$$

satisfying (1.6) is given. Then, for each $\theta \in \Theta$, define $h_\theta \ : \ \mathcal{X} \mapsto \mathcal{Y}$ as

$$h_\theta(x) := f_{m+1}^{\theta_{m+1}} \circ f_m^{\theta_m} \circ \cdots \circ f_0^{\theta_0}(x), \tag{1.7}$$

for $x \in \mathcal{X}$. We call $\mathcal{A}$ a *deep neural network* (DNN) architecture with m hidden layers, which can represent the classifiers in set

$$\mathcal{R}(\mathcal{A}) := \left\{ h_\theta \ : \ \theta \in \Theta \right\}, \tag{1.8}$$

that is a collection of functions with domain $\mathcal{X}$ and image $\mathcal{Y}$. In Figure 1.5 there is an illustration of such a DNN architecture.

Our definition of DNN does not seek to contemplate all kinds of DNNs used nowadays, but rather focus on less sophisticated architectures, which include Fully Connected DNNs, when $f_k^{\theta_k}$ is given by applying a multidimensional linear transformation to the input, followed by an activation function coordinate-wise. In this case, θ_k is organized as a matrix with dimensions d_{k+1} and d_k, considering that $t_k = d_k \times d_{k+1}$, and

$$f_k^{\theta_k}(w) = f(\theta w),$$

in which $f \ : \ \mathbb{R} \mapsto \mathbb{R}$ is a smooth function, called activation function, and is applied coordinate-wise to the vector θw, $w \in \mathbb{R}^{d_k}$. We refer to [3] for an introduction to DNNs.

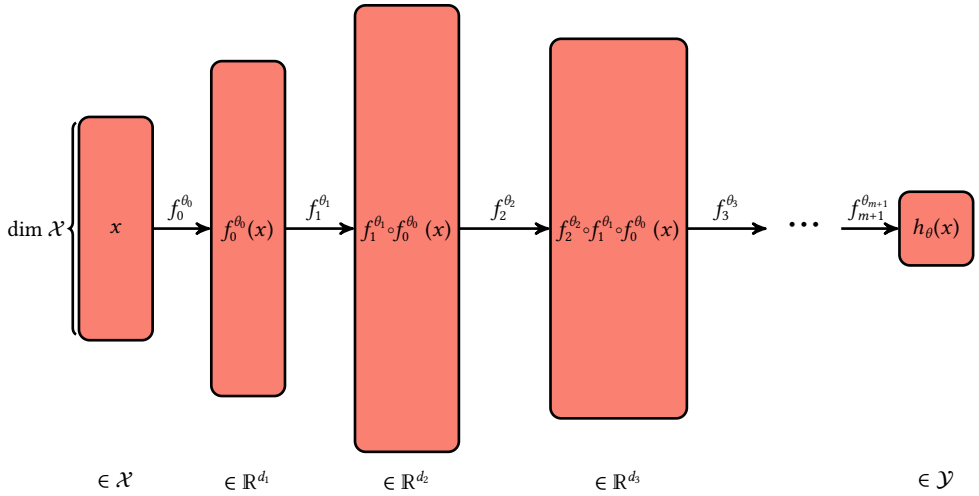

Figure 1.5: *Illustration of a DNN hypothesis, following architecture $\mathcal{A}$, as defined in (1.7). The rectangle's height is proportional to the dimension of the input variable of the respective layer, i.e., $d_1, \ldots, d_m$, which can change from layer to layer.*

There are a lot of loss functions which can be employed in DNNs besides the quadratic loss function, when $\mathcal{Y} \subset \mathbb{R}$ and has infinitely many points, and the simple loss functions in classification problems, when $|\mathcal{Y}| < \infty$. This is specially true in classification problems, and we refer to [75] for an overview of these loss functions. ■

These examples are not exhaustive: there are many more hypotheses spaces in learning theory, and we will define more of them in later chapters. Nevertheless, these are well-known, and quite useful, hypotheses spaces, which may aid the understanding of the theory developed here. Hence, when we present a new concept throughout this thesis, we will discuss it in view of one or more of the examples presented here, to ease its understanding.

1.2.3 Target hypotheses

The main goal when learning hypotheses is to approximate target hypotheses, that are hypotheses in $\mathcal{H}$ which minimize the out-of-sample error. These hypotheses are in set

$$h^\star := \arg\min_{h \in \mathcal{H}} L(h).$$

As $\mathcal{H}$ may be a proper subset of the space of all *possible hypotheses*, when it is defined[6], there may exist a possible hypothesis $g, g \notin \mathcal{H}$, with $L(g) < L(h^\star)$. However, we focus on approximating the best hypotheses in $\mathcal{H}$, so throughout this thesis we disregard all hypotheses outside it, and take $\mathcal{H}$ as all hypotheses one is willing to consider.

Furthermore, we will also be interested in target hypotheses of subsets of $\mathcal{H}$, which

[6] For example, if $\mathcal{H}$ is a functional hypotheses space containing functions from $\mathcal{X}$ to $\mathcal{Y}$, the space of all possible hypotheses is natural: the set of all measurable functions from $\mathcal{X}$ to $\mathcal{Y}$.

are in

$$h_i^\star := \arg\min_{h\in\mathcal{M}_i} L(h) \qquad\qquad h_{\mathcal{M}}^\star := \arg\min_{h\in\mathcal{M}} L(h),$$

depending on the subset. Observe that, since the data distribution is unknown, so is the out-of-sample error, and consequently the target hypotheses are unknown. Therefore, to approximate these hypotheses, one should estimate them via a sample of Z.

Under the Empirical Risk Minimization (ERM) paradigm [150], which proposes the minimization of the in-sample error as a method to approximate target hypotheses, we estimate the target hypotheses by

$$\hat{h}^{D_N} := \arg\min_{h\in\mathcal{H}} L_{D_N}(h),$$

while the estimated target hypotheses of models are in

$$\hat{h}_i^{D_N} := \arg\min_{h\in\mathcal{M}_i} L_{D_N}(h) \qquad\qquad \hat{h}_{\mathcal{M}}^{D_N} := \arg\min_{h\in\mathcal{M}} L_{D_N}(h).$$

We assume the minimum of L and L_{D_N} is achieved in $\mathcal{H}$, and in all subsets of it that we consider throughout this thesis, so the sets above are not empty. We also assume these minimums are $(\Omega, \mathcal{S})$-measurable. To ease notation, we may simply denote $\hat{h}$ as a hypothesis estimated by ERM or other algorithm, when the algorithm is not of importance.

We present the target hypotheses concept from the point of view of some of our examples.

Example 1.7 (Maximum Likelihood Methods)**.** Under the ERM principle, $\hat{\theta}^{D_N}$, a minimizer of (1.5) in $\mathcal{H}$, will be the Maximum Likelihood Estimator, that is a hypothesis which maximizes the likelihood function. Hence, estimation by Maximum Likelihood is a special case of learning hypotheses under the ERM principle.

In classic Statistics, it is assumed that

$$P(z) = \int_{A_z} f(z'|\theta^\star)\, d\lambda(z') \qquad\qquad \forall\, z \in \mathcal{Z}, \tag{1.9}$$

in which $A_z = \{z' \in \mathbb{R}^d \,:\, z' \leq z\}$, for a $\theta^\star \in \mathcal{H}$, and some measure λ (e.g., product of Lebesgue measures and counting measures). In other words, it is assumed a distribution for Z with density $f(\cdot|\theta^\star)$. Under a Statistical Learning framework, one does not assume any distribution for Z, so $\theta^\star$ will not be such that (1.9) holds, but will rather represent a distribution with density of form f which *best* approximates the unknown real distribution of Z, in the following sense.

A possible way of measuring the statistical *distance* between two distributions P and Q is through

$$D_{KL}(P\|Q) := \int_{\mathcal{Z}} \log\frac{P(z)}{Q(z)}\, dP(z),$$

the Kullback–Leibler divergence [87] of Q relative to P. It represents a divergence from

Q to P, when Q is used as an approximation to P, the real data distribution. If the loss function is of form (1.5), then

$$L(\theta) = D_{KL}(P\|Q_\theta) - \int_{\mathcal{Z}} \log P(z)\, dP(z) = D_{KL}(P\|Q_\theta) + H(P),$$

in which Q_θ is the distribution with density $f(\cdot|\theta)$, and $H(P)$ is the entropy of P.

Since $H(P)$ does not depend on θ, the minimizer of $L(\theta)$ is actually the minimizer of the Kullback–Leibler divergence in the space of probability measures with density $f(\cdot|\theta)$. Hence, $\theta^\star$ is the parameter of the distribution Q_θ that best approximates P according to the Kullback–Leibler divergence. If $D_{KL}(P\|Q_\theta) = 0$, then $P(z) = Q(z)$ for all $z \in \mathcal{Z}'$, with $\mathbb{P}(Z \in \mathcal{Z}') = 1$, and the statistical assumption (1.9) holds.

Nevertheless, classic Statistics methods take a step further: after estimating $\theta^\star$ by $\hat{\theta}^{\mathcal{D}_N}$, one then builds a confidence interval for $\theta^\star$, and tests *hypotheses* formulated in forms such as $\theta^\star = \theta_0$, tested via its statistical significance [136]. Therefore, our definition of hypotheses is different from that of classic Statistics: we call hypothesis any $\theta \in \boldsymbol{\mathcal{H}}$, while in classic Statistics hypothesis is a statement such as $\theta^\star =, \leq, \geq \theta_0$. ■

Example 1.8 (Regression)**.** As is the case in Example 1.7, in regression problems under the ERM principle, $\hat{h}^{\mathcal{D}_N}$ will be a Maximum Likelihood Estimator. Again, in classic Statistics, it is assumed that

$$\mathbb{P}\left(Y \leq y | X = x\right) = \int_{-\infty}^{y} f(y'|g_{h^\star}(x))\, d\lambda(y') \qquad \forall\, (x, y) \in \mathcal{Z},$$

for some $h^\star \in \boldsymbol{\mathcal{H}}$, i.e., that the conditional distribution of Y given $X = x$ has density (probability function) f with parameter $\theta = g_{h^\star}(x)$. Again, in a Statistical Learning framework, this is not necessary, as $h^\star$ will represent a conditional distribution of Y given X, in family f with link function g_h, which *best* approximates the real conditional distribution of Y given X in the Kullback–Leibler divergence sense.

Moreover, in classic Statistics, one builds confidence intervals for the coordinates of $h^\star$, and tests hypotheses such as $Ah^\star = h_0$, in which A is a matrix. In this case, the hypotheses are generally statements about the dependence between Y and X, through how the parameter θ of the conditional distribution of Y given X depends on X. In this context, the word hypotheses means statements about the dependence between X and Y, rather than $h \in \boldsymbol{\mathcal{H}}$. ■

Example 1.9 (Functional hypotheses spaces)**.** Target hypotheses of certain functional hypotheses spaces have some interesting meaning, and are equivalent to classical methods. For instance, consider

$$H_1 = \left\{ h(x) = a_0 + \sum_{i=1}^{d} a_i x_i \;:\; a_0, a_i \in \mathbb{R} \right\},$$

the space of linear functions from $\mathbb{R}^d$ to $\mathbb{R}$. Considering the quadratic loss function, the ERM principle is equivalent to the least squares method [109], and $\hat{h}^{\mathcal{D}_N}$ is the hyperplane that *better* approximates the points in $\boldsymbol{\mathcal{D}}_N$, in the sense of minimizing the mean square

distance between the hyperplane and the points. By choosing other hypotheses space with the quadratic loss function, one has other methods, such as polynomial fitting. Figure 1.6 shows examples of sample points and ERM hypothesis for the two-dimensional case, in which $\mathcal{X} = \mathbb{R}$ and $\mathcal{Y} = \mathbb{R}$.

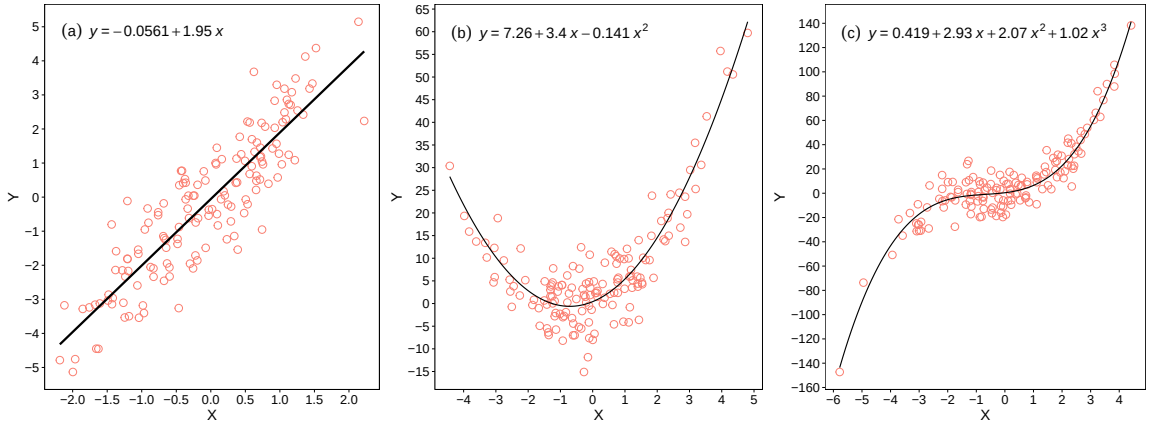

Figure 1.6: *Examples of (a) linear regression and polynomial fitting ((b) degree 2 and (c) degree 3). The points represent the training sample $\mathcal{D}_N$, and the curve is the minimizer of the mean quadratic error on the sample. These are the least square polynomials.*

On the other hand, considering for instance the linear classifiers with the simple loss function, the estimated hyperplanes will be such that the number of sample points in the subspace referring to its label is maximum, so the classification error in the sample is minimum. There will be, in general, infinite hyperplanes which minimize the error in the sample, and there are some specific estimation techniques, such as Support Vector Machine (SVM) [37], that enforce some restrictions on the solution, so it is more efficiently computed. Figure 1.2 in Section 1.1.2 shows the sample points and learned hypothesis via SVM for an example with $d = 2$. ■

1.2.4 VC dimension

In order to carry out the agenda of choosing collections of subsets of $\mathcal{H}$, ordered by complexity, as candidate models for Model Selection, we need to mathematically define the *complexity* of a model. Although other measures of complexity could be suitable for our purposes (see [23, 107], [139, Chapter 26] for examples), we consider the complexity of a hypotheses space, under a loss function ℓ, to be its VC dimension. We start by defining the shatter coefficient of a set of binary functions. See [150, Chapter 3] for more details.

Definition 1.4 (Shatter coefficient). *Let $\mathcal{G} = \{I : \mathcal{Z} \mapsto \{0,1\}\}$ be a set of binary functions with domain $\mathcal{Z}$. The N-shatter coefficient of $\mathcal{G}$ is defined as*

$$S(\mathcal{G}, N) = \max_{(z_1,\ldots,z_N)\in\mathcal{Z}^N} \left| \left\{ \big(I(z_1), \ldots, I(z_N)\big) \; : \; I \in \mathcal{G} \right\} \right|,$$

for $N \in \mathbb{Z}_+$, in which $|\cdot|$ is the cardinality of a set.

The maximum value $S(\mathcal{G}, N)$ may attain is 2^N. This is the case when there are N points in $\mathcal{Z}$ such that any sequence of binary numbers may be obtained by applying the functions

in $\mathcal{G}$ to these points. When this happens, we say that $\mathcal{G}$ *shatters* N points from $\mathcal{Z}$. When $S(\mathcal{G}, N)$ is lesser than this bound, it means that, applying only the functions in $\mathcal{G}$, it is not possible to classify some N points of $\mathcal{Z}$ in all possible values N points can be classified. In this case, $\mathcal{G}$ does not shatter N points of $\mathcal{Z}$.

The shattering of $\mathcal{Z}$ is related to the complexity of $\mathcal{G}$. If it shatters N points for N *great*, it means that there are many functions in $\mathcal{G}$ with distinct features, which permit them to classify in many ways N given points. Observe that, for instance,

$$S(\mathcal{G}, N) \leq |\mathcal{G}|,$$

so a great cardinality, with respect to N, of $\mathcal{G}$ is a necessary condition for it to shatter N points of $\mathcal{Z}$. A great cardinality and variability among the functions is what is behind the complexity of a set $\mathcal{G}$ of binary functions.

As an illustrative example, consider two functional hypotheses spaces whose hypotheses domain is $[0, 1]$:

$$\mathcal{H}_1 = \left\{ h(x) = \mathbb{1}\{x \leq p\} \,:\, p \in [0,1] \right\} \bigcup \left\{ h(x) = \mathbb{1}\{x \geq p\} \,:\, p \in [0,1] \right\}$$

$$\mathcal{H}_2 = \left\{ h(x) = \mathbb{1}\{p_1 \leq x \leq p_2\} : p_1, p_2 \in [0,1] \right\} \bigcup \left\{ h(x) = \mathbb{1}\{x \leq p_1 | x \geq p_2\} : p_1, p_2 \in [0,1] \right\}.$$

Observe that $\mathcal{H}_1 \subset \mathcal{H}_2$, as it is enough to take $p_1 = 0$ and $p_2 = p$, or $p_1 = p$ and $p_2 = 1$, to obtain in $\mathcal{H}_2$ a function in $\mathcal{H}_1$ indexed by p. Examples of functions in $\mathcal{H}_1$ and $\mathcal{H}_2$ are presented in Figure 1.7.

While $S(\mathcal{H}_1, N) = S(\mathcal{H}_2, N) = 2^N$ for $N = 1, 2$, we have that $S(\mathcal{H}_1, 3) = 6 < 8 = S(\mathcal{H}_2, 3)$. Indeed, given any three points $x_1 < x_2 < x_3$, there is no function in $\mathcal{H}_1$ which classifies then, respectively, in 010 and 101 since the functions in $\mathcal{H}_1$ are monotone, while these classifications are possible in $\mathcal{H}_2$, by selecting $x_1 < p_1 < x_2 < p_2 < x_3$ and considering the two functions indexed by these parameters (see in Figure 1.7 why this is the case).

Furthermore, we have that $S(\mathcal{H}_2, 4) = 14 < 16$, since the sequences 0101 and 1010 cannot be generated by classifying points $x_1 < x_2 < x_3 < x_4$. The functions in $\mathcal{H}_2$ are capable of changing the value of the next point in the sequence at most twice, and in these sequences the values change three times.

The shatter coefficient seems an intuitive measure of complexity for binary functional hypotheses spaces, which can be extended to include general hypotheses space. This extension, the VC dimension, depends on the choice of loss function, and is as follows.

Definition 1.5 (Vapnik-Chervonenkis dimension)**.** *Fixed a hypotheses space $\mathcal{H}$ and a loss function ℓ, set*

$$C = \sup_{\substack{z \in \mathcal{Z} \\ h \in \mathcal{H}}} \ell(z, h),$$

in which C can be infinity. Consider, for each $h \in \mathcal{H}$ and $\beta \in (0, C)$, the binary function

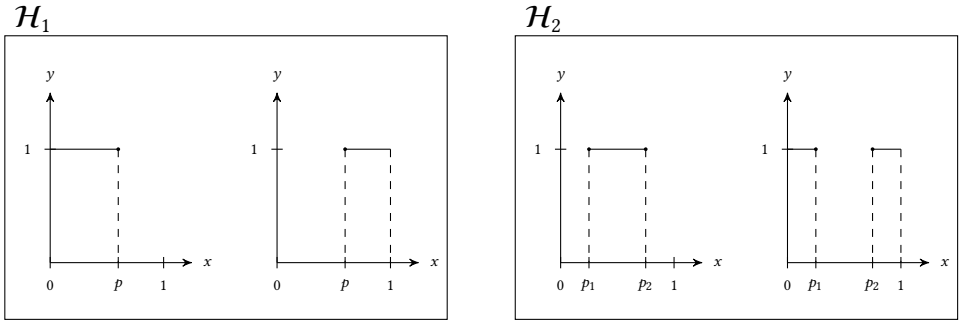

Figure 1.7: *Examples of functions in $\mathcal{H}_1$ and $\mathcal{H}_2$, indexed by parameters p and p_1, p_2, respectively. On the one hand, given any three points $x_1 < x_2 < x_3$, the sequence $h(x_1), h(x_2), h(x_3)$ can change values at most once if $h \in \mathcal{H}_1$, and such a change occurs when $\min\{x_1, x_2, x_3\} < p < \max\{x_1, x_2, x_3\}$, while it can take any value in $\{0,1\}^3$ when considering that $h \in \mathcal{H}_2$. On the other hand, given any four points $x_1 < x_2 < x_3 < x_4$, the sequence $h(x_1), h(x_2), h(x_3), h(x_4)$ can change values at most twice if $h \in \mathcal{H}_2$, and such changes occur when there exists $i \in \{2,3\}$ such that $x_1 < p_1 < x_i < p_2 < x_4$. In this instance, the sequence $0,1,0,1$ cannot be generated by a function in $\mathcal{H}_2$, since the value of $h(x_i)$ changes three times. Another way to see that it is not possible, is to note that the zeros and ones in this sequence should appear in clusters, and there should be at most two clusters of one value (zero or one), and at most one cluster of another. These clusters are represented by the regions $x < p_1, p_1 < x < p_2$ and $p_2 < x$. In the sequence $0,1,0,1$ we have four clusters, two of each value, what is not possible.*

$I(z; h, \beta) = \mathbb{1}\{\ell(z,h) \geq \beta\}$, *for* $z \in \mathcal{Z}$, *and denote*

$$\mathcal{G}_{\mathcal{H},\ell} = \Big\{ I(\cdot; h, \beta) \,:\, h \in \mathcal{H}, \beta \in (0, C) \Big\}.$$

We define the shatter coefficient of $\mathcal{H}$ under loss function ℓ as

$$S(\mathcal{H}, \ell, N) := S(\mathcal{G}_{\mathcal{H},\ell}, N).$$

The Vapnik-Chervonenkis (VC) dimension of $\mathcal{H}$ under loss function ℓ is the greatest integer $k \geq 1$ such that $S(\mathcal{H}, \ell, k) = 2^k$, and is denoted by $d_{VC}(\mathcal{H}, \ell)$. If $S(\mathcal{H}, \ell, k) = 2^k$, for all integer $k \geq 1$, we denote $d_{VC}(\mathcal{H}, \ell) = \infty$.

Remark 1.6. *If there is no confusion about which loss function we are referring, or when it is not of importance to our argument, we omit ℓ and denote the shatter coefficient and VC dimension simply by $S(\mathcal{H}, N)$ and $d_{VC}(\mathcal{H})$. We note that if the hypotheses in $\mathcal{H}$ are binary valued and ℓ is the simple loss function, then $\mathcal{H} = \mathcal{G}_{\mathcal{H},\ell}$, and its N-th shatter coefficient is actually the maximum number of dichotomies that can be generated by the functions in $\mathcal{H}$ with N points.*

Remark 1.7. *The definition of the shatter coefficient of real-valued loss functions arises from the proof of Glivenko-Cantelli theorem [42, Theorem 12.4], that is the basis of classical results in VC theory. We present and discuss this proof in Appendix A to illustrate the origin of the shatter coefficient and VC dimension concepts.*

The N-shatter coefficient of a set $\mathcal{G}$ is the maximum number of dichotomies, i.e., sequences of zeros and ones, which can be obtained by applying the functions in $\mathcal{G}$ to N points in $\mathcal{Z}$. As the upper bound for the number of such dichotomies is 2^N, if $S(\mathcal{G}, N) = 2^N$,

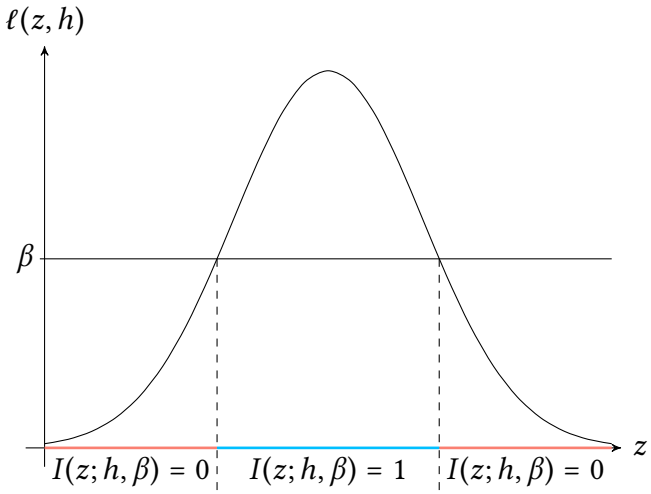

Figure 1.8: *Example of an indicator function $I(\cdot; h, \beta)$ in $\mathcal{G}_{\mathcal{H},\ell}$.*

then $\mathcal{G}$ is complex enough to *shatter* some N points in $\mathcal{Z}$, i.e., it is possible to construct all dichotomies of some N points in $\mathcal{Z}$ by applying functions in $\mathcal{G}$. Hence, the VC dimension is the highest number of points which $\mathcal{G}$ can shatter, so is a measure of the richness of functions in $\mathcal{G}$.

Therefore, the VC dimension of $\mathcal{H}$ is a measure of the complexity of $\mathcal{G}_{\mathcal{H},\ell}$, which represents actually the richness of functions in $\mathcal{H}$ according to loss function ℓ. See in Figure 1.8 an example of indicator function in $\mathcal{G}_{\mathcal{H},\ell}$. If the sign of $\ell(z_1, h) - \ell(z_2, h)$, for general $z_1, z_2 \in \mathcal{Z}$ fixed, varies when one changes $h \in \mathcal{H}$, then $\mathcal{G}_{\mathcal{H},\ell}$ shatters these points. An analogue fact may be established for N points, so for $\mathcal{G}_{\mathcal{H},\ell}$ to shatter them, $\{\ell(z_1, h), \dots, \ell(z_N, h)\}$ should vary as one changes $h \in \mathcal{H}$ in a specific manner. On the other hand, if there is little variation on the form of $\ell_h(z)$ as one changes $h \in \mathcal{H}$, then the functions are *similar* with respect to the value of ℓ, so $\mathcal{H}$ is not as rich.

As an example, consider linear regression in one variable with the quadratic loss function. In this case,

$$\mathcal{G}_{\mathcal{H},\ell} := \left\{ \mathbb{1}\left\{ (y - ax - b)^2 - \beta \geq 0 \right\} : a, b \in \mathbb{R}, \beta \in \mathbb{R}_+ \right\}, \tag{1.10}$$

so each function in it is equal to one if the distance from (x, y) to $(x, ax + b)$ is greater than $\sqrt{\beta}$, and is zero otherwise. Figure 1.9 presents an example of a $I(x, y; h, \beta)$ in this case, in which the points classified as zero are between two lines parallel to h, but with slope differing on $\pm\sqrt{\beta}$. We argue that the VC dimension in this case equals five.

On the one hand, in Figure 1.10 we present five points which can be shattered by the functions in (1.10), presenting three examples of dichotomies to illustrate why this is the case. On the other hand, Figure 1.11 presents dichotomies which cannot be performed with the functions in (1.10), for points which are the vertices of convex and non-convex polygons. Given six points, consider a dichotomy of three zeros and three ones, in which the triangles formed by uniting the points with the same label intersect, as those in Figure 1.11. This is not a possible dichotomy, since a dashed line delimiting the zero region must intersect two sides of the one triangle, what causes at least one of the one triangle vertices

to be labeled as zero. We conclude that the VC dimension is indeed five.

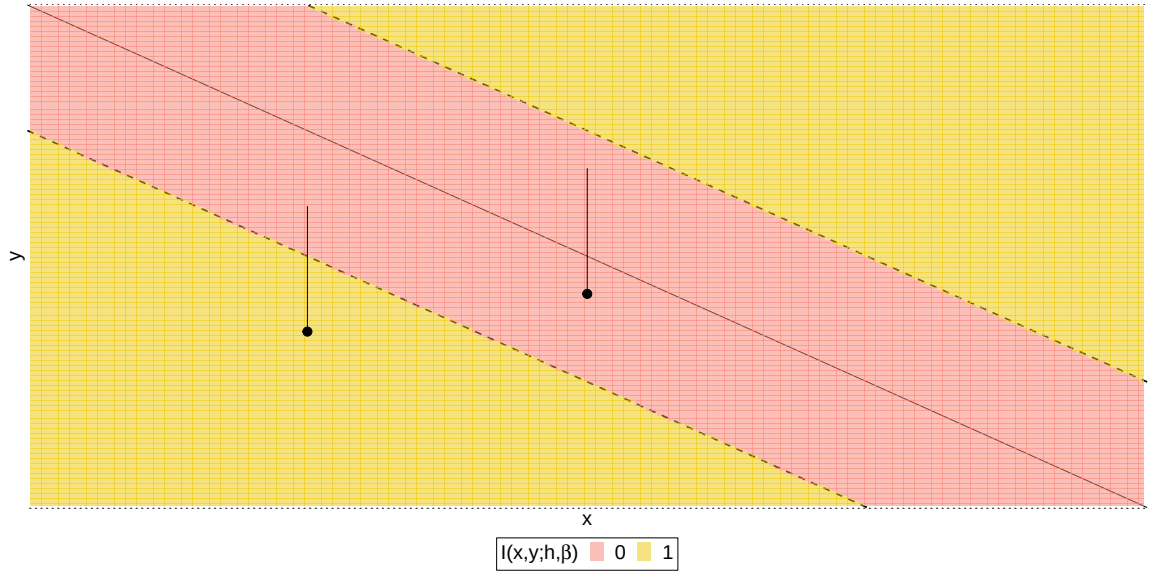

Figure 1.9: *Example of binary function $I(x, y; h, \beta)$, $x, y \in \mathbb{R}$, when $\mathcal{H} = \{h(x) = ax + b : a, b \in \mathbb{R}\}$ contains the linear functions of one variable, and $\ell((x, y), h) = [y - h(x)]^2$ is the quadratic loss function. The vertical lines represent a distance $\sqrt{\beta}$ from the respective point in the direction of $h(x)$, illustrating that points outside the pink region are at a distance greater than that from the line.*

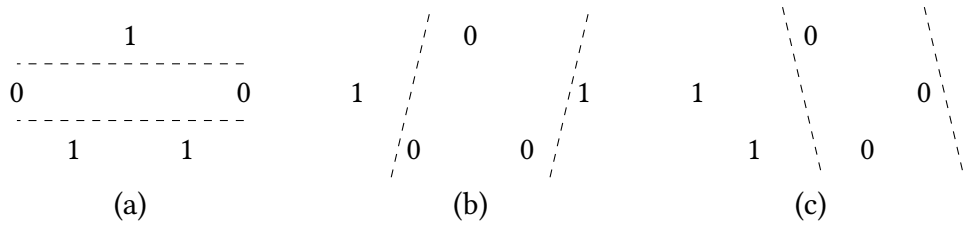

Figure 1.10: *Examples of five point dichotomies which can be generated by functions in $\mathcal{G}_{\mathcal{H},\ell}$ defined in (1.10). The dashed lines are the lines parallel to h, but with a slope differing on $\pm\sqrt{\beta}$, hence the region between them represents the points classified as zero. Set $\mathcal{G}_{\mathcal{H},\ell}$ shatters these five points.*

The complexity measured by VC dimension is more evident when $\mathcal{H}$ is a functional hypotheses space of binary functions, and $\ell((x, y), h) = \mathbb{1}\{h(x) \neq y\}$ is the simple loss function. In this instance, $\mathcal{G}_{\mathcal{H},\ell} = \mathcal{H}$, so the shatter coefficient is actually the number of dichotomies obtained by applying the functions in $\mathcal{H}$, hence it is a measure of the power of the functions in $\mathcal{H}$ in classifying instances of X into the categories of Y. Indeed, in the example of Figure 1.7 we have $d_{VC}(\mathcal{H}_1) = 2$ and $d_{VC}(\mathcal{H}_2) = 3$, so $\mathcal{H}_2$ is more complex than $\mathcal{H}_1$. This fact exemplifies an elementary, but interesting property of the VC dimension that will be explored later, which we prove below.

Lemma 1.8. *If $\mathcal{H}_1 \subset \mathcal{H}_2$, then $d_{VC}(\mathcal{H}_1) \leq d_{VC}(\mathcal{H}_2)$.*

Proof. If $\mathcal{H}_1 \subset \mathcal{H}_2$, then clearly $\mathcal{G}_{\mathcal{H}_1,\ell} \subset \mathcal{G}_{\mathcal{H}_2,\ell}$ so

$$S(\mathcal{G}_{\mathcal{H}_1,\ell}, k) \leq S(\mathcal{G}_{\mathcal{H}_2,\ell}, k),$$

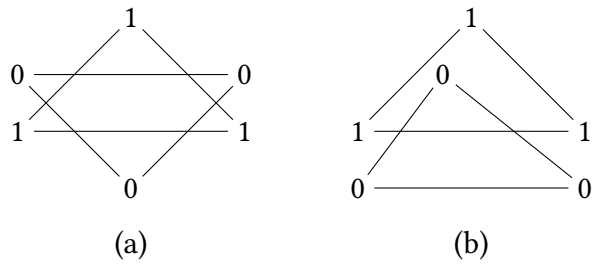

Figure 1.11: *Examples of six point dichotomies which cannot be generated by functions in $\mathcal{G}_{\mathcal{H},\ell}$ defined in (1.10), when the points form (a) convex and (b) non-convex polygons, illustrating that $\mathcal{G}_{\mathcal{H},\ell}$ cannot shatter six points.*

for all $k \geq 1$, from which follows that $S(\mathcal{G}_{\mathcal{H}_1,\ell}, d_{VC}(\mathcal{H}_1)) = 2^{d_{VC}(\mathcal{H}_1)} \leq S(\mathcal{G}_{\mathcal{H}_2,\ell}, d_{VC}(\mathcal{H}_1))$, so we conclude that $d_{VC}(\mathcal{H}_1) \leq d_{VC}(\mathcal{H}_2)$. □

1.2.5 Model error estimation

In order to carry out a Model Selection procedure, once defined the family of candidate models, it is necessary to *estimate* the error of the models. A Model Selection technique is actually an optimizer which seeks to minimize this estimated error among the candidate models, returning either a global minimum, or a suitable suboptimal solution, that is a model which is not a global minimum, but that has an estimated error *small enough*.

The error of a subspace of $\mathcal{H}$ is defined as

$$L(\mathcal{M}) := \min_{h \in \mathcal{M}} L(h) = L(h^\star_{\mathcal{M}}),$$

for $\mathcal{M} \subset \mathcal{H}$. A first idea to estimate $L(\mathcal{M})$ would be to consider the estimator $L_{\mathcal{D}_N}(\hat{h}^{\mathcal{D}_N}_{\mathcal{M}})$, that is, the in-sample error of an ERM hypothesis under $\mathcal{D}_N$. However, even though $\hat{h}^{\mathcal{D}_N}_{\mathcal{M}}$ is a consistent estimator of $h^\star_{\mathcal{M}}$, the resubstitution error $L_{\mathcal{D}_N}(\hat{h}^{\mathcal{D}_N}_{\mathcal{M}})$ is generally an optimistically biased estimator of $L(\mathcal{M})$, specially if the sample size is relatively small [110, Section 2.4]. Furthermore, since $L_{\mathcal{D}_N}(\hat{h}^{\mathcal{D}_N}_{\mathcal{M}_1}) \geq L_{\mathcal{D}_N}(\hat{h}^{\mathcal{D}_N}_{\mathcal{M}_2})$ if $\mathcal{M}_1 \subset \mathcal{M}_2$, selecting models based on this error is susceptible to *overfitting*, as minimizing it leads to the selection of more complex models, which may explain the sample very well, but do not generalize well to non-observed data.

The framework to Model Selection proposed here is not dependent on any specific estimator for $L(\mathcal{M})$, since it may be carried out employing many types of estimators. To illustrate the method, we consider two common estimators for $L(\mathcal{M})$, based on an independent validation sample, and cross-validation, which we define below. Other estimators, for instance based on a Bootstrap technique [51, 52, 53, 81, 110], could also be employed. When there is no need to specify which estimator of $L(\mathcal{M})$ we are referring, we denote simply $\hat{L}(\mathcal{M})$ to mean an arbitrary estimator.

Validation sample

Fix a sequence $\{V_N : N \geq 1\}$ such that $\lim_{N\to\infty} V_N = \lim_{N\to\infty} N - V_N = \infty$, and let

$$\mathcal{D}_N^{(\text{train})} = \{Z_l : 1 \leq l \leq N - V_N\}$$
$$\mathcal{D}_N^{(\text{val})} = \{Z_l : N - V_N < l \leq N\}$$

be a split of $\mathcal{D}_N$ into a training and validation sample. Proceeding in this manner, we have two samples $\mathcal{D}_N^{(\text{train})}$ and $\mathcal{D}_N^{(\text{val})}$, which are independent. The estimator under the validation sample is given by

$$\hat{L}_{\text{val}}(\mathcal{M}) := L_{\mathcal{D}_N^{(\text{val})}}(\hat{h}_{\mathcal{M}}^{(\text{train})}) = \frac{1}{V_N} \sum_{N - V_N < l \leq N} \ell\left(Z_l, \hat{h}_{\mathcal{M}}^{(\text{train})}\right), \tag{1.11}$$

in which

$$\hat{h}_{\mathcal{M}}^{(\text{train})} = \arg\min_{h\in\mathcal{M}} L_{\mathcal{D}_N^{(\text{train})}}(h),$$

is an ERM hypothesis of $\mathcal{M}$ under $\mathcal{D}_N^{(\text{train})}$.

Instances in the validation sample are not in the training data, hence may provide less biased information about the generalization quality of $\hat{h}_{\mathcal{M}}^{(\text{train})}$, i.e., its error when classifying unseen instances. This estimator is specially useful when there is a great sample available, so one can divide it in training and validation samples with great size themselves. However, when there is little data available, a method based on resampling may perform better [103].

K-fold cross-validation

Fix $k \in \mathbb{Z}_+$ and assume $N := kn$, for a $n \in \mathbb{Z}_+$. Then, let

$$\mathcal{D}_N^{(j)} := \{Z_l : (j-1)n < l \leq jn\}, \qquad j = 1, \ldots, k,$$

be a partition of $\mathcal{D}_N$:

$$\mathcal{D}_N = \bigcup_{j=1}^{k} \mathcal{D}_N^{(j)} \qquad \text{and} \qquad \mathcal{D}_N^{(j)} \cap \mathcal{D}_N^{(j')} = \emptyset \text{ if } j \neq j'.$$

We define

$$\hat{h}_{\mathcal{M}}^{(j)} := \arg\min_{h\in\mathcal{M}} L_{\mathcal{D}_N \setminus \mathcal{D}_N^{(j)}}(h) = \arg\min_{h\in\mathcal{M}} \frac{1}{(k-1)n} \sum_{\substack{l \leq (j-1)n \\ \cup\ l > jn}} \ell(Z_l, h)$$

as the ERM hypotheses of the sample $\mathcal{D}_N \setminus \mathcal{D}_N^{(j)}$, that is the sample composed by all folds, but the j-th, and

$$\hat{L}_{\text{cv(k)}}^{(j)}(\mathcal{M}) := L_{\mathcal{D}_N^{(j)}}(\hat{h}_{\mathcal{M}}^{(j)}) = \frac{1}{n} \sum_{(j-1)n < l \leq jn} \ell(Z_l, \hat{h}_{\mathcal{M}}^{(j)}),$$

as the validation error of the j-th fold.

The k-fold cross-validation estimator of $L(\mathcal{M})$ is then given by

$$\hat{L}_{\text{cv(k)}}(\mathcal{M}) := \frac{1}{k} \sum_{j=1}^{k} \hat{L}^{(j)}_{\text{cv(k)}}(\mathcal{M}), \tag{1.12}$$

that is the average validation error over the folds. Estimator (1.12) seeks to diminish the bias of (1.11), by applying a resampling strategy and averaging the validation error over these samples [110, Section 2.5]. Although we focus on the k-fold, other cross validation methods [7, 145] may also be employed in the framework for Model Selection proposed in this thesis.

Remark 1.9. *When there is more than one ERM hypothesis in a model according to a training sample, it is necessary to choose which one of them will be validated. In this thesis, we consider as the validation error the minimum empirical error of the ERM hypotheses under the respective validation sample.*

1.3 Learning Spaces

The choice of candidate models is the most important aspect of Model Selection, and demands a careful understanding about the problem at hand. All prior information about a target hypothesis $h^\star$ should be considered, so the candidate models reflect properties of $h^\star$.

Take, for instance, the classical problem of variable selection. In this case, the hypotheses space is formed by functions with domain in $\mathbb{R}^d$, for example $\mathcal{H} = \{h : \mathbb{R}^d \mapsto \mathbb{R}\}$, and it is assumed that $h^\star$ does not depend on all d coordinates of the input. Variable selection means the selection of the coordinates of the input on which $h^\star$ actually depends.

To solve this problem, one usually considers 2^d candidate models, each containing the hypotheses which depend solely on coordinates in a given subset of coordinates. Looking at these candidate models as partially ordered by inclusion, we see that it is actually a Boolean lattice, having a more complex structure associating the candidate models. This is the canonical example of a structured family of candidate models, and is formally defined in Example 1.11. The Learning Spaces are more general structured families of candidate models, which can be applied to a variety of problems beyond variable selection, depending on the prior information about $h^\star$.

Definition 1.10 (Learning Spaces)**.** *Fix a loss function ℓ and let $\mathcal{H}$ be a general hypotheses space with $d_{VC}(\mathcal{H}, \ell) < \infty$. Let $\mathbb{L}(\mathcal{H}) := \{\mathcal{M}_i : i \in \mathcal{J} \subset \mathbb{Z}_+\}$ be a finite subset of the power set of $\mathcal{H}$, i.e., $\mathbb{L}(\mathcal{H}) \subset \mathcal{P}(\mathcal{H})$ and $|\mathcal{J}| < \infty$. We say that the poset $(\mathbb{L}(\mathcal{H}), \subset)$ is a Learning Space under loss function ℓ if*

(i) $\bigcup_{i \in \mathcal{J}} \mathcal{M}_i = \mathcal{H}$;

(ii) $\mathcal{M}_1, \mathcal{M}_2 \in \mathbb{L}(\mathcal{H})$ *and* $\mathcal{M}_1 \subset \mathcal{M}_2$ *implies* $d_{VC}(\mathcal{M}_1, \ell) < d_{VC}(\mathcal{M}_2, \ell)$.

We define the VC dimension of $\mathbb{L}(\mathcal{H})$ *as*

$$d_{VC}(\mathbb{L}(\mathcal{H}), \ell) := \max_{i \in \mathcal{J}} d_{VC}(\mathcal{M}_i, \ell),$$

for which an upper bound is $d_{VC}(\mathcal{H}, \ell)$.

Remark 1.11. *We omit the dependence on* ℓ *from notation* $\mathbb{L}(\mathcal{H})$*, since it is either clear from the context, or not relevant to our argument. We may also omit it from* $d_{VC}(\mathbb{L}(\mathcal{H}), \ell)$ *denoting simply* $d_{VC}(\mathbb{L}(\mathcal{H}))$.

On the one hand, for $\mathbb{L}(\mathcal{H})$ to be a structuring of $\mathcal{H}$ it should cover $\mathcal{H}$, so the need for (i). On the other hand, condition (ii) implies that any element $\mathcal{M} \in \mathbb{L}(\mathcal{H})$ is maximal, in the sense that there does not exist $\mathcal{M}' \in \mathbb{L}(\mathcal{H})$ such that $d_{VC}(\mathcal{M}') = d_{VC}(\mathcal{M})$ and $\mathcal{M}' \subset \mathcal{M}$, so it guarantees that, if $\mathcal{M}_1 \subset \mathcal{M}_2$, then the complexity of $\mathcal{M}_2$ is greater than that of $\mathcal{M}_1$. By Lemma 1.8, the inequality in (ii) is always lesser or equal, so considering it to be strictly lesser is a constraint.

We note that one could choose $\{\mathcal{M}_1, \dots, \mathcal{M}_n\}$ without thinking of it as a decomposition of a hypotheses space $\mathcal{H}$. Nevertheless, if condition (ii) is satisfied, then it would be a Learning Space of $\mathcal{H} = \bigcup_i \mathcal{M}_i$, so taking $\mathcal{H}$ as this union, the only non-trivial condition is (ii).

As $\mathbb{L}(\mathcal{H})$ covers $\mathcal{H}$, when one searches for a model in $\mathbb{L}(\mathcal{H})$ on which to learn, no hypothesis of $\mathcal{H}$ is lost beforehand, as there is no prior constraint which exclude hypotheses from it. Indeed, all hypotheses in the candidate models, hence all hypotheses in $\mathcal{H}$, are available to be estimated, since it is enough that a model which contains it is selected, and then it is learned on it. A constraint is added to $\mathcal{H}$ a *posteriori*, and based on data, as the method to be proposed seeks to select, based solely on data, the model in $\mathbb{L}(\mathcal{H})$ on which to learn, that can be a constrained subspace $\mathcal{M} \subsetneq \mathcal{H}$.

Although $\mathbb{L}(\mathcal{H})$ is not unique, i.e., there are multiple subsets of $\mathcal{P}(\mathcal{H})$ which are Learning Spaces, there are classes of Learning Spaces that have some properties which enhance the efficiency of Model Selection algorithms (cf. Chapter 3). The main class of Learning Spaces are the Lattice Learning Spaces (see Section B.1 for a definition of complete lattice).

Definition 1.12. *Let* $\mathbb{L}(\mathcal{H})$ *be a Learning Space of* $\mathcal{H}$*. We say that* $\mathbb{L}(\mathcal{H})$ *is a Lattice Learning Space if* $(\mathbb{L}(\mathcal{H}), \subset, \wedge, \vee, \mathcal{O}, \mathcal{I})$ *is a complete lattice, that is a poset with the least* $(\mathcal{O})$ *and greatest* $(\mathcal{I})$ *model, and two operators defined for all subsets of it: the supremum operator* $\vee$ *and the infimum operator* $\wedge$.

In this thesis, we consider only Lattice Learning Spaces, or subsets of one, although the abstract framework is quite general, and may also be applied to other cases.

The main difference between Learning Spaces and usual collections employed as candidate models, is that $\mathbb{L}(\mathcal{H})$ has a richer structure, normally generated by an algebra, as is the case of Lattice Learning Spaces. This structure enables the systematization of Model Selection, and a more efficient search for optimal models, i.e., minimizers of an estimator $\hat{L}$. Indeed, the lack of a rich structure, represented by multi-faceted relations between the elements of a collection, prevents the employment of an algorithm apart from an exhaustive search to obtain optimal, and sometimes suitable suboptimal, solutions.

Therefore, the introduction of a structured collection of candidate models adds to the state-of-the art in Model Selection.

Remark 1.13. *Although we consider the VC dimension, other complexity measures of hypotheses spaces could be used to define the Learning Space (for example, Rademacher complexity [59]). We considered the VC dimension, since it is a property of the hypotheses space and loss function, and does not depend on the unknown distribution P. We note that the value of the VC dimension is not of importance to the algebraic aspect of the Learning Space definition, but only the fact that it increases when we consider nested models. Hence, any other complexity measure such that this increase is also observed for the chosen nested models would generate the same Learning Space.*

1.3.1 Building Learning Spaces

Since a hypotheses space $\mathcal{H}$ has numerous collections of subspaces which satisfy the conditions of a Learning Space, it may not be clear at first how to choose $\mathbb{L}(\mathcal{H})$ for a given practical problem. Nevertheless, suitable Learning Spaces emerge naturally in many important applications, and are built based on a *meaningful* representation of the hypotheses in $\mathcal{H}$, and on prior information about the problem at hand. In this section and the next, we present a general method of building Learning Spaces, and some examples of how they can be applied to important problems.

The first step in building a Learning Space is fixing an algebraic parametric representation of the hypotheses in $\mathcal{H}$. Some important families of learning models have a particular algebraic structure, with a parametric representation, from which a Learning Space can be built.

For example, in regression models, linear classifiers and some functional hypotheses spaces, the parameters represent weights attributed to the coordinates of the input vector, so the parameters are related to variables of the input vector, and each hypothesis is represented by the variables they depend on (non-zero weights) and their weights. This parametric representation leads to a Learning Space suitable for variable selection, and each model in it contains the hypotheses which depend solely on variables in a given subset. This case is formally defined in Example 1.11.

The algebraic structure of $(\mathbb{L}(\mathcal{H}), \subset)$ may be defined from the learning model and algebraic representation fixed. Let $(\mathcal{F}, \leq)$ be a poset, in which $\mathcal{F}$ is an arbitrary set with finite cardinality. Moreover, let $\mathcal{R} : \mathcal{F} \mapsto Im(\mathcal{R}) \subset \mathcal{P}(\mathcal{H})$ be a lattice isomorphism from set $(\mathcal{F}, \leq)$ to $(Im(\mathcal{R}), \subset)$, a subset of the power set of $\mathcal{H}$ partially ordered by inclusion. This means that $\mathcal{R}$ is bijective and if $a, b \in \mathcal{F}, a \leq b$, then $\mathcal{R}(a) \subset \mathcal{R}(b)$, so $\mathcal{R}$ preserves the partial order $\leq$ on $\mathcal{F}$ as the partial order on $Im(\mathcal{R})$ given by inclusion. Then, if

(i) $\bigcup_{a \in \mathcal{F}} \mathcal{R}(a) = \mathcal{H}$ and

(ii) $a, b \in \mathcal{F}, a \leq b, a \neq b$, implies $d_{VC}(\mathcal{R}(a)) < d_{VC}(\mathcal{R}(b))$,

we may define $\mathbb{L}(\mathcal{H}) := Im(\mathcal{R})$ as a Learning Space of $\mathcal{H}$. Isomorphisms which satisfy these conditions play a central role in the theory, and hence we formally define them.

Definition 1.14. *Given a partially ordered set $(\mathcal{F}, \leq)$, a Lattice isomorphism $\mathcal{R} : (\mathcal{F}, \leq) \mapsto (\mathbb{L}(\mathcal{H}), \subset)$, with $\mathbb{L}(\mathcal{H}) \subset \mathcal{P}(\mathcal{H})$, which satisfies (i) and (ii) is called a Learning Space generator.*

A Learning Space is completely defined by a triple $(\mathcal{F}, \leq, \mathcal{R})$, in which the elements of $\mathcal{F}$ may be interpreted as sets of parameters which describe a subset of hypotheses, i.e., the hypotheses in $\mathcal{R}(a)$, $a \in \mathcal{F}$, are represented by the parameters a, so that, in particular, $\mathcal{F}$ generates a parametric representation of the functions in $\mathcal{H}$. For this reason, we call $(\mathcal{F}, \leq)$ a parametric poset of $\mathcal{H}$. Therefore, in general, to build a Learning Space of $\mathcal{H}$, we apply a generator to a parametric poset of its hypotheses. Furthermore, since the generator $\mathcal{R}$ is an isomorphism, it preserves properties of $(\mathcal{F}, \leq)$, hence, for instance, by applying $\mathcal{R}$ to $(\mathcal{F}, \leq, \wedge, \vee, \mathcal{O}, \mathcal{I})$, a complete lattice, we obtain a Lattice Learning Space.

Parametric representations $\mathcal{F}$ are important for, if one can implement a routine to learn hypotheses on hypotheses spaces with representation $\mathcal{F}$, then one can not only learn hypotheses on $\mathcal{H}_1$, but also on $\mathcal{H}_2$, if $\mathcal{F}$ is a parametric representation of both. In this case, there will be a bijection between $\mathbb{L}(\mathcal{H}_1)$ and $\mathbb{L}(\mathcal{H}_2)$, and Model Selection routines implemented for one of them, should be easily modified to the other. Therefore, identifying general parametric representations is an important step to obtain multipurpose algorithms for Model Selection.

1.3.2 Examples of Learning Spaces

We present some examples of Learning Spaces which may be obtained by applying a generator to a parametric poset $(\mathcal{F}, \leq)$.

Example 1.10 (Maximum Likelihood)**.** Recall the Maximum Likelihood framework: let $\mathcal{H} \subset \mathbb{R}^d, d \geq 1$, $\mathcal{Z} \subset \mathbb{R}^{d_Z}, d_Z \geq 1$, and $f(\cdot|\theta) : \mathcal{Z} \mapsto \mathbb{R}_+$ be a probability function or probability density function, for each $\theta \in \mathcal{H}$.

Assume that the VC dimension is an increasing function of the number of free parameters. For example, $\mathcal{M}_1 = \{\theta \in \mathcal{H} : \theta_1 = 0\}$ has $d - 1$ free parameters, while $\mathcal{M}_2 = \{\theta \in \mathcal{H} : \theta_i = a_i : 1 \leq i \leq d'\}$, for a sequence of real numbers $a_1, \ldots, a_{d'}$, has $d - d'$ free parameters, so we should have $d_{VC}(\mathcal{M}_1) > d_{VC}(\mathcal{M}_2)$. In this framework, a Learning Space would be generated by a collection of *statistical hypotheses* of the form $\{\theta_i = a_i : i \in A \subset \{0, \ldots, d\}\}$, but rather than testing these hypotheses via statistical significance, we would be selecting a model $\mathcal{M}_i$ which would represent the *most suitable statistical hypothesis*, in some sense. For instance, if we take $a_i = 0, \forall i \in A$, we may have a special case of a Variable Selection Learning Space. Moreover, there may be a relation between such a Learning Space and likelihood ratio tests [136], depending on how one chooses the loss function and model error estimator. We leave this relation as a topic for future researches. ∎

Example 1.11 (Variable selection)**.** Let $\mathcal{H}$ be a functional hypotheses space, with domain $\mathcal{X} \subset \mathbb{R}^d, d > 1$, and image $\mathcal{Y} \subset \mathbb{R}$. Let $\mathcal{F} = \mathcal{P}(\{1, \ldots, d\})$ be the powerset of $\{1, \ldots, d\}$, partially ordered by inclusion, so that $(\mathcal{F}, \subset, \cap, \cup, \emptyset, \{1, \ldots, d\})$ is a Boolean lattice. Consider the Learning Space generator $\mathcal{R} : \mathcal{F} \mapsto Im(\mathcal{R}) \subset \mathcal{P}(\mathcal{H})$ given by

$$\mathcal{R}(a) = \left\{ h \in \mathcal{H} : h(x) = h(x'), \text{ if } x \equiv_a x' \right\},$$

in which $a = \{a_1, \dots, a_j\} \in \mathcal{F}$, and $x = (x_1, \dots, x_d) \equiv_a x' = (x'_1, \dots, x'_d)$ if, and only if, $x_{a_i} = x'_{a_i}$ for $i = 1, \dots, j$, so $\mathcal{R}(a)$ contains the hypotheses which depend solely on variables in a.

The lattice isomorphism $\mathcal{R}$ satisfies condition (i), and often satisfies (ii), as in many applications the VC dimension is an increasing function of the number of variables, so $Im(\mathcal{R})$ is often a Learning Space. See Figure 1.12 for an example of a parametric lattice for variable selection. ■

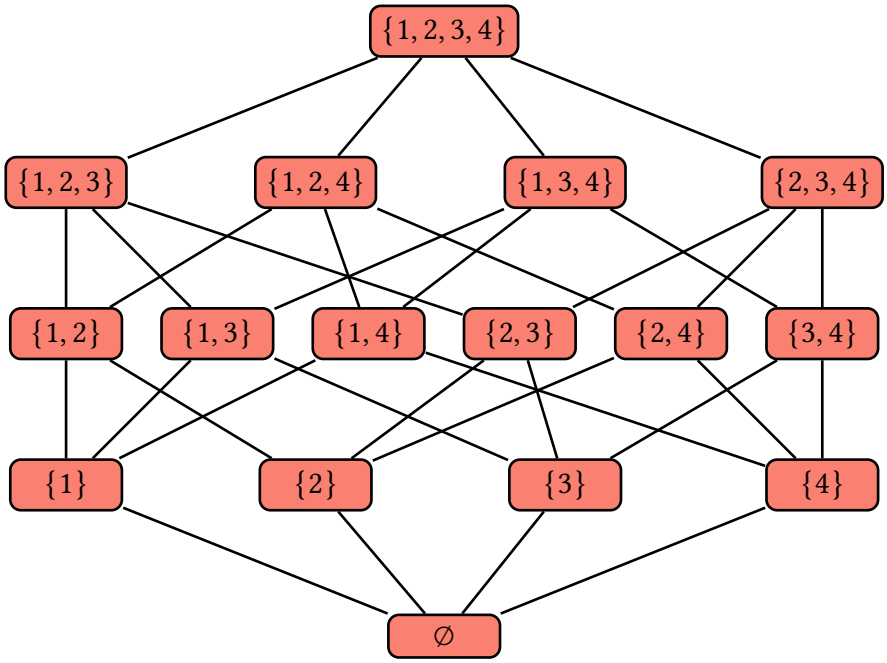

Figure 1.12: *Parametric lattice for variable selection when* $d = 4$.

Example 1.12 (Partition lattice)**.** Let $\mathcal{H} = \{h : \mathcal{X} \mapsto \{0, 1\}\}$ be the hypotheses space of all functions with domain $\mathcal{X}, |\mathcal{X}| < \infty$, and image $\{0, 1\}$, under the simple loss function. In this case, a target hypothesis $h^\star$ creates an equivalence class in $\mathcal{X}$, partitioning it according to its classification of each input value, generating an ordered partition $\mathcal{X}_0, \mathcal{X}_1$, with $\mathcal{X}_i = \{x \in \mathcal{X} : h^\star(x) = i\}, i = 0, 1$, in which each element is in exactly one of these sets, the one which represents its classification according to $h^\star$. Actually, every hypothesis $h \in \mathcal{H}$ generates an ordered partition according to its classification of the input values. Hence, there is a duality between hypotheses and the ordered partitions of $\mathcal{X}$ with two parts.

The partition-hypothesis duality brings upon the paradigm of learning a hypothesis through a partition. This task can be performed in two manners, either by first choosing explicitly an unordered partition and learning an ordination of it, which generates a hypothesis, or learning the partition implicitly while learning the hypothesis.

In both manners, it is not necessary to have a partition consisting of exactly two parts, as one could rather consider more sets that form a partition, with the constraint that elements in a same set should be classified in the same output. These sets represent an equivalence relation on $\mathcal{X}$, with equivalence between elements in a same set. This may ease the estimation process, since the original partition sets are broken into more simple ones, which may have better topological features, and be easier to estimate.

Assume we know a partition $\mathcal{X}_1, \ldots, \mathcal{X}_k$ of k greater than two parts, such that there is a hypothesis which respects it that well-approximates a target one. The set of hypotheses that respect a partition is composed by the ones that classify elements in a same part in the same output. Once we fix a partition $\mathcal{X}_1, \ldots, \mathcal{X}_k$, the learning is performed considering only hypotheses that respect it, that is a constrained hypotheses space, on which the learning may be *better* than on all of $\mathcal{H}$, since this constrained space (a) has lesser VC dimension (that is equal to k, as we will show), and (b) contains a hypothesis that *well* approximates a target one.

The hypotheses that respect each partition form the models in the Partition Lattice Learning Space (cf. Figure 1.13), so this Learning Space is a natural family of candidate models under the partition-hypothesis duality when $|\mathcal{X}|$ is finite. We now formally define this Learning Space.

Let $\mathcal{X}$ be such that $|\mathcal{X}| < \infty$, and let $\mathcal{H}$ contain all functions with domain $\mathcal{X}$ and image $\{0, 1\}$. A partition of $\mathcal{X}$ is a set π of non-empty subsets of $\mathcal{X}$, called blocks or parts, such that every element $x \in \mathcal{X}$ is in exactly one of these blocks. A partition π generates an equivalence relation on $\mathcal{X}$, in the sense that x and z in $\mathcal{X}$ are π-equivalent, i.e., $x \equiv_\pi z$, if, and only if, they are in the same block of partition π.

Define $\mathcal{F} := \{\pi : \pi \text{ is a partition of } \mathcal{X}\}$ as the set of all partitions of $\mathcal{X}$, partially ordered by $\leq$ defined as

$$\pi_1 \leq \pi_2 \text{ if, and only if, } x \equiv_{\pi_2} z \text{ implies } x \equiv_{\pi_1} z, \tag{1.13}$$

for $\pi_1, \pi_2 \in \mathcal{F}$, which is a complete lattice $(\mathcal{F}, \leq, \wedge, \vee, \{\mathcal{X}\}, \mathcal{X})$. Relation (1.13) is equivalent to: for every block $a \in \pi_2$, there exists a block $b \in \pi_1$ such that $a \subset b$. See Figure 1.13 for an example of Partition Lattice.

Consider $\mathcal{R} : \mathcal{F} \mapsto Im(\mathcal{R}) \subset \mathcal{P}(\mathcal{H})$ given by

$$\mathcal{R}(\pi) := \mathcal{H}|_\pi = \left\{ h \in \mathcal{H} : h(x) = h(z) \text{ if } x \equiv_\pi z \right\},$$

for $\pi \in \mathcal{F}$. The set $\mathcal{R}(\pi)$ is formed by all hypotheses which classify the points inside a block of π in a same category, that are the hypotheses which respect π. We show that $\mathcal{R}$ is a Learning Space generator, so that $\mathbb{L}(\mathcal{H}) := Im(\mathcal{R})$ is a Learning Space.

Proposition 1.15. *The function $\mathcal{R} : \mathcal{F} \mapsto Im(\mathcal{R}) \subseteq \mathcal{P}(\mathcal{H})$ given by $\mathcal{R}(\pi) := \mathcal{H}|_\pi, \pi \in \mathcal{F}$, is an isomorphism which satisfies conditions (i) and (ii).*

Proof. It is enough to show that a) $\pi_1 \leq \pi_2$ if, and only if, $H|_{\pi_1} \subseteq H|_{\pi_2}$; b) $d_{VC}(H|_\pi) = |\pi|$; and c) $\pi_1 \leq \pi_2$ implies $|\pi_1| < |\pi_2|$, as we have that $\mathcal{R}(\mathcal{X}) = \mathcal{H}$ and $\mathcal{X} \in \mathcal{F}$, as $\mathcal{X}$ is a partition of itself, in which each block is formed by a singleton, so condition (i) of Learning Spaces is satisfied.

a) On the one hand, if $\pi_1 \leq \pi_2$, then

$$\left\{ h \in \mathcal{H} : h(x) = h(y) \text{ if } x \equiv_{\pi_2} y \right\} \supseteq \left\{ h \in \mathcal{H} : h(x) = h(y) \text{ if } x \equiv_{\pi_1} y \right\},$$

as $x \equiv_{\pi_2} y$ implies $x \equiv_{\pi_1} y$ by definition of $\leq$, so that $\mathcal{H}|_{\pi_2} \supseteq \mathcal{H}|_{\pi_1}$. On the other hand, assume that $\mathcal{H}|_{\pi_1} \subseteq \mathcal{H}|_{\pi_2}$. Then,

$$\left\{ h \in \mathcal{H} \,:\, h(x) = h(y) \text{ if } x \equiv_{\pi_1} y \right\} \subseteq \left\{ h \in \mathcal{H} \,:\, h(x) = h(y) \text{ if } x \equiv_{\pi_2} y \right\},$$

so that $x \equiv_{\pi_2} y$ implies $x \equiv_{\pi_1} y$, and $\pi_1 \leq \pi_2$. Indeed, otherwise, there would be $x, y \in \mathcal{X}$, such that $x \equiv_{\pi_2} y$ and $x \not\equiv_{\pi_1} y$, what would imply that there is a function in $\mathcal{H}|_{\pi_1}$ that is not in $\mathcal{H}|_{\pi_2}$: any function such that $h(x) \neq h(y)$.

b) We have that $\mathcal{S}(H|_\pi, N) = 2^N$ if $N \leq |\pi|$ as, if $x_i \not\equiv_\pi x_j$, $1 \leq i < j \leq N$, then

$$\left| \left\{ \left(h(x_1), \dots, h(x_N) \right) \,:\, h \in H|_\pi \right\} \right| = 2^N,$$

as it is possible to classify $x_1, \dots, x_N$ *freely* in $\{0,1\}$ applying the functions in $H|_\pi$, as the x_i are pairwise non π-equivalent. On the other hand, if $N > |\pi|$ then in every sequence $x_1, \dots, x_N$ there exists at least one pair such that $x_i \equiv_\pi x_j$. Assume, without loss of generality, that $x_1 \equiv_\pi x_2$. Then, since all vectors such that $h(x_1) \neq h(x_2)$, i.e., starting in 01 or 10, are not in $\left\{ \left(h(x_1), \dots, h(x_N) \right) \,:\, h \in H|_\pi \right\}$, we have that $\left\{ \left(h(x_1), \dots, h(x_N) \right) \,:\, h \in H|_\pi \right\} \subsetneq \{0,1\}^N$, and

$$\left| \left\{ \left(h(x_1), \dots, h(x_N) \right) \,:\, h \in H|_\pi \right\} \right| < \left| \{0,1\}^N \right| = 2^N,$$

which establishes that $d_{VC}(H|_\pi) = |\pi|$.

c) If $\pi_1 \leq \pi_2$, then every block of π_2 is contained in a block of π_1, as it is a necessary condition for π_2-equivalence to imply π_1-equivalence. Then, it follows that $|\pi_1| \leq |\pi_2|$. However, $|\pi_1| \neq |\pi_2|$, for otherwise we would have $\pi_1 = \pi_2$ in order for the inclusion of blocks property to be satisfied. □

In the Partition Lattice Learning Space construction, the parameters of the hypotheses $h \in \mathcal{H}$ are the elements in their domain $\mathcal{X}$, in contrast, for example, to the variables they depend on as in Example 1.11. If $\mathcal{H}$ is given by the set of Boolean functions $h : \{0,1\}^d \mapsto \{0,1\}$, we have the special case of a Boolean partition lattice, which is studied in [32].

This Learning Space is quite general and has subsets which are themselves useful Learning Spaces, illustrating how one may drop subsets of $\mathbb{L}(\mathcal{H})$ to obtain other Learning Spaces according to the needs of the application at hand. For example, the Variable Selection Learning Space, when $\mathcal{X} \subset \mathbb{R}^d, |\mathcal{X}| < \infty$, is a sub-lattice of the Partition Lattice Learning Space, and is represented in orange in Figure 1.13, when $\mathcal{X} = \{0,1\}^2$. This Learning Space can also be obtained by applying the generator of Example 1.11.

Apart from dropping nodes, one can also obtain Learning Spaces for subsets $\mathcal{M} \subset \mathcal{H}$, by taking $\mathbb{L}(\mathcal{M}) = \mathbb{L}(\mathcal{H}) \cap \mathcal{M} := \{\mathcal{M}' \cap \mathcal{M} \,:\, \mathcal{M}' \in \mathbb{L}(\mathcal{H})\}$, which is often a Learning Space of $\mathcal{M}$. For example, by taking $\mathcal{M}$ as the non-decreasing hypotheses in $\mathcal{H}$ when $\mathcal{X} \subset \mathbb{R}$, i.e., $\mathcal{M} = \{h \in \mathcal{H} \,:\, h(x_1) \leq h(x_2), \text{ if } x_1 < x_2\}$, we obtain $\mathbb{L}(\mathcal{M})$ composed by the dashed nodes in Figure 1.13, whose hypotheses are the ones in $\mathcal{R}(\pi) \cap \mathcal{M}$. Some nodes may be dropped in order for $\mathbb{L}(\mathcal{H}) \cap \mathcal{M}$ to satisfy (ii). In Section 4.4, we introduce the learning of interval Boolean functions, that are quite important for image classification, which can be performed on a subset of the Boolean partition lattice.

These are some examples of how one can incorporate prior knowledge about the problem at hand into the Learning Space. If one believes that a target hypothesis does not depend on all variables, he may use the Variable Selection Learning Space, while if one believes a target hypothesis is non-decreasing, he may consider the respective subset of the Partition Lattice Learning Space. In both cases, the Learning Space is defined by constraining the Partition Lattice Learning Space according to prior information, so multipurpose learning algorithms based on the Partition Lattice may be applied on several instances, when distinct levels of prior information is available. ■

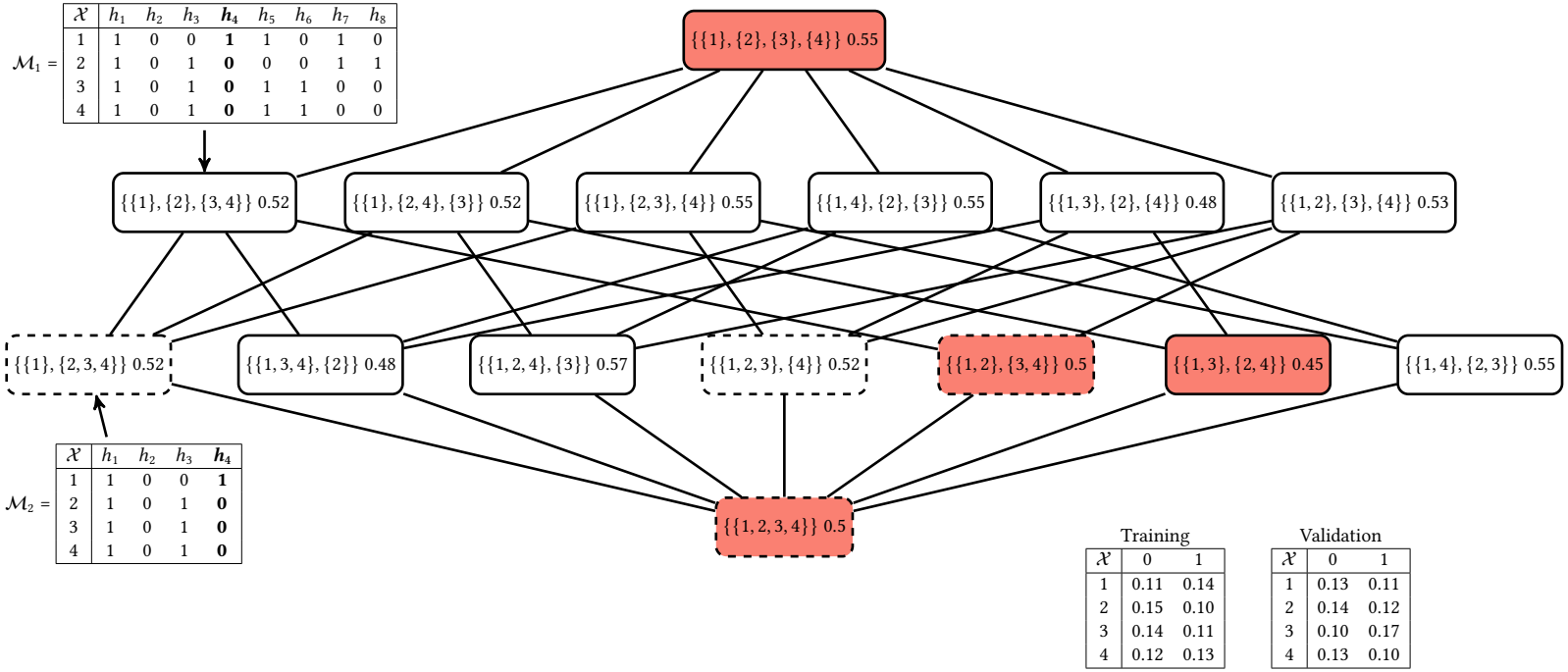

Figure 1.13: *Partition Lattice for Linear Classifiers with $d = 4$ or for $\mathcal{X} = \{1, 2, 3, 4\}$. The tables present the hypotheses in selected models $\mathcal{M}_1, \mathcal{M}_2$ of the Partition Lattice Learning Space for $\mathcal{X} = \{1, 2, 3, 4\}$. The orange nodes represent the Boolean lattice of variable selection when $\mathcal{X} = \{0, 1\}^2$, so its points are $1 = (0, 0), 2 = (0, 1), 3 = (1, 0)$ and $4 = (1, 1)$. The dashed nodes are the ones in $\mathbb{L}(\mathcal{H}) \cap \mathcal{M}$, in which $\mathcal{M}$ is composed by the non-decreasing hypotheses. We present an example of joint empirical frequencies observed in a training and validation sample. The number in each node represents its estimated error calculated as (1.11), by first estimating a hypothesis via ERM with the training sample, and then calculating its error on the validation sample. The bold hypotheses in each table represent the ERM hypothesis of the respective model. When there is more than one ERM hypothesis in a model, we consider the minimum validation error among them as its estimated error.*

Example 1.13 (Linear Classifiers)**.** Let $\mathcal{H}$ be given by the linear classifiers in $\mathbb{R}^d$, $d \geq 1$:

$$\mathcal{H} = \left\{ h_a(x) = \frac{1}{2}\operatorname{sgn}\left\{ a_0 + \sum_{i=1}^{d} a_i x_i \right\} + \frac{1}{2} \ : \ a_i \in \mathbb{R} \right\},$$

in which $x = (x_1, \dots, x_d) \in \mathbb{R}^d$, and h_a is the function indexed by its parameters $a = (a_0, \dots, a_d) \in \mathbb{R}^{d+1}$.

Denoting $\mathcal{A} = \{1, \dots, d\}$, we consider two distinct Learning Spaces generators: from the Boolean lattice $(\mathcal{P}(\mathcal{A}), \subset, \cap, \cup, \emptyset, \mathcal{A})$ and from the Partition Lattice $(\Pi_{\mathcal{A}}, \leq, \wedge, \vee, \{\mathcal{A}\}, \mathcal{A})$ of $\mathcal{A}$, in which $\mathcal{P}(\mathcal{A})$ is the power set of $\mathcal{A}$ and $\Pi_{\mathcal{A}}$ is the set of all partitions of $\mathcal{A}$. The Boolean lattice is represented in Figure 1.12, and the partition lattice in Figure 1.13 for $d = 4$.

Define $\mathcal{R}_1 \ : \ \mathcal{P}(\mathcal{A}) \mapsto \mathcal{P}(\mathcal{H})$ as

$$\mathcal{R}_1(A) = \left\{ h_a \in \mathcal{H} \ : \ a_j = 0 \text{ if } j \notin A \cup \{0\} \right\},$$

for $A \in \mathcal{P}(\mathcal{A})$, as a variable selection generator, and define $\mathcal{R}_2 \ : \ \Pi_{\mathcal{A}} \mapsto \mathcal{P}(\mathcal{H})$ as

$$\mathcal{R}_2(\pi) = \left\{ h_a \in \mathcal{H} \ : \ a_j = a_k \text{ if } j \equiv_\pi k \right\},$$

for $\pi \in \Pi_{\mathcal{A}}$, as a generator which equal parameters, i.e., create equivalence classes in $\mathcal{A}$. Both $\mathcal{R}_1, \mathcal{R}_2$ clearly satisfies (i), and

$$d_{VC}(\mathcal{R}_1(A)) = |A| + 1 \qquad\qquad d_{VC}(\mathcal{R}_2(\pi)) = |\pi| + 1,$$

so they also satisfy (ii).

Therefore, these lattice isomorphisms generate two distinct Lattice Learning Spaces for a same hypotheses space $\mathcal{H}$, and the application at hand will dictate which one is the more suitable to solve the problem. For example, if it is believed that $h^\star$ does not depend on all variables, then the Boolean Lattice Learning Space may be preferable; otherwise, one would rather choose the Partition Lattice Learning Space, a subset of it, or the intersection of it with a subset $\mathcal{M} \subset \mathcal{H}$, if one believes that the target linear classifier satisfies some specific property. ■

Example 1.14 (Deep neural networks)**.** Recall the definition of a DNN in Example 1.6, which are hypotheses spaces generated by a class

$$\mathcal{A} = \left\{ \{f_0^{\theta_0}, \dots, f_{m+1}^{\theta_{m+1}}\} \ : \ \theta \in \Theta \right\}$$

satisfying (1.6), i.e., an architecture $\mathcal{A}$, with hypotheses of the form

$$h_\theta(x) := f_{m+1}^{\theta_{m+1}} \circ f_m^{\theta_m} \circ \cdots \circ f_0^{\theta_0}(x),$$

for $x \in \mathcal{X}$ and $\theta \in \Theta$. In summary, the hypotheses space generated by architecture $\mathcal{A}$ is

$$\mathcal{R}(\mathcal{A}) := \left\{ h_\theta \, : \, \theta \in \Theta \right\}.$$

Since an architecture $\mathcal{A}$ generates a hypotheses space $\mathcal{H} := \mathcal{R}(\mathcal{A})$, the selection of an architecture, in what is known in the literature as Neural Architecture Search [54], could be performed via the selection of a model among the candidates in $\mathbb{L}(\mathcal{H})$, generated by distinct DNN architectures. The first step in this task is to define $\mathbb{L}(\mathcal{H})$, which would be as follows.

Let $\mathscr{A} = \{\mathcal{A}_1, \dots, \mathcal{A}_n\}$ be a collection of architectures, and $(\mathscr{A}, \leq)$ a poset, such that $\mathcal{H} = \bigcup_{i=1}^{n} \mathcal{R}(\mathcal{A}_i)$ and

$$\mathcal{A}_i \leq \mathcal{A}_j, i \neq j \implies \mathcal{R}(\mathcal{A}_i) \subset \mathcal{R}(\mathcal{A}_j) \text{ and } d_{VC}(\mathcal{R}(\mathcal{A}_i)) < d_{VC}(\mathcal{R}(\mathcal{A}_j)), \tag{1.14}$$

so that $\mathbb{L}(\mathcal{H}) := \{\mathcal{R}(\mathcal{A}_i) \, : \, i = 1, \dots, n\}$ is a Learning Space of $\mathcal{H}$. In this framework, selecting a subspace $\mathcal{M}_i := \mathcal{R}(\mathcal{A}_i)$ from $\mathbb{L}(\mathcal{H})$ would imply selecting an architecture $\mathcal{A}_i$ from $\mathscr{A}$.

Although easy to define, the construction of a Learning Space for DNNs is not a straightforward task. On the one hand, it is not clear which classifiers are in $\mathcal{R}(\mathcal{A}_i)$ for a given architecture $\mathcal{A}_i$. On the other hand, since there is a lot of redundancy among the parameters θ [41, 111], one usually does not know if a constraint in the parametric space Θ, e.g., by setting some parameters to zero, actually generates a constraint in the hypotheses space $\mathcal{H}$. Furthermore, the VC dimension of $\mathcal{R}(\mathcal{A}_i)$ is often not exactly known, and is not necessarily proportional to either the dimensionality of θ, that is the architecture number of parameters, or the number of layers [17, 65, 79, 144].

As opposite, in the examples above it is clear that constraints in the parametric space, such as dropping variables (variable selection and linear classifiers), equating parameters (liner classifiers) and creating equivalence classes on the classifier domain (partition lattice), not only represent constraints in the hypotheses space, but are also isomorphisms, which preserve the partial ordering on the parametric poset and satisfy the Learning Space property regarding the VC dimension of related models.

Nevertheless, a Learning Space for DNNs could be built from the top-down or from the bottom-up. The former would be performed by starting from the greatest architecture $\mathcal{A}$, and then adding constraints to its parametric space Θ successively, in various ways and ordinations, generating the chains of $\mathbb{L}(\mathcal{H})$ from the top. The latter would mean starting from concise building blocks, e.g., coordinates of layers, full layers or compositions of layers $f_j^{\theta_j}$, which can be combined in many ways to form more complex architectures, creating chains of $\mathbb{L}(\mathcal{H})$ from the bottom. This task could be accomplished with or without explicitly knowing the subspace $\mathcal{R}(\mathcal{A}_i)$ for all i, since it is enough to guarantee that (1.14) is in force. ■

1.3.3 Minimums of Learning Spaces

Model Selection via Learning Spaces relies on the concept of minimums of a $\mathbb{L}(\mathcal{H})$, since it will select a minimizer of error $\hat{L}$, that is a global minimum, and also a local minimum, of $\mathbb{L}(\mathcal{H})$. We start with the definition of continuous chain, which are basically chains with no *jumps* over models in $\mathbb{L}(\mathcal{H})$. Its definition is illustrated in Figure 1.14. In what follows, $d(\cdot,\cdot)$ means distance in the directed acyclic graph correspondent to $(\mathbb{L}(\mathcal{H}),\subset)$ (cf. Appendix B).

Definition 1.16. *A sequence $\mathcal{M}_{i_1} \subset \mathcal{M}_{i_2} \subset \cdots \subset \mathcal{M}_{i_k}$ is called a continuous chain of $\mathbb{L}(\mathcal{H})$ if, and only if, $d(\mathcal{M}_{i_j}, \mathcal{M}_{i_{j+1}}) = 1$ for all $j \in \{1,\ldots,k-1\}$.*

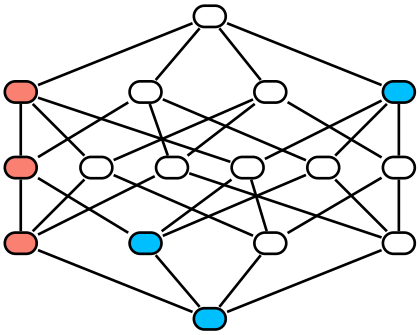

Figure 1.14: *Examples of a continuous chain (orange) and a chain that is not continuous (blue) within a Boolean Learning Space. Observe that there is no model in the Learning Space between two subsequent models of the orange chain, while in the blue chain there are two models between the second and third (from bottom to top) models of it.*

We now define the minimums of $\mathbb{L}(\mathcal{H})$.

Definition 1.17. *The model $\mathcal{M}_{i_\star}$ is:*

- *a **weak local minimum** of a continuous chain $\mathcal{M}_{i_1} \subset \mathcal{M}_{i_2} \subset \cdots \subset \mathcal{M}_{i_k}$ of $\mathbb{L}(\mathcal{H})$ if*

$$\hat{L}(\mathcal{M}_{i_{j_\star}}) \leq \min\left(\hat{L}(\mathcal{M}_{i_{j_\star-1}}), \hat{L}(\mathcal{M}_{i_{j_\star+1}})\right),$$

  in which $\hat{L}(\mathcal{M}_{i_0}) \equiv \hat{L}(\mathcal{M}_{i_{k+1}}) \equiv +\infty$;

- *a **strong local minimum** of $\mathbb{L}(\mathcal{H})$ if it is a weak local minimum of all continuous chains of $\mathbb{L}(\mathcal{H})$ which contain it, that is,*

$$\hat{L}(\mathcal{M}_{i_{j_\star}}) \leq \min\left\{\hat{L}(\mathcal{M}) \,:\, \mathcal{M} \in \mathbb{L}(\mathcal{H}), d(\mathcal{M}_{i_{j_\star}}, \mathcal{M}) = 1\right\};$$

- *a **sup-strong local minimum** of $\mathbb{L}(\mathcal{H})$ if*

$$\hat{L}(\mathcal{M}_{i_{j_\star}}) \leq \min\left\{\hat{L}(\mathcal{M}) \,:\, \mathcal{M} \in \mathbb{L}(\mathcal{H}), \mathcal{M}_{i_{j_\star}} \subset \mathcal{M}, d(\mathcal{M}_{i_{j_\star}}, \mathcal{M}) = 1\right\};$$

- *a **inf-strong local minimum** of $\mathbb{L}(\mathcal{H})$ if*

$$\hat{L}(\mathcal{M}_{i_{j_\star}}) \leq \min\left\{\hat{L}(\mathcal{M}) \,:\, \mathcal{M} \in \mathbb{L}(\mathcal{H}), \mathcal{M} \subset \mathcal{M}_{i_{j_\star}}, d(\mathcal{M}_{i_{j_\star}}, \mathcal{M}) = 1\right\};$$

- *a* ***global minimum of a continuous chain*** *if* $\hat{L}(\mathcal{M}_{i_{j_\star}}) = \min_{1 \le j \le k} \hat{L}(\mathcal{M}_{i_j})$;

- *a* ***global minimum*** *of* $\mathbb{L}(\mathcal{H})$ *if* $\hat{L}(\mathcal{M}_{i_{j_\star}}) = \min_{i \in \mathcal{J}} \hat{L}(\mathcal{M}_i)$.

In Definition 1.17, the error of a model $\mathcal{M}$ is estimated by a fixed estimator $\hat{L}(\mathcal{M})$, so the concept of minimums is dependent upon the choice of estimator. Since $\mathcal{M}_{i_j} \subset \mathcal{M}_{i_{j+1}}, j = 1, \dots, k-1$, the sequence $\{L_{D_N}(\hat{h}_{i_j}^{D_N}) : j = 1, \dots, k\}$ is non-increasing, hence if we estimated this error simply by the resubstitution $L_{D_N}(\hat{h}_{\mathcal{M}}^{D_N})$, the minimum would always be the greatest model of the chain, and these definitions would be meaningless.

On the other hand, employing an estimator involving validation samples, for example, as those in Section 1.2.5, allow the minimums to be within a continuous chain, hence be meaningful. See Figure 1.15 for a depiction of a strong and sup-strong local minimum. We note that a strong local minimum is both an inf and sup-strong local minimum.

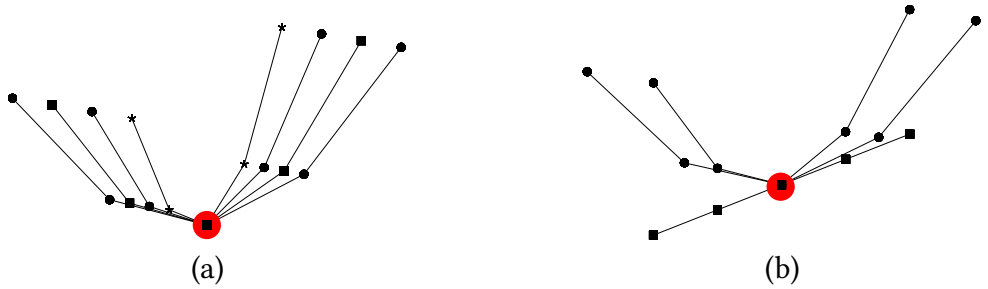

Figure 1.15: *Example of a (a) strong local minimum, that is a local minimum of all chains which contain it, and (b) sup-strong local minimum, that is a model with error lesser or equal to all models at a distance one from it that are greater. Observe that (b) is also a weak local minimum of four chains that pass through it.*

The concepts of local minimums of a Learning Space are essential to develop U-curve algorithms, and will be further discussed in Chapter 3.

1.4 Target model and main objective

The concept of target model is central in our approach. As is the case in all optimization problems, there must be an optimal solution to the Model Selection problem, which satisfies certain desired conditions. In the proposed framework, the optimal is the model in $\mathbb{L}(\mathcal{H})$ with the least VC dimension which contains a target hypothesis. To be exact in this definition, we need to consider equivalence classes of models, as it is not possible to differentiate some models with the concepts of the theory.

Define in $\mathbb{L}(\mathcal{H})$ the equivalence relation given by

$$\mathcal{M}_i \sim \mathcal{M}_j \text{ if, and only if, } d_{VC}(\mathcal{M}_i) = d_{VC}(\mathcal{M}_j) \text{ and } L(\mathcal{M}_i) = L(\mathcal{M}_j), \tag{1.15}$$

for $\mathcal{M}_i, \mathcal{M}_j \in \mathbb{L}(\mathcal{H})$: two models in $\mathbb{L}(\mathcal{H})$ are equivalent if they have the same VC dimension and error. Let

$$\mathcal{L}^\star = \arg\min_{\mathcal{M} \in \mathbb{L}(\mathcal{H})/\sim} L(\mathcal{M})$$

be the equivalence classes which contain a target hypothesis of $\mathcal{H}$, so their error is minimum. We define the target model $\mathcal{M}^\star \in {}^{\mathbb{L}(\mathcal{H})}\!/_{\sim}$ as

$$\mathcal{M}^\star = \arg\min_{\mathcal{M}\in\mathcal{L}^\star} d_{VC}(\mathcal{M}),$$

which is the class of the smallest models in $\mathbb{L}(\mathcal{H})$, in the VC dimension sense, that are not disjoint with $h^\star$. The target model has the lowest complexity among the unbiased models in $\mathbb{L}(\mathcal{H})$, that are models which contain a target hypothesis.

The intuition of the paradigm proposed by this thesis is presented in Figure 1.16, in which the ellipses represent some models in $\mathbb{L}(\mathcal{H})$, and their area is proportional to their VC dimension. Assume that $\mathcal{H}$ is all we have to learn on, i.e., we are not willing to consider any hypothesis outside $\mathcal{H}$. Then, if we could choose, we would like to learn on $\mathcal{M}^\star$: *the model in* $\mathbb{L}(\mathcal{H})$ *with the least VC dimension which contains a* $h^\star$.

Of course, this model is dependent on both $\mathbb{L}(\mathcal{H})$ and P, i.e., it is not distribution-free, and thus we cannot establish beforehand, without looking at data, on which model of $\mathbb{L}(\mathcal{H})$ we should learn. Moreover, even if we looked at data, it could not be possible to search $\mathbb{L}(\mathcal{H})$ exhaustively to properly estimate $\mathcal{M}^\star$ by a $\hat{\mathcal{M}}$ learned from data and, in the general case, there is nothing guaranteeing that it is possible to estimate $\mathcal{M}^\star$ anyhow.

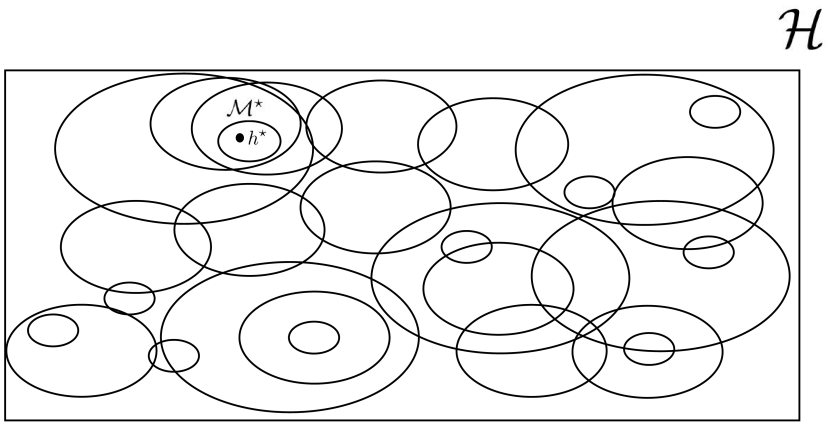

Figure 1.16: *Decomposition of* $\mathcal{H}$ *by a* $\mathbb{L}(\mathcal{H})$. *We omitted some models for a better visualization, since* $\mathbb{L}(\mathcal{H})$ *should cover* $\mathcal{H}$.

In fact, when learning on a model $\hat{\mathcal{M}} \in \mathbb{L}(\mathcal{H})$, that is random, since is learned (estimated) from data, we commit three types of errors, which are

$$\textbf{(II)}\ L(\hat{h}^{\mathbb{A}}_{\hat{\mathcal{M}}}) - L(h^\star_{\hat{\mathcal{M}}}) \qquad \textbf{(III)}\ L(h^\star_{\hat{\mathcal{M}}}) - L(h^\star) \qquad \textbf{(IV)}\ L(\hat{h}^{\mathbb{A}}_{\hat{\mathcal{M}}}) - L(h^\star), \tag{1.16}$$

that we call types II, III, and IV estimation errors, in which $\hat{h}^{\mathbb{A}}_{\hat{\mathcal{M}}} \in \hat{\mathcal{M}}$ is a hypothesis estimated by an algorithm $\mathbb{A}$ applied to $\hat{\mathcal{M}}$, which will be formally defined in Section 1.5.2.[7]

[7] We define type I estimation error in Section 1.5.2 (cf. (1.23)).

In a broad sense, type III error would represent the bias of learning on $\hat{\mathcal{M}}$, while type II would represent the variance within $\hat{\mathcal{M}}$, and type IV would be the error, with respect to $\mathcal{H}$, committed when learning on $\hat{\mathcal{M}}$ with algorithm $\mathbb{A}$. Indeed, type III error compares a target hypothesis of $\hat{\mathcal{M}}$ with a target hypothesis of $\mathcal{H}$, hence any difference between them would be a systematic bias of learning on $\hat{\mathcal{M}}$ when compared to learning on $\mathcal{H}$. Type II error compares the loss of the estimated hypothesis of $\hat{\mathcal{M}}$ and the loss of its target, assessing how much the estimated hypothesis varies from a target of $\hat{\mathcal{M}}$, while type IV estimation error is the effective error committed, since compares the estimated hypothesis of $\hat{\mathcal{M}}$ with a target $h^\star$ of $\mathcal{H}$. These estimation errors are illustrated in Figure 1.17.

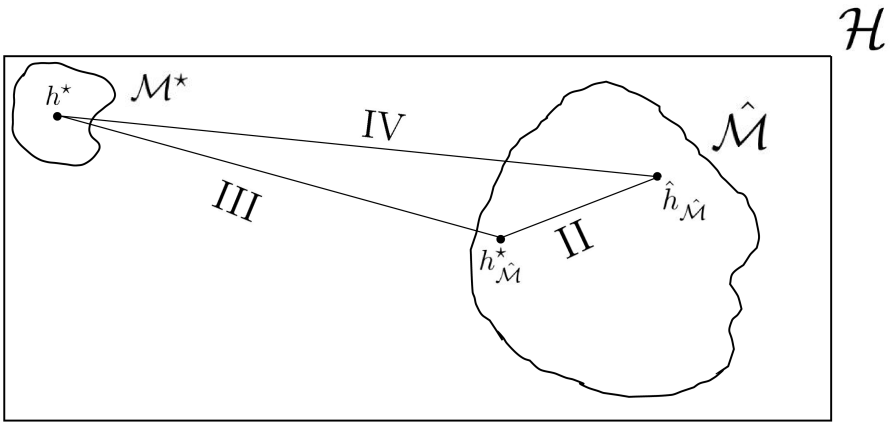

Figure 1.17: *Types II, III, and IV estimation errors when learning on $\hat{\mathcal{M}}$, in which $\hat{h}_{\hat{\mathcal{M}}} \equiv \hat{h}^{\mathbb{A}}_{\hat{\mathcal{M}}}$.*

As is often the case, there will be a bias-variance trade-off that should be minded when learning (on) $\hat{\mathcal{M}}$, so it is important to guarantee that, when N increases, all the estimation errors tend to zero. Furthermore, as can be seen in the examples of Section 1.3.2, the number of models in a Learning Space is usually at least exponential on $d_{VC}(\mathcal{H})$, or on the number of parameters which represent the hypotheses in $\mathcal{H}$, so an exhaustive search of $\mathbb{L}(\mathcal{H})$ may not be possible. Hence, the possibility of a Model Selection approach depends on algorithms which (a) are more efficient than an exhaustive search of $\mathbb{L}(\mathcal{H})$ and (b) guarantee that, when the sample size increases, types II and III, and consequently type IV, estimation errors converge to zero, so there is no systematic bias when learning on $\hat{\mathcal{M}}$, and it is statistically consistent to do so.

This thesis aims to present an approach to Model Selection satisfying (a) and (b). First, we define a learning framework consisting of an estimator $\hat{\mathcal{M}}$, for which we can rigorously show, by extending the VC theory, that under mild conditions it is possible to estimate $\mathcal{M}^\star$ by this $\hat{\mathcal{M}}$, a random model learned non-exhaustively from data, which converges to $\mathcal{M}^\star$ with probability one, when the sample size tends to infinity. Furthermore, in Chapter 2, we establish bounds for the tail probabilities of the estimation errors of learning on $\hat{\mathcal{M}}$ which may be tighter than those we have when learning on $\mathcal{H}$, i.e., by introducing a bias III, which converges to zero, we may decrease the variance II of the learning process, so it is more efficient to learn on a model learned from data.

Indeed, when one learns direct on $\mathcal{H}$ via ERM under the classical VC theory framework

(cf. Appendix A), he commits the error

$$\textbf{(II in } \mathcal{H}\textbf{)} \; L(\hat{h}^{\mathcal{D}_N}) - L(h^\star),$$

that is type II estimation error in $\mathcal{H}$, when algorithm $\mathbb{A}$ minimizes the empirical error under $\mathcal{D}_N$, while when one learns via a Learning Space he commits the type IV estimation error. These errors are depicted in Figure 1.18.

From results in Chapter 2 and Appendix A, we will establish bounds for the tail probabilities of the errors in Figure 1.18, such that the bounds for type IV estimation error are tighter than those for type II in $\mathcal{H}$, raising the possibility that, when learning via Learning Spaces, one may be committing a lesser error compared to learning on $\mathcal{H}$ directly. In Section 4.1, we present some simulations of learning via the Partition Lattice Learning Space that illustrate instances in which learning via Learning Space is indeed better, that is, type IV estimation error is lesser than type II in $\mathcal{H}$.

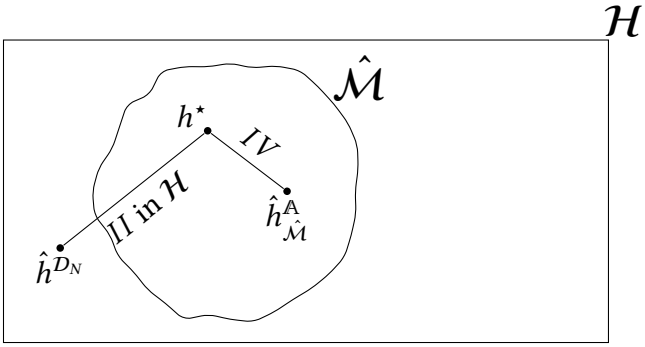

Figure 1.18: *Type IV estimation error and type II estimation error of learning on $\mathcal{H}$ via ERM with sample $\mathcal{D}_N$.*

This fact is a byproduct of the proposed approach, which was not developed specifically to beat the ERM framework, but was rather developed seeking a data-driven systematic, consistent and non-exhaustive approach for Model Selection. More than consistency, we established actually that the rate of convergence of type IV estimation error may be faster than that of type II in $\mathcal{H}$, enhancing the quality of learning when compared with the classical VC theory learning framework.

Since the computation of $\hat{\mathcal{M}}$ may be very expensive, this faster rate of convergence leads to the paradigm that *the lack of data may be mitigated by high computational power*, in this instance represented by the fact that, with a same sample $\mathcal{D}_N$, is it possible to better learn (type IV < type II in $\mathcal{H}$) by employing high computational power to compute $\hat{\mathcal{M}}$. This paradigm will be further discussed throughout the thesis.

Following the structure of the thesis, in Chapter 3, we discuss the U-curve phenomenon [133], and show how it can be explored, via the solution of a U-curve problem [131] through a U-curve algorithm [8, 55], to estimate a target hypothesis by first learning a model $\hat{\mathcal{M}}$ and then a hypothesis on it, without exhaustively searching $\mathbb{L}(\mathcal{H})$.

In Chapter 4, we instantiate the proposed method to solve specific learning problems, illustrating its qualities through simulations and applications to real data sets. In special, we

illustrate that suboptimal algorithms, which are much more efficient, but for which there are no theoretical guarantees of optimality, from both the statistical and computational perspective, work well in practice, and may allow the employment of the method to solve real problems when high computational resources are not available. We also illustrate instances in which it is better to learn via Learning Spaces than directly on $\mathcal{H}$, in the framework of Figure 1.18.

In summary, in this thesis we show that the proposed data-driven method for Model Selection is not only systematic, consistent and non-exhaustive, but may be better than learning under the classical VC theory framework directly on $\mathcal{H}$, and may be applicable to solve real problems, even when high computational resources are not available, via suboptimal efficient algorithms.

1.5 The learning of hypotheses via Learning Spaces

The Learning Space framework for learning hypotheses is composed of two steps: first learn a model $\hat{\mathcal{M}}$ from $\mathbb{L}(\mathcal{H})$ and then learn a hypothesis on $\hat{\mathcal{M}}$. In this section, we define $\hat{\mathcal{M}}$ and present two ways of learning hypotheses on it.

1.5.1 Learning model $\hat{\mathcal{M}}$

Model Selection via Learning Spaces is performed by applying a $(\Omega, \mathcal{S})$-measurable function $\mathbb{M}_{\mathbb{L}(\mathcal{H})}$, dependent on the Learning Space $\mathbb{L}(\mathcal{H})$, satisfying

$$\omega \in \Omega \xrightarrow{(\mathcal{D}_N, \hat{L})} (\mathcal{D}_N(\omega), \hat{L}(\omega)) \xrightarrow{\mathbb{M}_{\mathbb{L}(\mathcal{H})}} \hat{\mathcal{M}}(\omega) \in \mathbb{L}(\mathcal{H}), \tag{1.17}$$

which is such that, given $\mathcal{D}_N$ and an estimator $\hat{L}$ of the error of each candidate model, learns a $\hat{\mathcal{M}} \in \mathbb{L}(\mathcal{H})$. Note from (1.17) that $\hat{\mathcal{M}}$ is a $(\Omega, \mathcal{S})$-measurable $\mathbb{L}(\mathcal{H})$-valued function, as it is the composition of measurable functions, i.e., $\hat{\mathcal{M}} := \hat{\mathcal{M}}_{\mathcal{D}_N, \hat{L}, \mathbb{L}(\mathcal{H})} = \mathbb{M}_{\mathbb{L}(\mathcal{H})}\big(\mathcal{D}_N, \hat{L}\big)$. Even though $\hat{\mathcal{M}}$ depends on $\mathcal{D}_N$, $\hat{L}$ and $\mathbb{L}(\mathcal{H})$, we drop the subscripts to ease notation.

The main feature of $\mathbb{M}_{\mathbb{L}(\mathcal{H})}$ which allows the learning of models is that type III estimation error converges in probability to zero:

$$\mathbb{P}\left(L(h^\star_{\hat{\mathcal{M}}}) - L(h^\star) > \epsilon\right) \xrightarrow{N \to \infty} 0, \tag{1.18}$$

for all $\epsilon > 0$, which is equivalent to

$$\mathbb{P}\left(\hat{\mathcal{M}} \cap h^\star = \emptyset\right) \xrightarrow{N \to \infty} 0,$$

since $|\mathbb{L}(\mathcal{H})| < \infty$. In fact, it is desired the model learned by $\mathbb{M}_{\mathbb{L}(\mathcal{H})}$ to be as simple as it can be under the restriction that it converges to the target model. Therefore, we will develop a $\mathbb{M}_{\mathbb{L}(\mathcal{H})}$ such that

$$\hat{\mathcal{M}} = \mathbb{M}_{\mathbb{L}(\mathcal{H})}\big(\mathcal{D}_N, \hat{L}\big) \xrightarrow{N \to \infty} \mathcal{M}^\star \text{ with probability one,} \tag{1.19}$$

which implies (1.18).

A $\mathbb{M}_{\mathbb{L}(\mathcal{H})}$ which satisfies (1.19) may be defined by mimicking the definition of $\mathcal{M}^\star$, but employing the estimated error $\hat{L}$ instead of the out-of-sample error L. Define in $\mathbb{L}(\mathcal{H})$ the equivalence relation given by

$$\mathcal{M}_i \hat{\sim} \mathcal{M}_j \text{ if, and only if, } d_{VC}(\mathcal{M}_i) = d_{VC}(\mathcal{M}_j) \text{ and } \hat{L}(\mathcal{M}_i) = \hat{L}(\mathcal{M}_j),$$

for $\mathcal{M}_i, \mathcal{M}_j \in \mathbb{L}(\mathcal{H})$, which is a random $(\Omega, \mathcal{S})$-measurable equivalence relation. Let

$$\hat{\mathcal{L}} = \arg\min_{\mathcal{M} \in \mathbb{L}(\mathcal{H})/\hat{\sim}} \hat{L}(\mathcal{M})$$

be the classes in $\mathbb{L}(\mathcal{H})/\hat{\sim}$ which are global minimums of $\mathbb{L}(\mathcal{H})$ (cf. Definition 1.17). Then, $\mathbb{M}_{\mathbb{L}(\mathcal{H})}$ selects

$$\hat{\mathcal{M}} = \arg\min_{\mathcal{M} \in \hat{\mathcal{L}}} d_{VC}(\mathcal{M}), \tag{1.20}$$

the simplest global minimum of $\mathbb{L}(\mathcal{H})/\hat{\sim}$.

By selecting $\hat{\mathcal{M}}$ this way, we get to learn on relatively simple models, what is in accordance with the paradigm of selecting the simplest model that properly express reality [9], which in this case is represented by the fact that $\hat{\mathcal{M}}$ is the simplest global minimum. In Section 2.2, we show that (1.19) holds when we define $\hat{\mathcal{M}}$ as (1.20).

1.5.2 Learning hypotheses on $\hat{\mathcal{M}}$

Once $\hat{\mathcal{M}}$ is selected, we need to learn hypotheses on it, that will be employed to solve the practical problem at hand. Although other frameworks could be used to learn on $\hat{\mathcal{M}}$, we propose two manners of performing such learning, that are characterized, respectively, by resubstitution on the sample $\mathcal{D}_N$, and considering an independent sample, as follows.

A straightforward manner of learning on $\hat{\mathcal{M}}$ is to simply consider

$$\hat{h}_{\hat{\mathcal{M}}}^{\mathcal{D}_N} := \arg\min_{h \in \hat{\mathcal{M}}} L_{\mathcal{D}_N}(h), \tag{1.21}$$

that are the hypotheses which minimize the empirical error under $\mathcal{D}_N$ on $\hat{\mathcal{M}}$. Since $\mathcal{D}_N$ was employed on the selection of $\hat{\mathcal{M}}$, through $\hat{L}$, estimator $\hat{h}_{\hat{\mathcal{M}}}^{\mathcal{D}_N}$ may be biased [50], so type IV estimation error $L(\hat{h}_{\hat{\mathcal{M}}}^{\mathcal{D}_N}) - L(h^\star)$ may be great with high probability if N is not large enough. We call this framework *learning by reusing*.

Another manner of learning on $\hat{\mathcal{M}}$ is to consider a sample $\tilde{\mathcal{D}}_M = \{(\tilde{X}_l, \tilde{Y}_l) : 1 \leq l \leq M\}$ of M independent and identically distributed random vectors with distribution P, independent of $\mathcal{D}_N$, and consider

$$\hat{h}_{\hat{\mathcal{M}}}^{\tilde{\mathcal{D}}_M} := \arg\min_{h \in \hat{\mathcal{M}}} L_{\tilde{\mathcal{D}}_M}(h), \tag{1.22}$$

that are the hypotheses which minimize the empirical error under $\tilde{\mathcal{D}}_M$ on $\hat{\mathcal{M}}$. Since the

sample $\tilde{\mathcal{D}}_M$ is independent of $\mathcal{D}_N$, it may provide a less biased estimator of $h^\star_{\hat{\mathcal{M}}}$. We call this framework *learning with independent sample*. The two frameworks for learning on $\hat{\mathcal{M}}$ are depicted in Figure 1.19.

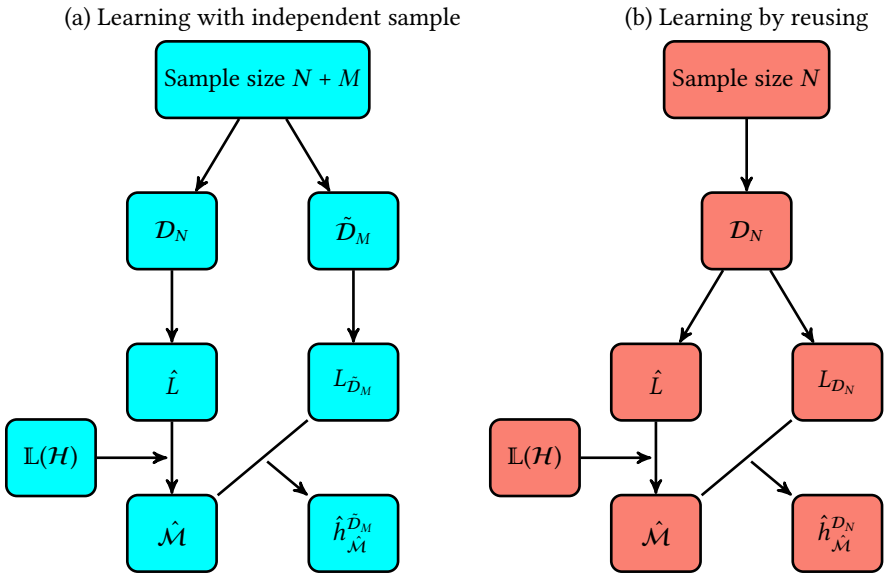

Figure 1.19: *Two frameworks for learning hypotheses via Learning Spaces. (a) A sample of size $N+M$ is split into two, one of size N that is used to estimate $\hat{\mathcal{M}}$ by minimization of $\hat{L}$ on $\mathbb{L}(\mathcal{H})$, and another of size M used to learn a hypothesis on $\hat{\mathcal{M}}$ by ERM. (b) The whole sample of size N is used for estimating $\hat{\mathcal{M}}$ by the minimization of $\hat{L}$ on $\mathbb{L}(\mathcal{H})$, and to estimate hypotheses on $\hat{\mathcal{M}}$ via ERM.*

On the one hand, if the available sample is *great enough*, then we may split it into $\mathcal{D}_N$ and $\tilde{\mathcal{D}}_M$, with *great size* themselves, and learn with independent sample. On the other hand, if few samples are available, it could be better to learn by reusing, even if such framework is biased, since dividing the sample into two would generate even smaller samples. In Chapter 2, we discuss what a sample *great enough* means in these cases, and better quantify the qualities and pitfalls of each framework.

Besides types II and IV (cf. (1.16)), there is another estimation error that depends on the algorithm one chooses to learn on $\hat{\mathcal{M}}$. The type I estimation error is defined as

$$\textbf{(I)} \begin{cases} \sup\limits_{h\in\hat{\mathcal{M}}} \left|L_{\tilde{\mathcal{D}}_M}(h) - L(h)\right| & \text{if learning with independent sample} \\ \sup\limits_{h\in\hat{\mathcal{M}}} \left|L_{\mathcal{D}_N}(h) - L(h)\right| & \text{if learning by reusing} \end{cases}, \tag{1.23}$$

which represents how well one can estimate the loss uniformly on $\hat{\mathcal{M}}$ by the empirical error under $\tilde{\mathcal{D}}_M$ or $\mathcal{D}_N$. In a posterior step, after $\hat{\mathcal{M}}$ is selected, one may wish to estimate the loss of the hypotheses in it, and how well this task is accomplished is measured by type I estimation error depending on how it is performed, by either reusing the same sample employed to obtain $\hat{\mathcal{M}}$, or by using an independent sample.

Remark 1.18. *We assume that both supremum in (1.23) are $(\Omega, \mathcal{S})$-measurable, so it is meaningful to calculate probabilities of events which involve them. We also assume throughout this thesis that these supremum, over any fixed $\mathcal{M} \in \mathbb{L}(\mathcal{H})$, are also $(\Omega, \mathcal{S})$-measurable.*

1.6 Next steps

In this chapter, we defined the main concepts of learning theory and introduced the Learning Spaces as structured collections of candidate models for Model Selection. We established an abstract method to build Learning Spaces, and presented examples of them for many problems of interest. Then, we outlined the main objective of this thesis, which revolves around the concept of target model $\mathcal{M}^\star$, more specifically the task of properly estimating it. We presented an estimator for $\mathcal{M}^\star$, and proposed two frameworks for learning on this estimated model.

We have established that the proposed framework for Model Selection is *data-driven* and systematic. This is true since, once the Learning Space, loss function and model error estimator are chosen, both the selection of $\hat{\mathcal{M}}$ and the learning on it are based solely on data, and performed following objective steps within a system, first computing $\hat{\mathcal{M}}$ as (1.20), and then estimating hypotheses on $\hat{\mathcal{M}}$ by (1.21) or (1.22).

An important facet of properly estimating $\mathcal{M}^\star$ is the consistency of doing so, which means the *high quality* of the proposed estimator when the sample size increases. In the next chapter, we will establish sufficient conditions for the convergence in probability of types I, II, III, and IV estimation errors to zero, and of $\hat{\mathcal{M}}$ to $\mathcal{M}^\star$ with probability one, what characterizes the consistency of the framework for Model Selection via Learning Spaces.

Chapter 2

Consistency of Model Selection via Learning Spaces

In Chapter 1, we proposed a data-driven systematic framework for Model Selection based on Learning Spaces. In this chapter, we study sufficient conditions for the consistency of this framework, which, in general lines, means the convergence in probability to zero of the estimation errors, and the convergence of the estimated model to the target, with probability one. The results presented here rely on classical VC theory, of which a review is presented in Appendix A.

We start by formally defining the consistency of Model Selection frameworks.

Definition 2.1. ***(Consistency)*** *A Model Selection framework is consistent if it returns a random model $\hat{\mathcal{M}}$, and an estimated hypothesis $\hat{h}^{\mathbb{A}} \in \hat{\mathcal{M}}$, such that types I, II, III, and IV estimation errors of learning on it converge in probability to zero, and $\hat{\mathcal{M}}$ converges to $\mathcal{M}^\star$ with probability one, when the sample size tends to infinity.*

Remark 2.2. *The convergence of $\hat{\mathcal{M}}$ to $\mathcal{M}^\star$ implies the convergence to zero of type III estimation error. Hence, as type IV estimation error reduces to type II when $\hat{\mathcal{M}} = \mathcal{M}^\star$, the non-trivial conditions for consistency are the convergence in probability to zero of types I and II estimation errors, and convergence of $\hat{\mathcal{M}}$ to $\mathcal{M}^\star$ with probability one. In some cases, depending on how the algorithm $\mathbb{A}$ is chosen to learn on $\hat{\mathcal{M}}$, convergence of type I estimation error implies convergence of type II, so convergence of type II estimation error may also be trivial (see Lemma A.18).*

In order to show the consistency of Model Selection via Learning Spaces, one should find bounds for the tail probabilities of types I, II, III, and IV estimation errors, implying their convergence to zero, and assert the convergence of $\hat{\mathcal{M}}$ to $\mathcal{M}^\star$. When proceeding this way, one will be not only establishing the consistency of the framework, but also controlling the rate of the respective convergences, what allows to compare the framework with learning directly on $\mathcal{H}$ via ERM, by comparing the errors in Figure 1.18.

In Section 2.1, we present some classical results of VC theory, which are bounds for tail probabilities of estimation errors. In Section 2.2, we show the convergence of $\hat{\mathcal{M}}$ to the target model, and in Section 2.3 we show the convergence to zero of the estimation

errors. We assume from now on that the loss function is bounded, that is, there exists a constant $C > 0$ such that

$$0 \leq \ell(z, h) \leq C \qquad \text{for all } z \in \mathcal{Z}, h \in \mathcal{H}.$$

The case of unbounded loss functions is treated separately in Section 2.4.

2.1 VC theory and PAC-learnability

In classical learning theory, or VC theory, there are two kinds of estimation errors, whose tail probabilities are

$$\mathbb{P}\left(\sup_{h\in\mathcal{H}} \left|L_{\mathcal{D}_N}(h) - L(h)\right| > \epsilon\right) \tag{2.1}$$

and

$$\mathbb{P}\left(L(\hat{h}^{\mathcal{D}_N}) - L(h^\star) > \epsilon\right), \tag{2.2}$$

for $\epsilon > 0$. In the terminology of this thesis, they are called, respectively, type I and II estimation error, when the target hypotheses of $\mathcal{H}$ are estimated via ERM with sample $\mathcal{D}_N$.

If (2.1) is small, for small ϵ, then we are confident we can estimate $L(h)$ by $L_{\mathcal{D}_N}(h)$, for all $h \in \mathcal{H}$, i.e., we can generalize the in-sample error $L_{\mathcal{D}_N}(h)$ to out-of-sample instances. On the other hand, if (2.2) is small, for small ϵ, then we are confident that a hypothesis which minimizes the in-sample error in $\mathcal{H}$ is as good as a target hypothesis.

The VC dimension (cf. Definition 1.5) is an important tool for bounding the tail probabilities of these estimation errors, and also plays an important role on the concept of *Probably Approximately Correct* (PAC) learning, first proposed by [148], which is a framework for the analysis of learning methods.

Definition 2.3. ***(Agnostic PAC-learnability)*** *A hypotheses space $\mathcal{H}$, under loss function ℓ, is Agnostic PAC-learnable*[1] *if there exists an algorithm $\mathbb{A}$, that processes a sample $\mathcal{D}_N$ and returns a hypothesis $\hat{h}^{\mathbb{A}}$, which is such that*

$$\lim_{N\to\infty} \mathbb{P}\left(L(\hat{h}^{\mathbb{A}}) - L(h^\star) > \epsilon\right) = 0, \tag{2.3}$$

for all $\epsilon > 0$.

If $\mathbb{A}$ returns an ERM hypothesis, then Agnostic PAC-learnability means convergence to zero of the tail probability of type II estimation error (2.2). Observe that (2.3) is equivalent to the existence of a function $m_{\mathcal{H}} : (0,1)^2 \longrightarrow \mathbb{Z}_+$ such that, for all $\epsilon, \delta \in (0,1)$ and $N \geq m(\epsilon, \delta)$, it is true that

$$\mathbb{P}\left(L(h^{\mathbb{A}}) - L(h^\star) > \epsilon\right) \leq \delta.$$

[1] The terminology agnostic means that $L(h^\star)$ may be greater than zero. The expression PAC-learnability by itself refers to the case when $L(h^\star) = 0$.

We note that PAC-learnability is a distribution-free concept, as the limit (2.3) should hold for any distribution P of Z. See [139, Definition 3.3] for more details.

In order to learn hypotheses from data, one should consider approaches which are PAC-learnable for, otherwise, not even an infinite number of samples suffice to estimate $\hat{h}$ such that $L(\hat{h})$ is close to $L(h^\star)$. Moreover, one should ensure that $L(h^\star)$ is *small* for, otherwise, it is impossible to estimate a *good enough* hypothesis from $\mathcal{H}$ anyhow. These two features imply that, for N sufficiently large, $L(\hat{h})$ will be close enough to $L(h^\star)$ with high probability and, since $L(h^\star)$ is *small*, one will then have small error when applying $\hat{h}$.

The concept of PAC-learnability is equivalent to finite VC dimension (see [139, Theorem 6.7] and Section A.4), and the rate of convergence of (2.2) to zero is decreasing on the VC dimension of $\mathcal{H}$. This is the main result of VC theory, which may be stated as follows, and is a consequence of Corollaries A.9 and A.19. Observe that the bounds do not depend on P, and are valid for any distribution Z may have, that is, are distribution-free.

Proposition 2.4. *Assume the loss function is bounded. Fixed a hypotheses space $\mathcal{H}$ with $d_{VC}(\mathcal{H}) < \infty$, there exist sequences $\{B^I_{N,\epsilon} : N \geq 1\}$ and $\{B^{II}_{N,\epsilon} : N \geq 1\}$ of positive real-valued increasing functions with domain $\mathbb{Z}_+$ satisfying*

$$\lim_{N\to\infty} B^I_{N,\epsilon}(k) = \lim_{N\to\infty} B^{II}_{N,\epsilon}(k) = 0,$$

for all $\epsilon > 0$ and $k \in \mathbb{Z}_+$ fixed, such that

$$\mathbb{P}\left(\sup_{h\in\mathcal{H}} \left|L_{D_N}(h) - L(h)\right| > \epsilon\right) \leq B^I_{N,\epsilon}(d_{VC}(\mathcal{H}))$$

$$\mathbb{P}\left(L(\hat{h}^{D_N}) - L(h^\star) > \epsilon\right) \leq B^{II}_{N,\epsilon}(d_{VC}(\mathcal{H})).$$

Furthermore, the following holds:

$$\sup_{h\in\mathcal{H}} \left|L_{D_N}(h) - L(h)\right| \xrightarrow[N\to\infty]{a.s.} 0$$

$$L(\hat{h}^{D_N}) - L(h^\star) \xrightarrow[N\to\infty]{a.s.} 0.$$

Proposition 2.4 tells us that, for fixed ϵ and N, as low the VC dimension of $\mathcal{H}$, the tightest are the VC bounds for types I and II estimation errors. Hence, as small the VC dimension, greater may be the probabilities of estimating uniformly the loss of the hypotheses in $\mathcal{H}$ by L_{D_N} with an error of at most ϵ, and of having the loss of the estimated hypothesis ϵ-close to the loss of a target one.

In view of the proposition, the nature of hypotheses learning brings upon a *trade-off* between the complexity of $\mathcal{H}$ and its minimum out-of-sample error $L(h^\star)$. On the one hand, if $\mathcal{H}$ has low VC dimension, then fewer samples are needed to properly approximate target hypotheses, but as the hypotheses in $\mathcal{H}$ may be rather simple, hence not able to

capture all the nuances of features of Z, $L(h^\star)$ may be too great, so that even if we knew a target $h^\star$, it would not be adequate.

On the other hand, if $\mathcal{H}$ is more complex, more samples are needed, but $L(h^\star)$ may be smaller, since $\mathcal{H}$ contains more complex functions which may better generalize, i.e., have smaller out-of-sample error. As $L(h^\star)$ is not known, and due to practical problems such as *overfitting*, managing this trade-off is not a simple task, as the only quantity really controllable and known in the learning framework is the VC dimension[2] of $\mathcal{H}$. Indeed, $h^\star$ and its error are unknown, and the sample size N is usually fixed and cannot be increased to decrease the bounds of Proposition 2.4. In summary, the only controllable way to decrease these bounds is to decrease the VC dimension, although doing so creates more errors to be controlled (types III and IV estimation errors defined in (1.16)).

Inspired by these bounds, the framework for Model Selection via Learning Spaces seeks to learn on a $\hat{\mathcal{M}}$ with low VC dimension and low bias, so the learning on it is more efficient than on $\mathcal{H}$, what may cause the effective error committed (type IV estimation error) to be smaller than the error committed when learning direct on $\mathcal{H}$ (type II estimation error (2.2)), both depicted in Figure 1.18.

The learning on $\hat{\mathcal{M}}$ tries to balance out the trade-off: since we cannot diminish $L(h^\star)$, as $\mathcal{H}$ contains all hypotheses one is willing to consider, could it be possible to at least better estimate a hypothesis as good as a target, even if a small bias is introduced? Furthermore, could it be better overall than trying to estimate $h^\star$ directly? We answer these two questions and discuss when learning via Learning Spaces is better than learning on $\mathcal{H}$ directly.

We note that, if a Model Selection framework is consistent (cf. Definition 2.1), then it is Agnostic PAC-learnable. Indeed, condition (2.3) is equivalent to convergence in probability to zero of type IV estimation error, in which $\hat{h}^{\mathbb{A}}$ is the hypothesis returned by the framework. In the case of Model Selection via Learning Spaces, this hypothesis may be given by learning on $\hat{\mathcal{M}}$ with an independent sample, or by reusing (cf. Figure 1.19). We state this fact as a proposition.

Proposition 2.5. *If a Model Selection framework is consistent, then it is Agnostic PAC-learnable.*

In order to show consistency, we start by establishing the convergence of $\hat{\mathcal{M}}$ to $\mathcal{M}^\star$ with probability one.

2.2 Convergence to the target model

We start by studying a result weaker than the convergence of $\hat{\mathcal{M}}$ to $\mathcal{M}^\star$, that is the convergence of $L(\hat{\mathcal{M}})$ to $L(\mathcal{M}^\star)$.

In order to have $L(\hat{\mathcal{M}}) = L(\mathcal{M}^\star)$, one does not need to know exactly $L(\mathcal{M})$ for all $\mathcal{M} \in \mathbb{L}(\mathcal{H})$, i.e., one does not need $\hat{L}(\mathcal{M}) = L(\mathcal{M})$, for all $\mathcal{M} \in \mathbb{L}(\mathcal{H})$. We argue that it suffices to have $\hat{L}(\mathcal{M})$ close enough to $L(\mathcal{M})$, for all $\mathcal{M} \in \mathbb{L}(\mathcal{H})$, so the global minimums of

[2] Although in some examples the VC dimension is not exactly known, there are usually upper bounds for it.

$\mathbb{L}(\mathcal{H})$ will have the same error as $\mathcal{M}^\star$, even if it is not possible to properly estimate their error. This "close enough" depends on P, hence is not distribution-free, and is given by the *maximum discrimination error* (MDE) of $\mathbb{L}(\mathcal{H})$ under P, which we define as

$$\epsilon^\star = \epsilon^\star(\mathbb{L}(\mathcal{H}), P) := \min_{\substack{\mathcal{M}\in\mathbb{L}(\mathcal{H}) \\ L(\mathcal{M})>L(\mathcal{M}^\star)}} L(\mathcal{M}) - L(\mathcal{M}^\star).$$

The MDE is the minimum difference between the out-of-sample error of a target hypothesis and the best hypothesis in a model which does not contain a target. In other words, it is the difference between the error of the best model $\mathcal{M}^\star$ and the second to best. The meaning of $\epsilon^\star$ is depicted in Figure 2.1.

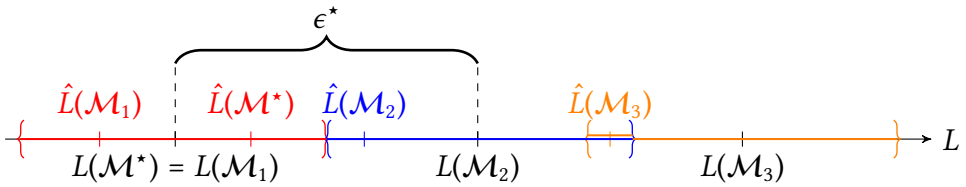

Figure 2.1: *The errors of the equivalence classes (cf. (1.15)) of $\mathbb{L}(\mathcal{H})$ in ascending order. The MDE $\epsilon^\star$ is the difference between the error of the target class $\mathcal{M}^\star$, and the second to best $\mathcal{M}_2$. The colored intervals represent a distance of $\epsilon^\star/2$ from the real error of each model, and the colored estimated errors $\hat{L}$ illustrate a case such that the estimated error is within $\epsilon^\star/2$ of the real error for all models. The class $\mathcal{M}_1$ has the same error as $\mathcal{M}^\star$, but has a smaller estimated error, and, by the definition of $\mathcal{M}^\star$, greater VC dimension. Note from the representation that, if one can estimate $\hat{L}$ within a margin of error of $\epsilon^\star/2$, then $\hat{\mathcal{M}}$ will be a model with the same error as $\mathcal{M}^\star$, in this case $\mathcal{M}_1$ (cf. Proposition 2.6).*

The MDE is defined only if there exists at least one $\mathcal{M} \in \mathbb{L}(\mathcal{H})$ such that $h^\star \cap \mathcal{M} = \emptyset$, i.e., there is a subset in $\mathbb{L}(\mathcal{H})$ which does not contain a target hypothesis. If $h^\star \cap \mathcal{M} \neq \emptyset$ for all $\mathcal{M} \in \mathbb{L}(\mathcal{H})$, then type III estimation error is zero, and type IV reduces to type II. From this point, we assume that $\epsilon^\star$ is well defined.

The terminology MDE is used for we can show that a fraction of $\epsilon^\star$ is the greatest error one can commit when estimating $L(\mathcal{M})$ by $\hat{L}(\mathcal{M})$, for all $\mathcal{M} \in \mathbb{L}(\mathcal{H})$, in order for $L(\hat{\mathcal{M}})$ to be equal to $L(\mathcal{M}^\star)$. This is the result of the next proposition.

Proposition 2.6. *Assume there exists $\delta > 0$ such that*

$$\mathbb{P}\left(\max_{i\in\mathcal{J}} \left|L(\mathcal{M}_i) - \hat{L}(\mathcal{M}_i)\right| < \epsilon^\star/2\right) \geq 1 - \delta. \tag{2.4}$$

Then

$$\mathbb{P}\left(L(\hat{\mathcal{M}}) = L(\mathcal{M}^\star)\right) \geq 1 - \delta. \tag{2.5}$$

Proof. If

$$\max_{i\in\mathcal{J}} \left|L(\mathcal{M}_i) - \hat{L}(\mathcal{M}_i)\right| < \epsilon^\star/2$$

then, for any $i \in \mathcal{J}$ such that $L(\mathcal{M}_i) > L(\mathcal{M}^\star)$, we have

$$\hat{L}(\mathcal{M}_i) - \hat{L}(\mathcal{M}^\star) > L(\mathcal{M}_i) - L(\mathcal{M}^\star) - \epsilon^\star \geq 0, \tag{2.6}$$

in which the last inequality follows from the definition of $\epsilon^\star$. From (2.6) follows that the global minimum of $\mathbb{L}(\mathcal{H})/_\sim$ with the least VC dimension, that is $\hat{\mathcal{M}}$, is such that $L(\hat{\mathcal{M}}) = L(\mathcal{M}^\star)$. Indeed, from (2.6) follows that $\hat{L}(\mathcal{M}) > \hat{L}(\mathcal{M}^\star)$ for all $\mathcal{M} \in \mathbb{L}(\mathcal{H})$ such that $L(\mathcal{M}) > L(\mathcal{M}^\star)$.

Hence, since $\hat{L}(\hat{\mathcal{M}}) \leq \hat{L}(\mathcal{M}^\star)$, we must have $L(\hat{\mathcal{M}}) = L(\mathcal{M}^\star)$. Therefore, we have the inclusion of events

$$\left\{ \max_{i \in \mathcal{J}} \left| L(\mathcal{M}_i) - \hat{L}(\mathcal{M}_i) \right| < \epsilon^\star/2 \right\} \subset \left\{ L(\hat{\mathcal{M}}) = L(\mathcal{M}^\star) \right\}, \tag{2.7}$$

which proves the result. □

Remark 2.7. *Since there may exist $\mathcal{M} \in \mathbb{L}(\mathcal{H})/_\sim$ with $L(\mathcal{M}) = L(\mathcal{M}^\star)$ and $d_{VC}(\mathcal{M}) > d_{VC}(\mathcal{M}^\star)$, condition (2.4) guarantees only that the estimated error of both $\mathcal{M}$ and $\mathcal{M}^\star$ is lesser than the estimated error of any model with error greater than theirs, but it may happen that $\hat{L}(\mathcal{M}) < \hat{L}(\mathcal{M}^\star)$ (see Figure 2.1 for an example). In this instance, we have $\hat{\mathcal{M}} = \mathcal{M}$ and $L(\hat{\mathcal{M}}) = L(\mathcal{M}^\star)$.*

From now on, we consider that $\hat{L}$ is of the form

$$\hat{L}(\mathcal{M}) = \frac{1}{m} \sum_{j=1}^{m} \hat{L}^{(j)}(\hat{h}^{(j)}_{\mathcal{M}}), \qquad \mathcal{M} \in \mathbb{L}(\mathcal{H}), \tag{2.8}$$

in which there are m pairs of independent training and validation samples, $\hat{L}^{(j)}$ is the empirical error under the j-th validation sample, and $\hat{h}^{(j)}_{\mathcal{M}}$ is an ERM hypothesis of $\mathcal{M}$ under the j-th training sample, denoted by $\mathcal{D}^{(j)}_N$.

We assume independence between samples within a pair j, but there may exist dependence between samples of distinct pairs j, j'. The validation sample and k-fold cross validation estimators, presented in Section 1.2.5, are of the form (2.8) with $m = 1$ and $m = k$, respectively. In this case, we may obtain a bound for (2.5) depending on $\epsilon^\star$, on $d_{VC}(\mathbb{L}(\mathcal{H}))$, and on bounds for tail probabilities of type I estimation error under each validation and training sample (cf. Proposition 2.4).

These bounds also depend on the number of maximal models of $\mathbb{L}(\mathcal{H})$, which are models in

$$\operatorname{Max} \mathbb{L}(\mathcal{H}) = \left\{ \mathcal{M} \in \mathbb{L}(\mathcal{H}) : \text{ if } \mathcal{M} \subset \mathcal{M}' \in \mathbb{L}(\mathcal{H}) \text{ then } \mathcal{M} = \mathcal{M}' \right\},$$

that are models not contained in any element in $\mathbb{L}(\mathcal{H})$ besides themselves. We denote

$$\mathfrak{m}(\mathbb{L}(\mathcal{H})) = |\operatorname{Max} \mathbb{L}(\mathcal{H})|.$$

If $\mathbb{L}(\mathcal{H})$ is a complete lattice, then the only maximal element of it is its greatest element, so $\mathfrak{m}(\mathbb{L}(\mathcal{H})) = 1$. We have the following rate of convergence of $L(\hat{\mathcal{M}})$ to $L(\mathcal{M}^\star)$, and condition for $\hat{\mathcal{M}}$ to converge to $\mathcal{M}^\star$ with probability one.

Theorem 2.8. *Assume the loss function is bounded. For each $\epsilon > 0$, let $\{B_{N,\epsilon} : N \geq 1\}$ and $\{\hat{B}_{N,\epsilon} : N \geq 1\}$ be sequences of positive real-valued increasing functions with domain $\mathbb{Z}_+$ satisfying*

$$\lim_{N\to\infty} B_{N,\epsilon}(k) = \lim_{N\to\infty} \hat{B}_{N,\epsilon}(k) = 0,$$

for all $\epsilon > 0$ and $k \in \mathbb{Z}_+$ fixed, and such that

$$\max_j \mathbb{P}\left(\sup_{h\in\mathcal{M}} \left|L_{D_N^{(j)}}(h) - L(h)\right| > \epsilon\right) \leq B_{N,\epsilon}(d_{VC}(\mathcal{M})) \text{ and}$$

$$\max_j \mathbb{P}\left(\sup_{h\in\mathcal{M}} \left|\hat{L}^{(j)}(h) - L(h)\right| > \epsilon\right) \leq \hat{B}_{N,\epsilon}(d_{VC}(\mathcal{M})),$$

for all $\mathcal{M} \in \mathbb{L}(\mathcal{M})$, recalling that $L_{D_N^{(j)}}$ and $\hat{L}^{(j)}$ represent the empirical error under the j-th training and validation samples, respectively. Let $\hat{\mathcal{M}} \in \mathbb{L}(\mathcal{H})$ be a random model learned by $\mathbb{M}_{\mathbb{L}(\mathcal{H})}$. Then,

$$\mathbb{P}\left(L(\hat{\mathcal{M}}) \neq L(\mathcal{M}^\star)\right) \leq m\, \mathfrak{m}(\mathbb{L}(\mathcal{H}))\left[B_{N,\epsilon^\star/8}(d_{VC}(\mathbb{L}(\mathcal{H}))) + \hat{B}_{N,\epsilon^\star/4}(d_{VC}(\mathbb{L}(\mathcal{H})))\right], \tag{2.9}$$

in which m is the number of pairs considered to calculate (2.8). Furthermore, if

$$\max_{\mathcal{M}\in\mathbb{L}(\mathcal{H})} \max_j \sup_{h\in\mathcal{M}} \left|L_{D_N^{(j)}}(h) - L(h)\right| \xrightarrow{a.s.} 0 \text{ and} \tag{2.10}$$

$$\max_{\mathcal{M}\in\mathbb{L}(\mathcal{H})} \max_j \sup_{h\in\mathcal{M}} \left|\hat{L}^{(j)}(h) - L(h)\right| \xrightarrow{a.s.} 0,$$

then

$$\lim_{N\to\infty} \mathbb{P}\left(\hat{\mathcal{M}} = \mathcal{M}^\star\right) = 1.$$

Proof. We will apply Proposition 2.6. Denoting $\hat{h}_i^{(j)} := \hat{h}_{\mathcal{M}_i}^{(j)}$,

$$\begin{aligned}
\mathbb{P}\left(\max_{i\in\mathcal{J}} \left|L(\mathcal{M}_i) - \hat{L}(\mathcal{M}_i)\right| \geq \epsilon^\star/2\right) &\leq \mathbb{P}\left(\max_{i\in\mathcal{J}} \sum_{j=1}^m \frac{1}{m}\left|L(\mathcal{M}_i) - \hat{L}^{(j)}(\hat{h}_i^{(j)})\right| > \epsilon^\star/2\right) \\
&\leq \mathbb{P}\left(\max_j \max_{i\in\mathcal{J}} \left|L(\mathcal{M}_i) - \hat{L}^{(j)}(\hat{h}_i^{(j)})\right| > \epsilon^\star/2\right) \\
&\leq \mathbb{P}\left(\bigcup_{j=1}^m \left\{\max_{i\in\mathcal{J}} \left|L(\mathcal{M}_i) - \hat{L}^{(j)}(\hat{h}_i^{(j)})\right| > \epsilon^\star/2\right\}\right) \\
&\leq \sum_{j=1}^m \mathbb{P}\left(\max_{i\in\mathcal{J}} \left|L(\mathcal{M}_i) - \hat{L}^{(j)}(\hat{h}_i^{(j)})\right| > \epsilon^\star/2\right) \\
&= \sum_{j=1}^m \mathbb{P}\left(\max_{i\in\mathcal{J}} \left|L(\mathcal{M}_i) - L(\hat{h}_i^{(j)}) + L(\hat{h}_i^{(j)}) - \hat{L}^{(j)}(\hat{h}_i^{(j)})\right| > \epsilon^\star/2\right)
\end{aligned}$$

$$
\begin{aligned}
&\leq \sum_{j=1}^{m} \mathbb{P}\left(\max_{i\in\mathcal{J}} L(\hat{h}_i^{(j)}) - L(\mathcal{M}_i) + \max_{i\in\mathcal{J}} \left| L(\hat{h}_i^{(j)}) - \hat{L}^{(j)}(\hat{h}_i^{(j)}) \right| > \epsilon^\star/2 \right) \\
&\leq \sum_{j=1}^{m} \mathbb{P}\left(\max_{i\in\mathcal{J}} L(\hat{h}_i^{(j)}) - L(\mathcal{M}_i) > \epsilon^\star/4 \right) + \mathbb{P}\left(\max_{i\in\mathcal{J}} \left| L(\hat{h}_i^{(j)}) - \hat{L}^{(j)}(\hat{h}_i^{(j)}) \right| > \epsilon^\star/4 \right) \\
&\leq \sum_{j=1}^{m} \mathbb{P}\left(\max_{i\in\mathcal{J}} L(\hat{h}_i^{(j)}) - L(\mathcal{M}_i) > \epsilon^\star/4 \right) + \mathbb{P}\left(\max_{i\in\mathcal{J}} \sup_{h\in\mathcal{M}_i} \left| \hat{L}^{(j)}(h) - L(h) \right| > \epsilon^\star/4 \right) \qquad (2.11)
\end{aligned}
$$

in which in the first inequality we applied the definition of $\hat{L}(\mathcal{M})$. For each j, the first probability in (2.11) is equal to

$$
\begin{aligned}
&\mathbb{P}\left(\max_{i\in\mathcal{J}} L(\hat{h}_i^{(j)}) - L_{\mathcal{D}_N^{(j)}}(\hat{h}_i^{(j)}) + L_{\mathcal{D}_N^{(j)}}(\hat{h}_i^{(j)}) - L(\mathcal{M}_i) > \epsilon^\star/4 \right) \\
&\quad \leq \mathbb{P}\left(\max_{i\in\mathcal{J}} L(\hat{h}_i^{(j)}) - L_{\mathcal{D}_N^{(j)}}(\hat{h}_i^{(j)}) + L_{\mathcal{D}_N^{(j)}}(h_i^\star) - L(\mathcal{M}_i) > \epsilon^\star/4 \right) \\
&\quad \leq \mathbb{P}\left(\left\{ \max_{i\in\mathcal{J}} \left| L(\hat{h}_i^{(j)}) - L_{\mathcal{D}_N^{(j)}}(\hat{h}_i^{(j)}) \right| > \epsilon^\star/8 \right\} \bigcup \left\{ \max_{i\in\mathcal{J}} \left| L_{\mathcal{D}_N^{(j)}}(h_i^\star) - L(\mathcal{M}_i) \right| > \epsilon^\star/8 \right\} \right) \\
&\quad \leq \mathbb{P}\left(\max_{i\in\mathcal{J}} \sup_{h\in\mathcal{M}_i} \left| L_{\mathcal{D}_N^{(j)}}(h) - L(h) \right| > \epsilon^\star/8 \right),
\end{aligned}
$$

in which the first inequality follows from the fact that $L_{\mathcal{D}_N^{(j)}}(\hat{h}_i^{(j)}) \leq L_{\mathcal{D}_N^{(j)}}(h_i^\star)$, and the last follows since $L(\mathcal{M}_i) = L(h_i^\star)$. We conclude that

$$
\begin{aligned}
&\mathbb{P}\left(\max_{i\in\mathcal{J}} \left| L(\mathcal{M}_i) - \hat{L}(\mathcal{M}_i) \right| \geq \epsilon^\star/2 \right) \\
&\leq \sum_{j=1}^{m} \mathbb{P}\left(\max_{i\in\mathcal{J}} \sup_{h\in\mathcal{M}_i} \left| L_{\mathcal{D}_N^{(j)}}(h) - L(h) \right| > \epsilon^\star/8 \right) + \mathbb{P}\left(\max_{i\in\mathcal{J}} \sup_{h\in\mathcal{M}_i} \left| \hat{L}^{(j)}(h) - L(h) \right| > \epsilon^\star/4 \right).
\end{aligned}
$$

If $\mathcal{M}_1 \subset \mathcal{M}_2$ then, for any $\epsilon > 0$ and $j = 1, \ldots, m$, we have the following inclusion of events

$$
\begin{aligned}
\left\{ \sup_{h\in\mathcal{M}_1} \left| \hat{L}^{(j)}(h) - L(h) \right| > \epsilon \right\} &\subset \left\{ \sup_{h\in\mathcal{M}_2} \left| \hat{L}^{(j)}(h) - L(h) \right| > \epsilon \right\} \\
\left\{ \sup_{h\in\mathcal{M}_1} \left| L_{\mathcal{D}_N^{(j)}}(h) - L(h) \right| > \epsilon \right\} &\subset \left\{ \sup_{h\in\mathcal{M}_2} \left| L_{\mathcal{D}_N^{(j)}}(h) - L(h) \right| > \epsilon \right\},
\end{aligned}
$$

hence it is true that

$$
\begin{aligned}
\left\{ \max_{i\in\mathcal{J}} \sup_{h\in\mathcal{M}_i} \left| \hat{L}^{(j)}(h) - L(h) \right| > \epsilon^\star/4 \right\} &\subset \left\{ \max_{\mathcal{M}\in \operatorname{Max}\mathbb{L}(\mathcal{H})} \sup_{h\in\mathcal{M}} \left| \hat{L}^{(j)}(h) - L(h) \right| > \epsilon^\star/4 \right\} \\
\left\{ \max_{i\in\mathcal{J}} \sup_{h\in\mathcal{M}_i} \left| L_{\mathcal{D}_N^{(j)}}(h) - L(h) \right| > \epsilon^\star/8 \right\} &\subset \left\{ \max_{\mathcal{M}\in \operatorname{Max}\mathbb{L}(\mathcal{H})} \sup_{h\in\mathcal{M}} \left| L_{\mathcal{D}_N^{(j)}}(h) - L(h) \right| > \epsilon^\star/8 \right\},
\end{aligned}
$$

which yields

$$\begin{aligned}
&\mathbb{P}\left(\max_{i\in\mathcal{J}}\left|L(\mathcal{M}_i)-\hat{L}(\mathcal{M}_i)\right|\geq\epsilon^\star/2\right)\\
&\leq\sum_{j=1}^{m}\sum_{\mathcal{M}\in\operatorname{Max}\mathbb{L}(\mathcal{H})}\mathbb{P}\left(\sup_{h\in\mathcal{M}}\left|L_{\mathcal{D}_N^{(j)}}(h)-L(h)\right|>\epsilon^\star/8\right)+\mathbb{P}\left(\sup_{h\in\mathcal{M}}\left|\hat{L}^{(j)}(h)-L(h)\right|>\epsilon^\star/4\right)\\
&\leq m\sum_{\mathcal{M}\in\operatorname{Max}\mathbb{L}(\mathcal{H})}\left[B_{N,\epsilon^\star/8}(d_{VC}(\mathcal{M}))+\hat{B}_{N,\epsilon^\star/4}(d_{VC}(\mathcal{M}))\right]\\
&\leq m\left|\operatorname{Max}\mathbb{L}(\mathcal{H})\right|\left[B_{N,\epsilon^\star/8}(d_{VC}(\mathbb{L}(\mathcal{H})))+\hat{B}_{N,\epsilon^\star/4}(d_{VC}(\mathbb{L}(\mathcal{H})))\right],
\end{aligned}\tag{2.12}$$

in which the last inequality follows from the fact that both $\hat{B}_{N,\epsilon^\star/4}$ and $B_{N,\epsilon^\star/8}$ are increasing functions, and $d_{VC}(\mathbb{L}(\mathcal{H}))=\max_{\mathcal{M}\in\mathbb{L}(\mathcal{H})}d_{VC}(\mathcal{M})$. The result follows from Proposition 2.6 since

$$\{L(\hat{\mathcal{M}})\neq L(\mathcal{M}^\star)\}\subset\left\{\max_{i\in\mathcal{J}}\left|L(\mathcal{M}_i)-\hat{L}(\mathcal{M}_i)\right|\geq\epsilon^\star/2\right\}.$$

If the almost sure convergences (2.10) hold, then

$$\hat{L}(\mathcal{M})\xrightarrow[N\to\infty]{\text{a.s.}}L(\mathcal{M})\tag{2.13}$$

for all $\mathcal{M}\in\mathbb{L}(\mathcal{H})$, since, if $L(h)=\hat{L}^{(j)}(h)=L^{(j)}_{\mathcal{D}_N}(h)$ for all $j=1,\ldots,m$ and $h\in\mathcal{H}$, then $\hat{L}(\mathcal{M})=L(\mathcal{M})$ for all $\mathcal{M}\in\mathbb{L}(\mathcal{H})$. Observe that

$$\left\{\max_{\mathcal{M}\in\mathbb{L}(\mathcal{H})}\left|L(\mathcal{M})-\hat{L}(\mathcal{M})\right|=0\right\}\subset\left\{\hat{\mathcal{M}}=\mathcal{M}^\star\right\},\tag{2.14}$$

since, if the estimated error $\hat{L}$ is equal to the real error L, then the definitions of $\hat{\mathcal{M}}$ and $\mathcal{M}^\star$ coincide. As the probability of the event on the left hand-side of (2.14) converges to one if (2.13) is true, we conclude that, if (2.10) hold, then $\hat{\mathcal{M}}$ converges to $\mathcal{M}^\star$ with probability one. □

A bound for $\mathbb{P}(L(\hat{\mathcal{M}})\neq L(\mathcal{M}^\star))$, and the almost sure convergence of $\hat{\mathcal{M}}$ to $\mathcal{M}^\star$ in the case of k-fold cross validation, follow from Proposition 2.4, recalling that the sample size in each training and validation sample is $(k-1)n$ and n, respectively, with $N=kn$. Analogously, we may obtain a bound when an independent validation sample is considered. This result is stated in the next theorem.

Theorem 2.9. *Assume the loss function is bounded. If $\hat{L}$ is given by k-fold cross-validation or by an independent validation sample, then $\hat{\mathcal{M}}$ converges with probability one to $\mathcal{M}^\star$.*

Proof. We need to show that (2.10) holds in these instances. For any $\epsilon>0$, by Corollary A.9,

$$\mathbb{P}\left(\max_{\mathcal{M}\in\mathbb{L}(\mathcal{H})}\max_{j}\sup_{h\in\mathcal{M}}\left|L_{\mathcal{D}_N^{(j)}}(h)-L(h)\right|>\epsilon\right)\leq\sum_{j=1}^{m}\mathbb{P}\left(\sup_{h\in\mathcal{H}}\left|L_{\mathcal{D}_N^{(j)}}(h)-L(h)\right|>\epsilon\right)$$

$$\leq m\, 8 \exp\left\{ d_{VC}(\mathcal{H}) \left(1 + \ln \frac{N_j}{d_{VC}(\mathcal{H})} - N_j \frac{\epsilon^2}{32C^2} \right) \right\}$$

in which N_j is the size of the j-th training sample. By the inequality above, and Borel-Cantelli Lemma (cf. Lemma B.14), the first convergence in (2.10) holds. The second convergence holds since the inequality above is also true, but with N_j interchanged by $\hat{N}_j$, the size of the j-th validation sample. □

From bound (2.9), follows that we have to better estimate with the training samples, that require a precision of $\epsilon^\star/8$ in contrast to a precision $\epsilon^\star/4$ with the validation samples. Hence, as is done in k-fold cross validation, it is better to consider a greater sample size for training rather than for validation.

Moreover, from this bound follows that, with a fixed sample size, we can have a tighter bound for $\mathbb{P}(L(\hat{\mathcal{M}}) \neq L(\mathcal{M}^\star))$ by choosing a Learning Space with small $d_{VC}(\mathbb{L}(\mathcal{H}))$ and few maximal elements, while attempting to increase $\epsilon^\star$. Of course, there is a trade-off between $d_{VC}(\mathbb{L}(\mathcal{H}))$ and the number of maximal elements of $\mathbb{L}(\mathcal{H})$, the only known free quantities in bound (2.9), since the sample size is fixed and $\epsilon^\star$ is unknown.

As an illustrative example, let $\mathbb{L}(\mathcal{H})$ be the Partition Lattice Learning Space (cf. Example 1.12), and

$$\mathbb{L}_2(\mathcal{H}) := \{\mathcal{M} \in \mathbb{L}(\mathcal{H}) \,:\, d_{VC}(\mathcal{M}) \leq 2\} \tag{2.15}$$

be its models with VC dimension not greater than two. The collection $\mathbb{L}_2(\mathcal{H})$ has $2^{|\mathcal{X}|-1}$ models, that is the number of partitions of $\mathcal{X}$ with at most two blocks, and is a Learning Space, since condition (ii) is inherited from $\mathbb{L}(\mathcal{H})$, and it covers $\mathcal{H}$: any given $h \in \mathcal{H}$ is in the model generated by partition $\{\{x \in \mathcal{X} \,:\, h(x) = 0\}, \{x \in \mathcal{X} \,:\, h(x) = 1\}\}$, which has at most two blocks.

On the one hand, as $\mathbb{L}(\mathcal{H})$ is a complete lattice, it has only one maximal element, and $d_{VC}(\mathbb{L}(\mathcal{H})) = d_{VC}(\mathcal{H})$ since its maximal element is $\mathcal{H}$. On the other hand, $\mathfrak{m}(\mathbb{L}_2(\mathcal{H})) = 2^{|\mathcal{X}|-1} - 1$ since the only element in it that is not maximal is the model of the constant hypotheses, and $d_{VC}(\mathbb{L}_2(\mathcal{H})) = 2$ by definition. Furthermore, $\mathcal{M}^\star$, which has VC dimension at most 2, and $\epsilon^\star$, are the same in both Learning Spaces.

The form of bound (2.9) may dictate on which of these Learning Spaces we can have the tightest bound for $\mathbb{P}(L(\hat{\mathcal{M}}) \neq L(\mathcal{M}^\star))$, so may guide the choice of the Learning Space in this scenario. Nevertheless, in practice, it is also important to consider the computational complexity of $\hat{\mathcal{M}}$ in each instance. For this particular case, we discuss in Section 3.5 a non-exhaustive algorithm to compute $\hat{\mathcal{M}}$ in $\mathbb{L}(\mathcal{H})$, while we need an exhaustive search of $\mathbb{L}_2(\mathcal{H})$ for this task. Therefore, the choice of a Learning Space should mind the consistency of $\hat{\mathcal{M}}$ and all prior information about the problem at hand, but also the computational aspect that enables the application of the method. We discuss the computational aspects in more details in Chapter 3.

The bound of Theorem 2.8 is the first result which supports that by properly modeling the Learning Space one may better learn on $\mathcal{H}$, in this instance by having a greater probability of learning on a model with the same error as $\mathcal{M}^\star$, the best model in $\mathbb{L}(\mathcal{H})$.

When this happens, one does not lose all target hypotheses of $\mathcal{H}$ when learning on $\hat{\mathcal{M}}$, so no bias is introduced. In the next section, we develop bounds for types I, II, III, and IV estimation errors on $\hat{\mathcal{M}}$, which also support this paradigm.

2.3 Convergence of estimation errors on $\hat{\mathcal{M}}$

The type III estimation error depends solely on $\hat{\mathcal{M}}$, while types I, II, and IV depend on $\hat{\mathcal{M}}$, but also on the choice of algorithm $\mathbb{A}$ employed to learn a hypothesis $\hat{h}^{\mathbb{A}}_{\hat{\mathcal{M}}} \in \hat{\mathcal{M}}$ (cf. (1.16) and (1.23)). In this section, we consider two possible algorithms, described in Figure 1.19 as learning by reusing, in which we consider an ERM hypothesis of $\hat{\mathcal{M}}$ under sample $\mathcal{D}_N$, and learning with independent sample, in which we consider an ERM hypothesis of $\hat{\mathcal{M}}$ under sample $\tilde{\mathcal{D}}_M$, independent of $\mathcal{D}_N$. We start by discussing in detail the case of an independent sample, and then briefly discuss learning by reusing.

2.3.1 Learning with independent sample

Bounds for types I and II estimation errors when learning on a random model with a sample independent of the one employed to compute such random model, may be obtained when there is a bound for them on each $\mathcal{M} \in \mathbb{L}(\mathcal{H})$ under the independent sample. This is the content of Theorem 2.10.

Theorem 2.10. *Fix a bounded loss function. Assume we are learning with an independent sample $\tilde{\mathcal{D}}_M$, and that for each $\epsilon > 0$ there exist sequences $\{B^I_{M,\epsilon} : M \geq 1\}$ and $\{B^{II}_{M,\epsilon} : M \geq 1\}$ of positive real-valued increasing functions with domain $\mathbb{Z}_+$ satisfying*

$$\lim_{M\to\infty} B^I_{M,\epsilon}(k) = \lim_{M\to\infty} B^{II}_{M,\epsilon}(k) = 0,$$

for all $\epsilon > 0$ and $k \in \mathbb{Z}_+$ fixed, such that

$$\mathbb{P}\left(\sup_{h\in\mathcal{M}} \left|L_{\tilde{\mathcal{D}}_M}(h) - L(h)\right| > \epsilon\right) \leq B^I_{M,\epsilon}(d_{VC}(\mathcal{M})) \text{ and} \tag{2.16}$$

$$\mathbb{P}\left(L(\hat{h}^{\tilde{\mathcal{D}}_M}_{\mathcal{M}}) - L(h^\star_{\mathcal{M}}) > \epsilon\right) \leq B^{II}_{M,\epsilon}(d_{VC}(\mathcal{M})),$$

for all $\mathcal{M} \in \mathbb{L}(\mathcal{H})$. Let $\hat{\mathcal{M}} \in \mathbb{L}(\mathcal{H})$ be a random model learned by $\mathbb{M}_{\mathbb{L}(\mathcal{H})}$. Then, for any $\epsilon > 0$,

$$\textbf{(I)}\ \mathbb{P}\left(\sup_{h\in\hat{\mathcal{M}}} \left|L_{\tilde{\mathcal{D}}_M}(h) - L(h)\right| > \epsilon\right) \leq \mathbb{E}\left[B^I_{M,\epsilon}(d_{VC}(\hat{\mathcal{M}}))\right] \leq B^I_{M,\epsilon}\left(d_{VC}(\mathbb{L}(\mathcal{H}))\right)$$

and

$$\textbf{(II)}\ \mathbb{P}\left(L(\hat{h}^{\tilde{\mathcal{D}}_M}_{\hat{\mathcal{M}}}) - L(h^\star_{\hat{\mathcal{M}}}) > \epsilon\right) \leq \mathbb{E}\left[B^{II}_{M,\epsilon}(d_{VC}(\hat{\mathcal{M}}))\right] \leq B^{II}_{M,\epsilon}\left(d_{VC}(\mathbb{L}(\mathcal{H}))\right),$$

in which the expectations are over all samples $\mathcal{D}_N$, from which $\hat{\mathcal{M}}$ is calculated. Since $d_{VC}(\mathbb{L}(\mathcal{H})) < \infty$, both probabilities above converge to zero when $M \longrightarrow \infty$.

Proof. We first note that

$$\begin{aligned}\mathbb{P}\left(\sup_{h\in\hat{\mathcal{M}}}\left|L_{\tilde{\mathcal{D}}_M}(h)-L(h)\right|>\epsilon\right)&=\mathbb{E}\left(\mathbb{P}\left(\sup_{h\in\hat{\mathcal{M}}}\left|L_{\tilde{\mathcal{D}}_M}(h)-L(h)\right|>\epsilon\middle|\hat{\mathcal{M}}\right)\right)\\&=\sum_{i\in\mathcal{J}}\mathbb{P}\left(\sup_{h\in\hat{\mathcal{M}}}\left|L_{\tilde{\mathcal{D}}_M}(h)-L(h)\right|>\epsilon\middle|\hat{\mathcal{M}}=\mathcal{M}_i\right)\mathbb{P}(\hat{\mathcal{M}}=\mathcal{M}_i)\\&=\sum_{i\in\mathcal{J}}\mathbb{P}\left(\sup_{h\in\mathcal{M}_i}\left|L_{\tilde{\mathcal{D}}_M}(h)-L(h)\right|>\epsilon\middle|\hat{\mathcal{M}}=\mathcal{M}_i\right)\mathbb{P}(\hat{\mathcal{M}}=\mathcal{M}_i).\end{aligned}\tag{2.17}$$

Fix $\mathcal{M}\in\mathbb{L}(\mathcal{H})$ with $\mathbb{P}(\hat{\mathcal{M}}=\mathcal{M})>0$. We claim that

$$\mathbb{P}\left(\sup_{h\in\mathcal{M}}\left|L_{\tilde{\mathcal{D}}_M}(h)-L(h)\right|>\epsilon\middle|\hat{\mathcal{M}}=\mathcal{M}\right)=\mathbb{P}\left(\sup_{h\in\mathcal{M}}\left|L_{\tilde{\mathcal{D}}_M}(h)-L(h)\right|>\epsilon\right).\tag{2.18}$$

Indeed, since $\tilde{\mathcal{D}}_M$ is independent of $\mathcal{D}_N$, the event

$$\left\{\sup_{h\in\mathcal{M}}\left|L_{\tilde{\mathcal{D}}_M}(h)-L(h)\right|>\epsilon\right\}$$

is independent of $\{\hat{\mathcal{M}}=\mathcal{M}\}$, as the former depends solely on $\tilde{\mathcal{D}}_M$, and the latter solely on $\mathcal{D}_N$. Hence, by applying bound (2.16) to each positive probability in the sum (2.17), we obtain that

$$\begin{aligned}\mathbb{P}\left(\sup_{h\in\hat{\mathcal{M}}}\left|L_{\tilde{\mathcal{D}}_M}(h)-L(h)\right|>\epsilon\right)&\leq\sum_{i\in\mathcal{J}}B^I_{M,\epsilon}(d_{VC}(\mathcal{M}_i))\mathbb{P}(\hat{\mathcal{M}}=\mathcal{M}_i)\\&=\mathbb{E}\left(B^I_{M,\epsilon}(d_{VC}(\hat{\mathcal{M}}))\right)\leq B^I_{N,\epsilon}(d_{VC}(\mathbb{L}(\mathcal{H}))),\end{aligned}$$

as desired, in which the last inequality follows from the fact that $B^I_{M,\epsilon}$ is an increasing function and $d_{VC}(\mathbb{L}(\mathcal{H}))=\max_{\mathcal{M}\in\mathbb{L}(\mathcal{H})}d_{VC}(\mathcal{M})$.

The bound for type II estimation error may be obtained similarly, since

$$\begin{aligned}\mathbb{P}\left(L(\hat{h}^{\tilde{\mathcal{D}}_M}_{\hat{\mathcal{M}}})-L(h^\star_{\hat{\mathcal{M}}})>\epsilon\right)&=\mathbb{E}\left(\mathbb{P}\left(L(\hat{h}^{\tilde{\mathcal{D}}_M}_{\hat{\mathcal{M}}})-L(h^\star_{\hat{\mathcal{M}}})>\epsilon\middle|\hat{\mathcal{M}}\right)\right)\\&=\sum_{i\in\mathcal{J}}\mathbb{P}\left(L(\hat{h}^{\tilde{\mathcal{D}}_M}_{\hat{\mathcal{M}}})-L(h^\star_{\hat{\mathcal{M}}})>\epsilon\middle|\hat{\mathcal{M}}=\mathcal{M}_i\right)\mathbb{P}(\hat{\mathcal{M}}=\mathcal{M}_i)\\&=\sum_{i\in\mathcal{J}}\mathbb{P}\left(L(\hat{h}^{\tilde{\mathcal{D}}_M}_{\mathcal{M}_i})-L(h^\star_{\mathcal{M}_i})>\epsilon\middle|\hat{\mathcal{M}}=\mathcal{M}_i\right)\mathbb{P}(\hat{\mathcal{M}}=\mathcal{M}_i)\\&=\sum_{i\in\mathcal{J}}\mathbb{P}\left(L(\hat{h}^{\tilde{\mathcal{D}}_M}_{\mathcal{M}_i})-L(h^\star_{\mathcal{M}_i})>\epsilon\right)\mathbb{P}(\hat{\mathcal{M}}=\mathcal{M}_i),\end{aligned}$$

and $B^{II}_{M,\epsilon}(d_{VC}(\mathcal{M}_i))$ is a bound for the probabilities inside the sum by (2.16). The assertion that types I and II estimation errors are asymptotically zero when $d_{VC}(\mathbb{L}(\mathcal{H}))<\infty$ is immediate from the established bounds. □

Our definition of $\hat{\mathcal{M}}$ ensures that it is going to have the smallest VC dimension under the constraint that it is a global minimum of $\mathbb{L}(\mathcal{H})$ (cf. Definition 1.17). As the quantities inside the expectations of Theorem 2.10 are increasing functions of VC dimension, fixed ϵ and M, we tend to have smaller expectations, thus tighter bounds for types I and II estimation errors.

Furthermore, it follows from Theorem 2.10 that the sample complexity needed to learn on $\hat{\mathcal{M}}$ is at most that of $d_{VC}(\mathbb{L}(\mathcal{H}))$. This implies that this complexity is at most that of $\mathcal{H}$, but may be much lesser if $d_{VC}(\mathbb{L}(\mathcal{H})) \ll d_{VC}(\mathcal{H})$. We conclude that the bounds for the tail probabilities of types I and II estimation errors on $\hat{\mathcal{M}}$ are tighter than that on $\mathcal{H}$ (cf. Corollaries A.9 and A.19), and the sample complexity needed to learn on $\hat{\mathcal{M}}$ is at most that of $\mathbb{L}(\mathcal{H})$, and not of $\mathcal{H}$.

However, even when these inequalities guarantee the consistency of $\hat{\mathcal{M}}$ regarding types I and II estimation errors, it is still necessary to check that types III and IV estimation errors are small to attest the consistency of learning on $\hat{\mathcal{M}}$: if $L(h^\star_{\hat{\mathcal{M}}})$ is too greater than $L(h^\star)$, well estimating $h^\star_{\hat{\mathcal{M}}}$ (small type II estimation error) is useless, so having small types I and II estimation errors is not enough to properly approximate $h^\star$, that is the main objective of the learning problem.

A bound for type III estimation error may be obtained using methods similar to that we employed to prove Theorem 2.8. As in that theorem, the bound for type III estimation error depends on $\epsilon^\star$, on bounds for type I estimation error under each training and validation sample, and on $\mathbb{L}(\mathcal{H})$, more specifically, on its VC dimension and number of maximal elements. To ease notation, we denote $\epsilon \vee \epsilon^\star := \max\{\epsilon, \epsilon^\star\}$ for any $\epsilon > 0$.

Theorem 2.11. *Assume the premises of Theorem 2.8 are in force. Let $\hat{\mathcal{M}} \in \mathbb{L}(\mathcal{H})$ be a random model learned by $\mathbb{M}_{\mathbb{L}(\mathcal{H})}$. Then, for any $\epsilon > 0$,*

$$\textbf{\textit{(III)}}\ \mathbb{P}\left(L(h^\star_{\hat{\mathcal{M}}}) - L(h^\star) > \epsilon\right) \leq m\, \mathfrak{m}(\mathbb{L}(\mathcal{H}))\left[B_{N,(\epsilon\vee\epsilon^\star)/8}(d_{VC}(\mathbb{L}(\mathcal{H}))) + \hat{B}_{N,(\epsilon\vee\epsilon^\star)/4}(d_{VC}(\mathbb{L}(\mathcal{H})))\right].$$

In particular,

$$\lim_{N\to\infty} \mathbb{P}\left(L(h^\star_{\hat{\mathcal{M}}}) - L(h^\star) > \epsilon\right) = 0,$$

for any $\epsilon > 0$.

Proof. We first show that

$$\mathbb{P}\left(L(h^\star_{\hat{\mathcal{M}}}) - L(h^\star) > \epsilon\right) \leq \mathbb{P}\left(\max_{i\in\mathcal{J}} \left|\hat{L}(\mathcal{M}_i) - L(\mathcal{M}_i)\right| > (\epsilon \vee \epsilon^\star)/2\right). \tag{2.19}$$

If $\epsilon \leq \epsilon^\star$ then, by the inclusion of events (2.7) in the proof of Proposition 2.6, we have that

$$\left\{\max_{i\in\mathcal{J}} \left|\hat{L}(\mathcal{M}_i) - L(\mathcal{M}_i)\right| < (\epsilon \vee \epsilon^\star)/2\right\} \subset \left\{L(\hat{\mathcal{M}}) = L(\mathcal{M}^\star)\right\} \subset \left\{L(h^\star_{\hat{\mathcal{M}}}) - L(h^\star) < \epsilon\right\}, \tag{2.20}$$

since $L(h^\star_{\hat{\mathcal{M}}}) = L(\hat{\mathcal{M}})$ and $L(h^\star_{\mathcal{M}^\star}) = L(\mathcal{M}^\star)$, so (2.19) follows in this case.

Now, if $\epsilon > \epsilon^\star$ and $\max_{i\in\mathcal{J}} \left|\hat{L}(\mathcal{M}_i) - L(\mathcal{M}_i)\right| < \epsilon/2$, then

$$\begin{aligned} L(\hat{\mathcal{M}}) - L(\mathcal{M}^\star) &= [L(\hat{\mathcal{M}}) - \hat{L}(\mathcal{M}^\star)] - [L(\mathcal{M}^\star) - \hat{L}(\mathcal{M}^\star)] \\ &\leq [L(\hat{\mathcal{M}}) - \hat{L}(\hat{\mathcal{M}})] - [L(\mathcal{M}^\star) - \hat{L}(\mathcal{M}^\star)] \\ &\leq \epsilon/2 + \epsilon/2 = \epsilon, \end{aligned}$$

in which the first inequality follows from the fact that the minimum of $\hat{L}$ is attained at $\hat{\mathcal{M}}$, and the last inequality follows from $\max_{i\in\mathcal{J}} \left|\hat{L}(\mathcal{M}_i) - L(\mathcal{M}_i)\right| < \epsilon/2$. Since $L(\hat{\mathcal{M}}) - L(\mathcal{M}^\star) = L(h^\star_{\hat{\mathcal{M}}}) - L(h^\star)$, we also have the inclusion of events

$$\left\{ \max_{i\in\mathcal{J}} \left|\hat{L}(\mathcal{M}_i) - L(\mathcal{M}_i)\right| < (\epsilon \vee \epsilon^\star)/2 \right\} \subset \left\{ L(h^\star_{\hat{\mathcal{M}}}) - L(h^\star) < \epsilon \right\}, \tag{2.21}$$

when $\epsilon > \epsilon^\star$. From (2.20) and (2.21) follows (2.19), as desired.

Substituting $\epsilon^\star$ by $\epsilon \vee \epsilon^\star$ in (2.12) we obtain

$$\begin{aligned} &\mathbb{P}\left(\max_{i\in\mathcal{J}} \left|L(\mathcal{M}_i) - \hat{L}(\mathcal{M}_i)\right| \geq (\epsilon \vee \epsilon^\star)/2 \right) \leq \\ &\quad m\,|\mathrm{Max}\,\mathbb{L}(\mathcal{H})| \left[B_{N,(\epsilon\vee\epsilon^\star)/8}(d_{VC}(\mathbb{L}(\mathcal{H}))) + \hat{B}_{N,(\epsilon\vee\epsilon^\star)/4}(d_{VC}(\mathbb{L}(\mathcal{H}))) \right]. \end{aligned} \tag{2.22}$$

The result follows combining (2.19) and (2.22). □

Remark 2.12. *Type III estimation error, and its bound presented in Theorem 2.11, do not depend on the algorithm $\mathbb{A}$ employed to learn on $\hat{\mathcal{M}}$, hence this theorem is true for both frameworks in Figure 1.19, holding also when learning by reusing.*

On the one hand, by definition of $\epsilon^\star$, if $\epsilon < \epsilon^\star$, then type III estimation error is lesser than ϵ if, and only if, $L(\hat{\mathcal{M}}) = L(\mathcal{M}^\star)$, so this error is actually zero, and the result of Theorem 2.8 is a bound for type III estimation error in this case. On the other hand, if $\epsilon > \epsilon^\star$, one way of having type III estimation error lesser than ϵ is to have the estimated error of each $\mathcal{M}$ at a distance at most $\epsilon/2$ from its real error and, as can be inferred from the proof of Theorem 2.8, this can be accomplished if one has type I estimation error not greater than a fraction of ϵ under each training and validation sample considered, so a modification of Theorem 2.8 also applies to this case.

Finally, as the tail probability of type IV estimation error may be bounded by the following inequality, involving the tail probabilities of types II and III estimation errors,

$$\begin{aligned} \textbf{(IV)}\ &\mathbb{P}\left(L(\hat{h}^{\tilde{D}_M}_{\hat{\mathcal{M}}}) - L(h^\star) > \epsilon \right) \\ &\leq \mathbb{P}\left(L(\hat{h}^{\tilde{D}_M}_{\hat{\mathcal{M}}}) - L(h^\star_{\hat{\mathcal{M}}}) > \epsilon/2 \right) + \mathbb{P}\left(L(h^\star_{\hat{\mathcal{M}}}) - L(h^\star) > \epsilon/2 \right), \end{aligned} \tag{2.23}$$

a bound for (2.23) is a direct consequence of Theorems 2.10 and 2.11.

Corollary 2.13. *Assume the premises of Theorem 2.8 and 2.10 are in force. Let $\hat{\mathcal{M}} \in \mathbb{L}(\mathcal{H})$*

be a random model learned by $\mathbb{M}_{\mathbb{L}(\mathcal{H})}$. *Then, for any* $\epsilon > 0$,

$$
\begin{aligned}
&\textbf{\textit{(IV)}}\ \mathbb{P}\left(L(\hat{h}^{\tilde{D}_M}_{\hat{\mathcal{M}}}) - L(h^\star) > \epsilon\right) \\
&\leq \mathbb{E}\left[B^{II}_{M,\epsilon/2}(d_{VC}(\hat{\mathcal{M}}))\right] + m\ \mathfrak{m}(\mathbb{L}(\mathcal{H}))\left[B_{N,(\epsilon/2\vee\epsilon^\star)/8}(d_{VC}(\mathbb{L}(\mathcal{H}))) + \hat{B}_{N,(\epsilon/2\vee\epsilon^\star)/4}(d_{VC}(\mathbb{L}(\mathcal{H})))\right] \\
&\leq B^{II}_{M,\epsilon/2}(d_{VC}(\mathbb{L}(\mathcal{H}))) + m\ \mathfrak{m}(\mathbb{L}(\mathcal{H}))\left[B_{N,(\epsilon/2\vee\epsilon^\star)/8}(d_{VC}(\mathbb{L}(\mathcal{H}))) + \hat{B}_{N,(\epsilon/2\vee\epsilon^\star)/4}(d_{VC}(\mathbb{L}(\mathcal{H})))\right].
\end{aligned}
$$

In particular,

$$
\lim_{\substack{N\to\infty \\ M\to\infty}} \mathbb{P}\left(L(\hat{h}^{\tilde{D}_M}_{\hat{\mathcal{M}}}) - L(h^\star) > \epsilon\right) = 0,
$$

for any $\epsilon > 0$.

From Theorems 2.8, 2.9, 2.10 and 2.11, and Corollary 2.13, follow the consistency of the Model Selection framework given by selecting $\hat{\mathcal{M}}$ via $\mathbb{M}_{\mathbb{L}(\mathcal{H})}$ and learning on it with an independent sample, when we consider $\hat{L}$ given by k-fold cross-validation or an independent validation sample. We state this result in the next corollary.

Corollary 2.14. *Assume the loss function is bounded. The Model Selection framework given by*

(a) estimating $L(\mathcal{M})$ *by k-fold cross validation with a fixed k or by an independent validation sample,*

(b) selecting $\hat{\mathcal{M}}$ *via* $\mathbb{M}_{\mathbb{L}(\mathcal{H})}$,

(c) and learning with an independent sample on $\hat{\mathcal{M}}$,

is consistent.

Proof. If $\hat{L}(\mathcal{M})$ is estimated by a validation sample or via k-fold cross-validation, the result follows from Theorems 2.8, 2.9, 2.10 and 2.11, and Corollary 2.13, since the bounds $B_{N,\epsilon}$, $\hat{B}_{N,\epsilon}$, $B^I_{M,\epsilon}$ and $B^{II}_{M,\epsilon}$ follow from classical VC theory applied to the independent training and validation samples, and independent sample $\tilde{D}_M$ (by Corollaries A.9 and A.19). □

Remark 2.15. *Actually, when the loss function is bounded, types I, II, III, and IV estimation errors converge to zero not only in probability, but also almost surely, since, by Corollaries A.9 and A.19, the functions* $B_{N,\epsilon}$, $\hat{B}_{N,\epsilon}$, $B^I_{M,\epsilon}$ *and* $B^{II}_{M,\epsilon}$ *are exponential on* N *and* M *hence, by Borel-Cantelli Lemma (cf. Lemma B.14), the convergences also hold almost surely.*

From the results established, follow that the Learning Space plays an important role on the rate of convergence of $\mathbb{P}(L(\hat{\mathcal{M}}) = L(\mathcal{M}^\star))$ to one, and of the estimation errors to zero, through $\epsilon^\star$ and $d_{VC}(\mathcal{M}^\star)$. Moreover, these results also shed light on manners of improving the quality of the learning, i.e., decreasing the estimation errors, specially type IV, when the sample size is fixed. We now discuss some implications of the results above.

First, since $\hat{\mathcal{M}}$ converges to $\mathcal{M}^\star$ with probability one,

$$
\mathbb{E}(G(\hat{\mathcal{M}})) \xrightarrow{N\to\infty} G(\mathcal{M}^\star),
$$

by the Dominated Convergence Theorem, in which $G : \mathbb{L}(\mathcal{H}) \mapsto \mathbb{R}$ is any real-valued function, since the domain of G is finite. The convergence of $\mathbb{E}(G(\hat{\mathcal{M}}))$ ensures that the expectations of functions of $\hat{\mathcal{M}}$ on the right-hand side of inequalities of Theorem 2.10 and Corollary 2.13 tend to the same functions evaluated at $\mathcal{M}^\star$, when N tends to infinity. Hence, if one was able to isolate $h^\star$ within a model $\mathcal{M}^\star$ with small VC dimension, the bounds for types I, II and IV estimation errors will tend to be tighter for a sample of a given size $N + M$.

Second, if the MDE of $\mathbb{L}(\mathcal{H})$ under P is great, then we need less precision when estimating $L(\mathcal{M})$ for $L(\hat{\mathcal{M}})$ to be equal to $L(\mathcal{M}^\star)$, and for types III and IV estimation errors to be lesser than a $\epsilon \ll \epsilon^\star$ with high probability, so fewer samples are needed to learn a model as good as $\mathcal{M}^\star$ and to have lesser types III and IV estimation errors. Moreover, the sample complexity to learn this model is that of the most complex model in $\mathbb{L}(\mathcal{H})$, hence is at most the complexity of a model with VC dimension $d_{VC}(\mathbb{L}(\mathcal{H}))$, which may be lesser than that of $\mathcal{H}$.

Therefore, by embedding into $\mathbb{L}(\mathcal{H})$ all prior information about $h^\star$ and P, seeking to increase $\epsilon^\star$ and decrease $d_{VC}(\mathcal{M}^\star)$, one may, with a given sample of size $N + M$, better learn on $\mathcal{H}$, that is, better approximate $h^\star$ by a $\hat{h}^{\tilde{\mathcal{D}}_M}_{\hat{\mathcal{M}}}$ (small type IV estimation error). Hence, the results of this section also support that, by properly modeling the Learning Space, one may better learn on $\mathcal{H}$, which means having small type IV estimation error.

By the deductions above, under the framework detailed in Corollary 2.14, it follows that all estimation errors converge in probability to zero and that $\hat{\mathcal{M}}$ tends to $\mathcal{M}^\star$ with probability one, when N and M tend to infinity. However, we are not able, by making use of the bounds provided by VC theory and extended to $\hat{\mathcal{M}}$ in this case, to find bounds for these estimation errors which do not depend on $\epsilon^\star$, and thus on P. In other words, we have established the distribution-free consistency of the framework, but not a distribution-free rate to the considered convergences.

Although not distribution-free, the convergences proved reflect an important property of our approach, which may have been overlooked by other methods: the sample complexity does depend on the target hypotheses, in this case through the target model. If one can isolate a target hypothesis inside a *simple* model (see Figure 1.16) such that $\epsilon^\star$ is *great*, then *few* samples are needed to *properly* estimate this target, independently of its "complexity", as $\hat{\mathcal{M}}$ would be equal to $\mathcal{M}^\star$ with high probability for a relatively small sample, as $\epsilon^\star$ is large, and types I and II estimation errors on $\hat{\mathcal{M}}$ would probably be *small*, as it is with high probability equal to $\mathcal{M}^\star$, which is *simple*.

Hence, without constraining beforehand the hypotheses space $\mathcal{H}$, which contains all hypotheses one is willing to consider, and without gathering more samples to increase a sample of size $N + M$, one could, theoretically, still estimate $h^\star$ by a hypothesis which well generalizes, by properly modeling $\mathbb{L}(\mathcal{H})$. Such a modeling should be done by embedding into $\mathbb{L}(\mathcal{H})$ all prior information about $h^\star$, P and the practical problem at hand.

We illustrate a case where it is possible to better estimate a target hypothesis when learning via Learning Spaces at the cost of computational power.

Learning with independent sample on the Partition Lattice Learning Space

Let $\mathcal{H} = \{\mathcal{X} \mapsto \{0,1\}\}$ be the set of all binary functions with domain $\mathcal{X}$, $|\mathcal{X}| < \infty$, let $\mathbb{L}(\mathcal{H})$ be the respective Partition Lattice Learning Space (cf. Example 1.12), and let $\mathbb{L}_2(\mathcal{H})$ be the models in $\mathbb{L}(\mathcal{H})$ with VC dimension at most 2 (cf. (2.15)).

We will compare the bounds for type IV estimation error when learning with independent sample on $\mathbb{L}_2(\mathcal{H})$ (cf. Corollary 2.13), with samples of size N and $M = N$, with classical VC theory bounds for type II estimation error when learning directly on $\mathcal{H}$, by considering the estimator $\hat{h}^{\mathcal{D}_{2N}}$, an ERM hypothesis in $\mathcal{H}$ of the whole sample of size $2N$ (cf. (2.2)). These errors, depicted in Figure 1.18, are the effective errors committed when learning with independent sample on $\mathbb{L}_2(\mathcal{H})$, and when learning as in classical VC theory.

On the one hand, by Corollary A.19 in Appendix A, we have that

$$\mathbb{P}\left(L(\hat{h}^{\mathcal{D}_{2N}}) - L(h^\star) > \epsilon\right) \leq 8\exp\left\{d_{VC}(\mathcal{H})\left(1 + \ln\frac{2N}{d_{VC}(\mathcal{H})}\right) - 2N\frac{\epsilon^2}{128}\right\}. \tag{2.24}$$

On the other hand, by Corollaries 2.13, A.8 and A.19, by learning with independent sample on $\mathbb{L}_2(\mathcal{H})$, when the independent sample has size N, the validation sample has size cN and the training sample has size $(1-c)N$, with $0 < c < 1/2$, it follows that

$$\begin{aligned}\mathbb{P}\left(L(\hat{h}_{\hat{\mathcal{M}}}(\tilde{\mathcal{D}}_N)) - L(h^\star) > \epsilon\right) \leq\ & 8\exp\left\{2\left(1 + \ln\frac{N}{2}\right) - N\frac{\epsilon^2}{512}\right\} \\ & + 8\left(2^{d_{VC}(\mathcal{H})-1} - 1\right)\left[\exp\left\{2\left(1 + \ln\frac{cN}{2}\right) - cN\frac{(\epsilon/2 \vee \epsilon^\star)^2}{512}\right\}\right. \\ & \left. + \exp\left\{2\left(1 + \ln\frac{(1-c)N}{2}\right) - (1-c)N\frac{(\epsilon/2 \vee \epsilon^\star)^2}{2048}\right\}\right]. \end{aligned} \tag{2.25}$$

In Figure 2.2, we present, for selected values of $d_{VC}(\mathcal{H})$, ϵ and $\epsilon^\star$, the value of N such that the bounds (2.24) and (2.25) are equal to 0.05, considering $c = 0.2$. We first note that, in any case, this value of N is of order at least 10^6, as is often the case with the pessimistic distribution-free bounds of VC theory. Nevertheless, we see that the bound for type IV estimation error is actually tighter when $2\epsilon \leq \epsilon^\star$, meaning that, for ϵ small enough, one needs fewer samples to properly estimate $h^\star$ when learning with an independent sample on $\mathbb{L}_2(\mathcal{H})$, when compared with learning via ERM on $\mathcal{H}$ directly.

However, there is a downside of learning on $\mathbb{L}_2(\mathcal{H})$: it is necessary an exhaustive search of it to calculate $\hat{\mathcal{M}}$. Hence, this example illustrates the following interesting paradigm: the lack of data may be mitigated by high computational power. Indeed, in this example, for a fixed sample of size $2N$, one may better estimate $h^\star$ by applying high computational power to learn on $\mathbb{L}_2(\mathcal{H})$.

Although the learning on $\mathbb{L}_2(\mathcal{H})$ could be a way to better learn with a sample of a given fixed size, one could better estimate with lesser computational power. For instance, if one considered the whole Partition Lattice Learning Space, and assumed that $d_{VC}(\hat{\mathcal{M}}) \approx$

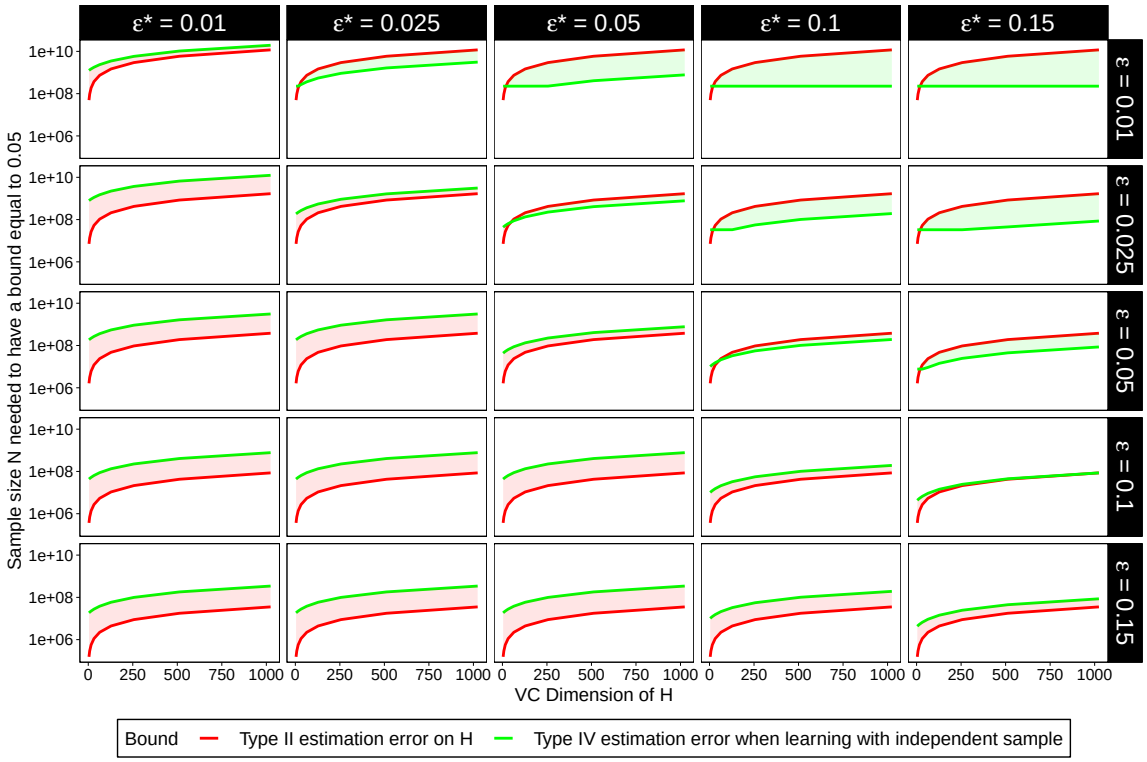

Figure 2.2: *Sample size N needed to have bounds (2.24) and (2.25) equal to 0.05 as a function of $d_{VC}(\mathcal{H})$, for distinct values of $\epsilon^\star$ (columns) and ϵ (lines), and $c = 0.2$. The curves of type II bound (2.24) are in red, and the ones of type IV bound (2.25) are in green. When the red curve is below the green one, we have a tighter bound for type II estimation error when learning directly on $\mathcal{H}$ with a sample of size $2N$, while when the green curve is below the red one, we have a tighter bound for type IV estimation error when learning with independent sample on $\mathbb{L}_2(\mathcal{H})$, with a training sample of size $0.8N$, a validation sample of size $0.2N$, and an independent sample of size N. To aid in the visualization, we painted the space between the two curves in green when the bound of type IV estimation error (2.25) is tighter, and in red when the bound of type II estimation error (2.24) is tighter.*

$d_{VC}(\mathcal{M}^\star) = 2$, then he would have the following bound for type IV estimation error:

$$\begin{aligned}\mathbb{P}\left(L(\hat{h}_{\hat{\mathcal{M}}}(\tilde{\mathcal{D}}_N)) - L(h^\star) > \epsilon\right) \leq\ & 8\exp\left\{2\left(1+\ln\frac{N}{2}\right) - N\frac{\epsilon^2}{512}\right\} \\ & + 8\left[\exp\left\{d_{VC}(\mathcal{H})\left(1+\ln\frac{cN}{d_{VC}(\mathcal{H})}\right) - cN\frac{(\epsilon/2\vee\epsilon^\star)^2}{512}\right\}\right. \\ & \left. + \exp\left\{d_{VC}(\mathcal{H})\left(1+\ln\frac{(1-c)N}{d_{VC}(\mathcal{H})}\right) - (1-c)N\frac{(\epsilon/2\vee\epsilon^\star)^2}{2048}\right\}\right].\end{aligned} \tag{2.26}$$

In Figure 2.3, we present, for selected values of $d_{VC}(\mathcal{H})$, ϵ and $\epsilon^\star$, the value of N such that the bounds (2.24) and (2.26) are equal to 0.05, again considering $c = 0.2$. We see in this case that bound (2.26) may also be tighter than (2.24), but ϵ should be much lesser than $\epsilon^\star$, for instance, $5\epsilon < \epsilon^\star$. In Chapter 3, we present a non-exhaustive algorithm to learn on the

Partition Lattice Learning Space, which might be less complex than an exhaustive search of $\mathbb{L}_2(\mathcal{H})$, although is still quite complex. Hence, this is another example of a case in which lack of data may be mitigated by high computational power.

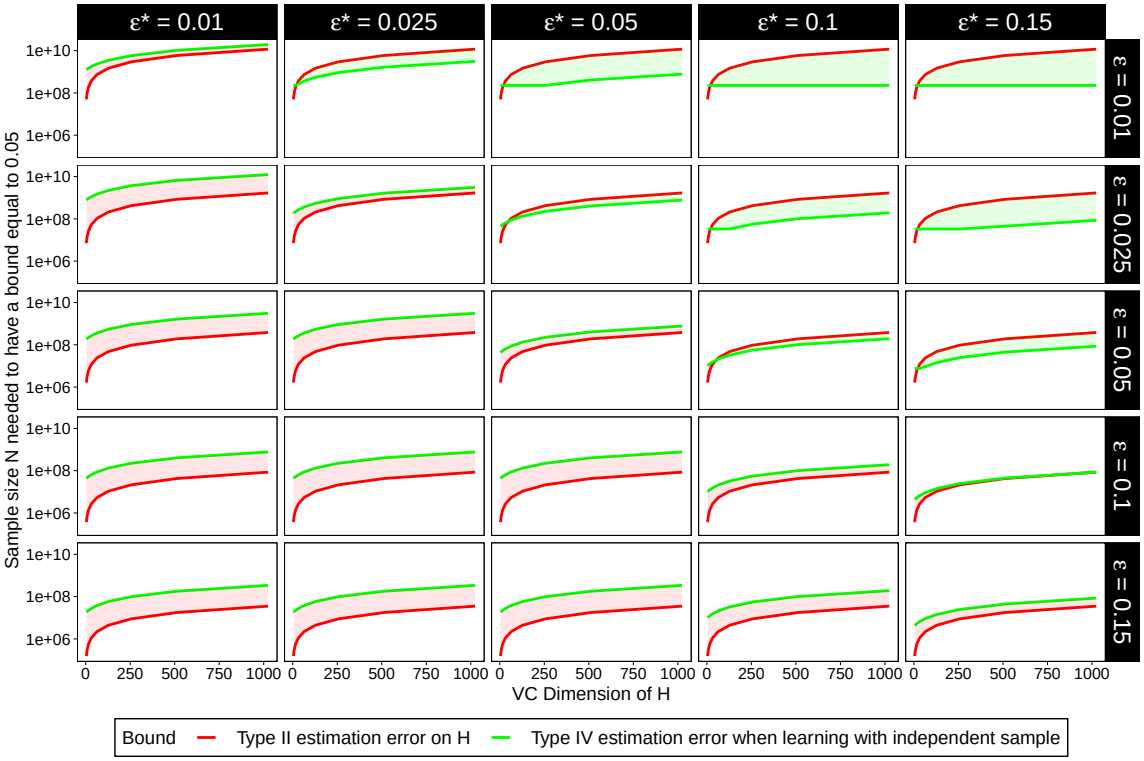

Figure 2.3: *Sample size N needed to have bounds (2.24) and (2.26) equal to 0.05 as a function of $d_{VC}(\mathcal{H})$, for distinct values of $\epsilon^\star$ (columns) and ϵ (lines), and $c = 0.2$. The curves of type II bound (2.24) are in red, and the ones of type IV bound (2.26) are in green. When the red curve is below the green one, we have a tighter bound for type II estimation error when learning directly on $\mathcal{H}$ with a sample of size $2N$, while when the green curve is below the red one, we have a tighter bound for type IV estimation error when learning with independent sample on $\mathbb{L}(\mathcal{H})$, the Partition Lattice Learning Space, with a training sample of size $0.8N$, a validation sample of size $0.2N$ and an independent sample of size N. To aid in the visualization, we painted the space between the two curves in green when the bound of type IV estimation error (2.26) is tighter, and in red when the bound of type II estimation error (2.24) is tighter.*

Remark 2.16. *The sample sizes obtained in classical VC theory, and also in our method, which are based on it, are known to be quite pessimistic, since the bounds are only meaningful, i.e., lesser than one, for very large values of N. Hence, although our method may perform better with a same sample of size $2N$ when compared to estimation via ERM on $\mathcal{H}$, it still needs a sample of a very great size to imply meaningful bounds in a distribution-free scenario. Better bounds may be obtained in a distribution dependent setting, for example by considering the Rademacher complexity [59] of the models in $\mathbb{L}(\mathcal{H})$, or assuming that the distribution P is in a certain class. In these distribution dependent situations, our method may also perform better, but with sample sizes of a much lesser order. We do not study, and leave as a topic for future researches, the development of bounds for type IV estimation error from a distribution dependent perspective, since it is out of the scope of this thesis.*

2.3.2 Learning by reusing

When learning by reusing, one is employing the same sample points to both estimate $\hat{\mathcal{M}}$ and learn a hypothesis $\hat{h}^{D_N}_{\hat{\mathcal{M}}} \in \hat{\mathcal{M}}$ from it, so there is a dependence between types I and II estimation errors and the events $\{\hat{\mathcal{M}} = \mathcal{M}\}$, $\mathcal{M} \in \mathbb{L}(\mathcal{H})$. Indeed, an equality like (2.18) may not be true in this case, that is, we may have

$$\mathbb{P}\left(\sup_{h\in\hat{\mathcal{M}}}\left|L_{D_N}(h) - L(h)\right| > \epsilon \middle| \hat{\mathcal{M}} = \mathcal{M}\right) \neq \mathbb{P}\left(\sup_{h\in\mathcal{M}}\left|L_{D_N}(h) - L(h)\right| > \epsilon\right),$$

since, conditioned on $\{\hat{\mathcal{M}} = \mathcal{M}\}$, not only the distribution of each sample point (X_l, Y_l), $l = 1, \dots, N$, changes, but also these points are now dependent: they must be such that $\hat{\mathcal{M}} = \mathcal{M}$, hence, cannot be independent. Therefore, the argument of the proof of Theorem 2.10 does not hold in this instance.

Nevertheless, since $\hat{\mathcal{M}}$ converges with probability one to $\mathcal{M}^\star$ by Theorem 2.8, we may obtain a bound for types I and II estimation errors when learning by reusing which depends on such bounds in $\mathcal{M}^\star$, and on the rate of convergence of $\hat{\mathcal{M}}$ to $\mathcal{M}^\star$.

Theorem 2.17. *Fix a bounded loss function. Assume we are learning by reusing and that, for each $\epsilon > 0$, there exist sequences $\{B^I_{N,\epsilon} : N \geq 1\}$ and $\{B^{II}_{N,\epsilon} : N \geq 1\}$ of positive real-valued increasing functions with domain $\mathbb{Z}_+$ satisfying*

$$\lim_{N\to\infty} B^I_{N,\epsilon}(k) = \lim_{N\to\infty} B^{II}_{N,\epsilon}(k) = 0,$$

for all $\epsilon > 0$ and $k \in \mathbb{Z}_+$ fixed, such that

$$\mathbb{P}\left(\sup_{h\in\mathcal{M}}\left|L_{D_N}(h) - L(h)\right| > \epsilon\right) \leq B^I_{N,\epsilon}(d_{VC}(\mathcal{M})) \text{ and}$$

$$\mathbb{P}\left(L(\hat{h}^{D_N}_{\mathcal{M}}) - L(h^\star_{\mathcal{M}}) > \epsilon\right) \leq B^{II}_{N,\epsilon}(d_{VC}(\mathcal{M})),$$

for all $\mathcal{M} \in \mathbb{L}(\mathcal{H})$. Let $\hat{\mathcal{M}} \in \mathbb{L}(\mathcal{H})$ be a random model learned by $\mathbb{M}_{\mathbb{L}(\mathcal{H})}$. Then, for any $\epsilon > 0$,

$$\textbf{\textit{(I)}}\ \mathbb{P}\left(\sup_{h\in\hat{\mathcal{M}}}\left|L_{D_N}(h) - L(h)\right| > \epsilon\right) \leq B^I_{N,\epsilon}(d_{VC}(\mathcal{M}^\star)) + \mathbb{P}\left(\hat{\mathcal{M}} \neq \mathcal{M}^\star\right)$$

and

$$\textbf{\textit{(II)}}\ \mathbb{P}\left(L(\hat{h}^{D_N}_{\hat{\mathcal{M}}}) - L(h^\star_{\hat{\mathcal{M}}}) > \epsilon\right) \leq B^{II}_{N,\epsilon}(d_{VC}(\mathcal{M}^\star)) + \mathbb{P}\left(\hat{\mathcal{M}} \neq \mathcal{M}^\star\right).$$

If conditions (2.10) of Theorem 2.8 are satisfied, both probabilities above converge to zero when $N \longrightarrow \infty$.

Proof. The bound for type I estimation error follows from the inequality

$$\begin{aligned}
&\mathbb{P}\left(\sup_{h\in\hat{\mathcal{M}}}\left|L_{\mathcal{D}_N}(h)-L(h)\right|>\epsilon\right)\\
&=\mathbb{P}\left(\sup_{h\in\hat{\mathcal{M}}}\left|L_{\mathcal{D}_N}(h)-L(h)\right|>\epsilon,\hat{\mathcal{M}}=\mathcal{M}^\star\right)+\mathbb{P}\left(\sup_{h\in\hat{\mathcal{M}}}\left|L_{\mathcal{D}_N}(h)-L(h)\right|>\epsilon,\hat{\mathcal{M}}\neq\mathcal{M}^\star\right)\\
&\leq\mathbb{P}\left(\sup_{h\in\mathcal{M}^\star}\left|L_{\mathcal{D}_N}(h)-L(h)\right|>\epsilon\right)+\mathbb{P}\left(\hat{\mathcal{M}}\neq\mathcal{M}^\star\right),
\end{aligned}$$

by noting that $B^I_{N,\epsilon}(d_{VC}(\mathcal{M}^\star))$ is a bound for the first probability. With a similar argument, we have the bound for type II estimation error. □

By inequality (2.23) and the bound for type III estimation error established in Theorem 2.11, that is also true when learning by reusing (cf. Remark 2.12), we have that the tail probability of type IV estimation error converges to zero as N tends to infinity, a result analogous to Corollary 2.13. Hence, it is consistent to learn by reusing.

Corollary 2.18. *Assume the loss function is bounded. The Model Selection framework given by*

(a) estimating $L(\mathcal{M})$ by k-fold cross validation with a fixed k or by an independent validation sample,

(b) selecting $\hat{\mathcal{M}}$ via $\mathbb{M}_{\mathbb{L}(\mathcal{H})}$,

(c) and learning by reusing on $\hat{\mathcal{M}}$,

is consistent.

Proof. If $\hat{L}(\mathcal{M})$ is estimated by a validation sample or via k-fold cross-validation, the result follows from Theorems 2.8, 2.11 and 2.17, and inequality (2.23), since the bounds $B_{N,\epsilon}$, $\hat{B}_{N,\epsilon}$, $B^I_{N,\epsilon}$ and $B^{II}_{N,\epsilon}$ follow from classical VC theory applied to the independent training and validation samples, and the whole sample $\mathcal{D}_N$ (by Corollaries A.9 and A.19). □

The bounds for types I and II estimation errors in Theorem 2.17 outline that, if $\mathcal{M}^\star$ has a *small* VC dimension and $\mathbb{M}_{\mathbb{L}(\mathcal{H})}$ is such that $\hat{\mathcal{M}}=\mathcal{M}^\star$ with *high* probability, then one can learn by reusing and still *properly* estimate on $\hat{\mathcal{M}}$. This result also supports the paradigm of, by properly modeling $\mathbb{L}(\mathcal{H})$ seeking to have the target $\mathcal{M}^\star$ with small VC dimension, one can better estimate with a fixed sample size. As was also the case for learning with an independent sample, we have established the distribution-free consistency of learning by reusing, but have not obtained a distribution-free rate for the convergence of the estimation errors.

A drawback of learning by reusing is the selection bias of $\hat{h}^{\mathcal{D}_N}_{\hat{\mathcal{M}}}$, which increases the risk of overfitting. This has been very well empirically studied in [33] for some specific cases, and may or may not be an issue, and further empirical studies are needed to better understand when it will be. Nevertheless, this approach could in theory be better when

the sample size available is not *great enough*, since the drawback of learning with an independent sample is the need for large sample sizes N and M (cf. Corollary 2.13).

2.4 Unbounded loss functions

When the loss function is unbounded, we need to reformulate the meaning of consistency. Indeed, we have to deviate a bit from the distribution-free framework, since, if random variable Z has *very heavy tails*, then the convergence of estimation errors may be too slow, i.e., not with exponential rate, or may not happen at all.

Heavy tail distributions are classically defined as those with a tail heavier than that of exponential distributions [57]. Nevertheless, in the context of learning, the tail weight of P should take into account the loss function ℓ. Hence, for $1 < p < \infty$ and a fixed hypotheses space $\mathcal{H}$, we measure the weight of the tails of distribution P by

$$\tau_p := \sup_{h \in \mathcal{H}} \frac{\left(\int_{\mathcal{Z}} \ell^p(z,h)\, dP(z)\right)^{\frac{1}{p}}}{\int_{\mathcal{Z}} \ell(z,h)\, dP(z)} = \sup_{h \in \mathcal{H}} \frac{L^p(h)}{L(h)},$$

in which $L^p(h) := \left(\int_{\mathcal{Z}} \ell^p(z,h)\, dP(z)\right)^{\frac{1}{p}}$. We omit the dependence of τ_p on ℓ, P and $\mathcal{H}$ to simplify notation, since they will be clear from context. The weight of the tails of distribution P may be defined based on τ_p, as follows. Our presentation is analogous to [149, Section 5.7].

Definition 2.19. *We say that distribution P on $\mathcal{H}$ under ℓ has:*

- *Light tails, if there exists a $p > 2$ such that $\tau_p < \infty$;*
- *Heavy tails, if there exists a $1 < p \leq 2$ such that $\tau_p < \infty$, but $\tau_p = \infty$ for all $p > 2$;*
- *Very heavy tails, if $\tau = \infty$ for all $p > 1$;*

In order to obtain bounds for the four estimation errors, we assume that P has at most heavy tails, which means there exists a $p > 1$, that can be lesser than 2, with

$$\tau_p < \tau^\star < \infty, \tag{2.27}$$

that is, P is in a class of distributions for which bound (2.27) holds. From now on, fix a $p > 1$ and a $\tau^\star$ such that (2.27) holds.

Besides the constraint (2.27) in the distribution tails, we also assume that the loss function is greater or equal to one: $\ell(z,h) \geq 1$ for all $z \in \mathcal{Z}, h \in \mathcal{H}$. This is done to ease the presentation, and without loss of generality, since it is enough to sum one to any unbounded loss function to have this property and, in doing so, not only the minimizers of L_{D_N} and L in each model in a $\mathbb{L}(\mathcal{H})$ remain the same, but also $\epsilon^\star$ does not change. Hence, by summing one to the loss, the estimated model $\hat{\mathcal{M}}$ and learned hypotheses from it do not change, and the result of the Model Selection framework is the same. We refer to Remark A.17 for the technical reason we choose to consider loss functions greater than one.

Finally, we assume that ℓ has a finite moment of order p, under P and under the

empirical measure, for all $h \in \mathcal{H}$. That is, defining[3]

$$L^p_{\mathcal{D}_N}(h) := \left(\frac{1}{N} \sum_{i=1}^{N} \ell^p(Z_i, h) \right)^{\frac{1}{p}}, \tag{2.28}$$

we assume that

$$\sup_{h \in \mathcal{H}} L^p_{\mathcal{D}_N}(h) < \infty \qquad \text{and} \qquad \sup_{h \in \mathcal{H}} L^p(h) < \infty, \tag{2.29}$$

in which the first inequality should hold with probability one, for all possible samples $\mathcal{D}_N$. Since the moments $L^p(h)$ are increasing in p, (2.29) actually implies (2.27), so (2.29) is the non-trivial constraint in distribution P. Although this is a deviation from the distribution-free framework, it is a mild constraint in distribution P, which ought to be satisfied by the distributions of data used in many applications of interest.

Indeed, on the one hand, the condition on L^p is usually satisfied for distributions observed in real data (see [149, Section 5.7] for examples with Normal, Uniform, and Laplacian distributions under the quadratic loss function). On the other hand, the condition on $L^p_{\mathcal{D}_N}$ is more a feature of the loss function, than of the distribution P, and can be guaranteed if one excludes from $\mathcal{H}$ some hypotheses with arbitrarily large loss in a way that $h^\star$ and $d_{VC}(\mathcal{H})$ remain the same (see Lemma A.11 and Remark A.16 for more details).

When the loss function is unbounded, besides the constraints in the moments of ℓ, under P and the empirical measure, we also have to consider variants of the estimation errors. Since $L(h)$ may be very large, having L arbitrarily close to $L_{\mathcal{D}_N}$, uniformly in a $\mathcal{M} \in \mathbb{L}(\mathcal{H})$, is not reasonable, since this difference is expected to be proportional to L, that is, the *arbitrarily close* concept should be relative to the value of L. In order to have a meaningful *distance* concept in this instance, we divide the estimation errors by L, implying that the closeness of an estimated loss to the respective out-of-sample error is relative to the out-of-sample error.

Hence, in place of the estimation errors, we consider the relative estimation errors:

$$\textbf{(I)} \begin{cases} \sup\limits_{h \in \hat{\mathcal{M}}} \left| \frac{L(h) - L_{\mathcal{D}_N}(h)}{L(h)} \right| \\ \sup\limits_{h \in \hat{\mathcal{M}}} \left| \frac{L(h) - L_{\tilde{\mathcal{D}}_M}(h)}{L(h)} \right| \end{cases} \qquad \textbf{(II)} \ \frac{L(\hat{h}^{\mathbb{A}}_{\hat{\mathcal{M}}}) - L(h^\star_{\hat{\mathcal{M}}})}{L(\hat{h}^{\mathbb{A}}_{\hat{\mathcal{M}}})}$$

$$\textbf{(III)} \ \frac{L(h^\star_{\hat{\mathcal{M}}}) - L(h^\star)}{L(h^\star_{\hat{\mathcal{M}}})} \qquad \textbf{(IV)} \ \frac{L(\hat{h}^{\mathbb{A}}_{\hat{\mathcal{M}}}) - L(h^\star)}{L(\hat{h}^{\mathbb{A}}_{\hat{\mathcal{M}}})}$$

where algorithm $\mathbb{A}$, and type I relative estimation error, are dependent on the estimation technique, that is either learning with independent sample or by reusing (cf. Figure 1.19).

[3] We elevate (2.28) to the $1/p$ power to be consistent with the definitions in Appendix A.

We are now in position to define consistency when the loss function is unbounded.

Definition 2.20. ***(Consistency for unbounded loss functions)*** *When the loss function is unbounded and (2.29) is satisfied, a Model Selection framework is consistent if it returns a random model $\hat{\mathcal{M}}$ and an estimated hypothesis $\hat{h}^{\mathbb{A}} \in \hat{\mathcal{M}}$ such that relative types I, II, III, and IV estimation errors of learning on it converge in probability to zero, and $\hat{\mathcal{M}}$ converges to $\mathcal{M}^\star$ with probability one, when the sample size tends to infinity.*

In order to establish the consistency of Model Selection via Learning Spaces for unbounded loss functions, we need to prove analogues of Theorems 2.8, 2.10, 2.11 and 2.17. Before starting the study of the convergence of $\hat{\mathcal{M}}$ to $\mathcal{M}^\star$, we state a result analogous to Proposition 2.4 about the convergence of relative type I and II estimation errors on $\mathcal{H}$, which is a consequence of Corollaries A.14 and A.20.

Proposition 2.21. *Assume the loss function is unbounded and P is such that (2.29) hold. Fixed a hypotheses space $\mathcal{H}$ with $d_{VC}(\mathcal{H}) < \infty$, there exist sequences $\{B^I_{N,\epsilon} : N \geq 1\}$ and $\{B^{II}_{N,\epsilon} : N \geq 1\}$ of positive real-valued increasing functions with domain $\mathbb{Z}_+$ satisfying*

$$\lim_{N\to\infty} B^I_{N,\epsilon}(k) = \lim_{N\to\infty} B^{II}_{N,\epsilon}(k) = 0,$$

for all $\epsilon > 0$ and $k \in \mathbb{Z}_+$ fixed, such that

$$\mathbb{P}\left(\sup_{h\in\mathcal{H}}\left|\frac{L_{\mathcal{D}_N}(h) - L(h)}{L(h)}\right| > \epsilon\right) \leq B^I_{N,\epsilon}(d_{VC}(\mathcal{H}))$$

$$\mathbb{P}\left(\frac{L(\hat{h}^{\mathcal{D}_N}) - L(h^\star)}{L(\hat{h}^{\mathcal{D}_N})} > \epsilon\right) \leq B^{II}_{N,\epsilon}(d_{VC}(\mathcal{H})).$$

Furthermore, the following holds:

$$\sup_{h\in\mathcal{H}}\left|\frac{L_{\mathcal{D}_N}(h) - L(h)}{L(h)}\right| \xrightarrow[N\to\infty]{a.s.} 0 \qquad \text{and} \qquad \frac{L(\hat{h}^{\mathcal{D}_N}) - L(h^\star)}{L(\hat{h}^{\mathcal{D}_N})} \xrightarrow[N\to\infty]{a.s.} 0.$$

The results of this section seek to establish the consistency in the case of unbounded loss functions, rather than obtain the tightest possible bounds. Hence, in some results, the simplicity of the bounds is preferred over its tightness, and tighter bounds may be readily obtained from the proofs.

2.4.1 Convergence to the target model

In order to show the convergence to the target model, we start by showing a result similar to Theorem 2.8.

Theorem 2.22. *Assume the loss function is unbounded and P is such that (2.29) hold. For each $\epsilon > 0$, let $\{B_{N,\epsilon} : N \geq 1\}$ and $\{\hat{B}_{N,\epsilon} : N \geq 1\}$ be sequences of positive real-valued*

increasing functions with domain $\mathbb{Z}_+$ *satisfying*

$$\lim_{N\to\infty} B_{N,\epsilon}(k) = \lim_{N\to\infty} \hat{B}_{N,\epsilon}(k) = 0,$$

for all $\epsilon > 0$ *and* $k \in \mathbb{Z}_+$ *fixed, and such that*

$$\max_j \mathbb{P}\left(\sup_{h\in\mathcal{M}} \left| \frac{L(h) - L_{D_N^{(j)}}(h)}{L(h)} \right| > \epsilon \right) \leq B_{N,\epsilon}(d_{VC}(\mathcal{M})) \text{ and}$$

$$\max_j \mathbb{P}\left(\sup_{h\in\mathcal{M}} \left| \frac{L(h) - \hat{L}^{(j)}(h)}{L(h)} \right| > \epsilon \right) \leq \hat{B}_{N,\epsilon}(d_{VC}(\mathcal{M})),$$

for all $\mathcal{M} \in \mathbb{L}(\mathcal{M})$*, recalling that* $L_{D_N^{(j)}}$ *and* $\hat{L}^{(j)}$ *represent the empirical error under the* j*-th training and validation samples, respectively. Let* $\hat{\mathcal{M}} \in \mathbb{L}(\mathcal{H})$ *be a random model learned by* $\mathbb{M}_{\mathbb{L}(\mathcal{H})}$*. Then,*

$$\mathbb{P}\left(L(\hat{\mathcal{M}}) \neq L(\mathcal{M}^\star) \right) \leq 2m\, \mathfrak{m}(\mathbb{L}(\mathcal{H})) \left[\hat{B}_{N,\frac{\delta(1-\delta)}{2}}(d_{VC}(\mathbb{L}(\mathcal{H}))) + B_{N,\frac{\delta(1-\delta)}{4}}(d_{VC}(\mathbb{L}(\mathcal{H}))) \right], \quad (2.30)$$

in which m *is the number of pairs considered to calculate* (2.8) *and*

$$\delta := \frac{\epsilon^\star}{2 \max_{i\in\mathcal{J}} L(\mathcal{M}_i)}.$$

Furthermore, if

$$\max_{\mathcal{M}\in\mathbb{L}(\mathcal{H})} \max_j \sup_{h\in\mathcal{M}} \left| \frac{L(h) - L_{D_N^{(j)}}(h)}{L(h)} \right| \xrightarrow[N\to\infty]{a.s.} 0 \text{ and} \quad (2.31)$$

$$\max_{\mathcal{M}\in\mathbb{L}(\mathcal{H})} \max_j \sup_{h\in\mathcal{M}} \left| \frac{L(h) - \hat{L}^{(j)}(h)}{L(h)} \right| \xrightarrow[N\to\infty]{a.s.} 0,$$

then

$$\lim_{N\to\infty} \mathbb{P}\left(\hat{\mathcal{M}} = \mathcal{M}^\star \right) = 1.$$

Proof. We claim that

$$1 - \delta < \frac{\hat{L}(\mathcal{M}_i)}{L(\mathcal{M}_i)} < 1 + \delta,\ \forall i \in \mathcal{J} \implies \max_{i\in\mathcal{J}} \left| \hat{L}(\mathcal{M}_i) - L(\mathcal{M}_i) \right| < \frac{\epsilon^\star}{2}. \quad (2.32)$$

Indeed, the left-hand side of (2.32) implies

$$\begin{cases} L(\mathcal{M}_i) - \hat{L}(\mathcal{M}_i) < \frac{\epsilon^\star L(\mathcal{M}_i)}{2 \max_{i\in\mathcal{J}} L(\mathcal{M}_i)} < \frac{\epsilon^\star}{2} \\ \hat{L}(\mathcal{M}_i) - L(\mathcal{M}_i) < \frac{\epsilon^\star L(\mathcal{M}_i)}{2 \max_{i\in\mathcal{J}} L(\mathcal{M}_i)} < \frac{\epsilon^\star}{2} \end{cases} \quad \forall i \in \mathcal{J},$$

as desired. In particular, it follows from inclusion (2.7) in the proof of Proposition 2.6 that

$$\mathbb{P}\left(L(\hat{\mathcal{M}}) \neq L(\mathcal{M}^\star)\right) \leq \mathbb{P}\left(\min_{i\in\mathcal{J}} \frac{\hat{L}(\mathcal{M}_i)}{L(\mathcal{M}_i)} \leq 1-\delta\right) + \mathbb{P}\left(\max_{i\in\mathcal{J}} \frac{\hat{L}(\mathcal{M}_i)}{L(\mathcal{M}_i)} \geq 1+\delta\right) \tag{2.33}$$

hence it is enough to bound both probabilities on the right-hand side of the expression above.

The first probability in (2.33) may be written as

$$\mathbb{P}\left(\max_{i\in\mathcal{J}} \frac{L(\mathcal{M}_i) - \hat{L}(\mathcal{M}_i)}{L(\mathcal{M}_i)} \geq \delta\right) \leq \sum_{j=1}^{m} \mathbb{P}\left(\max_{i\in\mathcal{J}} \frac{L(\mathcal{M}_i) - \hat{L}^{(j)}(\hat{h}_i^{(j)})}{L(\mathcal{M}_i)} \geq \delta\right), \tag{2.34}$$

in which the inequality follows from a union bound. Since $x \mapsto \frac{x-\alpha}{x}$ is increasing, and $L(\mathcal{M}_i) \leq L(\hat{h}_i^{(j)})$ for every $j = 1, \dots, m$, each probability in (2.34) is bounded by

$$\mathbb{P}\left(\max_{i\in\mathcal{J}} \frac{L(\hat{h}_i^{(j)}) - \hat{L}^{(j)}(\hat{h}_i^{(j)})}{L(\hat{h}_i^{(j)})} \geq \delta\right) \leq \mathbb{P}\left(\max_{i\in\mathcal{J}} \sup_{h\in\mathcal{M}_i} \left|\frac{L(h) - \hat{L}^{(j)}(h)}{L(h)}\right| \geq \delta\right). \tag{2.35}$$

We turn to the second probability in (2.33) which can be written as

$$\mathbb{P}\left(\max_{i\in\mathcal{J}} \frac{\hat{L}(\mathcal{M}_i) - L(\mathcal{M}_i)}{L(\mathcal{M}_i)} \geq \delta\right) \leq \sum_{j=1}^{m} \mathbb{P}\left(\max_{i\in\mathcal{J}} \frac{\hat{L}^{(j)}(\hat{h}_i^{(j)}) - L(\mathcal{M}_i)}{L(\mathcal{M}_i)} \geq \delta\right), \tag{2.36}$$

in which again the inequality follows from a union bound. In order to bound each probability in (2.36) we intersect its event with

$$\max_{i\in\mathcal{J}} \frac{L(\hat{h}_i^{(j)})}{L(\mathcal{M}_i)} \leq \frac{1}{1-\delta} \iff \max_{i\in\mathcal{J}} \frac{L(\hat{h}_i^{(j)}) - L(\mathcal{M}_i)}{L(\hat{h}_i^{(j)})} \leq \delta,$$

and its complement, to obtain

$$\begin{aligned}
&\mathbb{P}\left(\max_{i\in\mathcal{J}} \frac{\hat{L}^{(j)}(\hat{h}_i^{(j)}) - L(\mathcal{M}_i)}{L(\mathcal{M}_i)} \geq \delta\right) \leq \mathbb{P}\left(\max_{i\in\mathcal{J}} \frac{L(\hat{h}_i^{(j)}) - L(\mathcal{M}_i)}{L(\hat{h}_i^{(j)})} \geq \delta\right) \\
&+ \mathbb{P}\left(\max_{i\in\mathcal{J}} \left(\frac{L(\hat{h}_i^{(j)})}{L(\mathcal{M}_i)}\right) \frac{\hat{L}^{(j)}(\hat{h}_i^{(j)}) - L(\mathcal{M}_i)}{L(\hat{h}_i^{(j)})} \geq \delta, \max_{i\in\mathcal{J}} \frac{L(\hat{h}_i^{(j)})}{L(\mathcal{M}_i)} \leq \frac{1}{1-\delta}\right) \\
&\leq \mathbb{P}\left(\max_{i\in\mathcal{J}} \frac{L(\hat{h}_i^{(j)}) - L(\mathcal{M}_i)}{L(\hat{h}_i^{(j)})} \geq \delta\right) + \mathbb{P}\left(\max_{i\in\mathcal{J}} \frac{\hat{L}^{(j)}(\hat{h}_i^{(j)}) - L(\mathcal{M}_i)}{L(\hat{h}_i^{(j)})} \geq \delta(1-\delta)\right) \\
&\leq \mathbb{P}\left(\max_{i\in\mathcal{J}} \sup_{h\in\mathcal{M}_i} \left|\frac{L_{D_N}^{(j)}(h) - L(h)}{L(h)}\right| \geq \frac{\delta}{2}\right) + \mathbb{P}\left(\max_{i\in\mathcal{J}} \frac{\hat{L}^{(j)}(\hat{h}_i^{(j)}) - L(\mathcal{M}_i)}{L(\hat{h}_i^{(j)})} \geq \delta(1-\delta)\right)
\end{aligned} \tag{2.37}$$

in which the last inequality follows from Lemma A.18.

It remains to bound the second probability in (2.37). We have that it is equal to

$$\begin{aligned}
&\mathbb{P}\left(\max_{i\in\mathcal{J}} \frac{\hat{L}^{(j)}(\hat{h}_i^{(j)}) - L(\hat{h}_i^{(j)}) + L(\hat{h}_i^{(j)}) - L(\mathcal{M}_i)}{L(\hat{h}_i^{(j)})} \geq \delta(1-\delta)\right) \\
&\leq \mathbb{P}\left(\max_{i\in\mathcal{J}} \frac{\hat{L}^{(j)}(\hat{h}_i^{(j)}) - L(\hat{h}_i^{(j)})}{L(\hat{h}_i^{(j)})} \geq \frac{\delta(1-\delta)}{2}\right) + \mathbb{P}\left(\max_{i\in\mathcal{J}} \frac{L(\hat{h}_i^{(j)}) - L(\mathcal{M}_i)}{L(\hat{h}_i^{(j)})} \geq \frac{\delta(1-\delta)}{2}\right) \\
&\leq \mathbb{P}\left(\max_{i\in\mathcal{J}} \sup_{h\in\mathcal{M}_i} \left|\frac{\hat{L}^{(j)}(h) - L(h)}{L(h)}\right| \geq \frac{\delta(1-\delta)}{2}\right) + \mathbb{P}\left(\max_{i\in\mathcal{J}} \sup_{h\in\mathcal{M}_i} \left|\frac{L(h) - L_{\mathcal{D}_N}(h)}{L(h)}\right| \geq \frac{\delta(1-\delta)}{4}\right),
\end{aligned} \tag{2.38}$$

in which the last inequality follows again from Lemma A.18. From (2.33-2.38), follows that

$$\begin{aligned}
\mathbb{P}\left(L(\hat{\mathcal{M}}) \neq L(\mathcal{M}^\star)\right) \leq & 2\sum_{j=1}^{m}\left[\mathbb{P}\left(\max_{i\in\mathcal{J}} \sup_{h\in\mathcal{M}_i} \left|\frac{\hat{L}^{(j)}(h) - L(h)}{L(h)}\right| \geq \frac{\delta(1-\delta)}{2}\right) + \right. \\
& \left. \mathbb{P}\left(\max_{i\in\mathcal{J}} \sup_{h\in\mathcal{M}_i} \left|\frac{L(h) - L_{\mathcal{D}_N}(h)}{L(h)}\right| \geq \frac{\delta(1-\delta)}{4}\right)\right] \\
\leq & 2\sum_{j=1}^{m} \sum_{\mathcal{M}\in \operatorname{Max}\mathbb{L}(\mathcal{H})}\left[\mathbb{P}\left(\sup_{h\in\mathcal{M}} \left|\frac{\hat{L}^{(j)}(h) - L(h)}{L(h)}\right| \geq \frac{\delta(1-\delta)}{2}\right) + \right. \\
& \left. \mathbb{P}\left(\sup_{h\in\mathcal{M}} \left|\frac{L(h) - L_{\mathcal{D}_N}(h)}{L(h)}\right| \geq \frac{\delta(1-\delta)}{4}\right)\right],
\end{aligned}$$

in which the inequality holds by the same arguments as in (2.12). From this follows

$$\mathbb{P}\left(L(\hat{\mathcal{M}}) \neq L(\mathcal{M}^\star)\right) \leq 2m\,|\operatorname{Max}\mathbb{L}(\mathcal{H})|\left[\hat{B}_{N,\frac{\delta(1-\delta)}{2}}(d_{VC}(\mathbb{L}(\mathcal{H}))) + B_{N,\frac{\delta(1-\delta)}{4}}(d_{VC}(\mathbb{L}(\mathcal{H})))\right],$$

as desired.

If the almost sure convergences (2.31) hold, then $L(h) = L^{(j)}_{\mathcal{D}_N}(h) = \hat{L}^{(j)}(h)$ for all j and $h \in \mathcal{H}$, hence the definitions of $\mathcal{M}^\star$ and $\hat{\mathcal{M}}$ coincide. □

As was also the case for bounded loss functions, it follows from bound (2.30) that we have to better estimate with the training samples, that require a precision of $(\delta(1-\delta))/4$, in contrast to a precision $(\delta(1-\delta))/2$ with the validation samples. We note that the discussion at the end of Section 2.2 also applies to the case of unbounded loss functions.

In this instance, a bound for $\mathbb{P}(L(\hat{\mathcal{M}}) \neq L(\mathcal{M}^\star))$, and the almost sure convergence of $\hat{\mathcal{M}}$ to $\mathcal{M}^\star$, in the case of k-fold cross validation and independent validation sample, follow from Proposition 2.21 in a manner analogous to Theorem 2.9. We state the almost sure convergence in Theorem 2.23, whose proof is analogous to that of Theorem 2.9, and follows from Corollary A.14.

Theorem 2.23. *Assume the loss function is unbounded and P is such that (2.29) hold. If $\hat{L}$ is given by k-fold cross-validation or by an independent validation sample, then $\hat{\mathcal{M}}$ converges*

with probability one to $\mathcal{M}^\star$.

2.4.2 Convergence of estimation errors on $\hat{\mathcal{M}}$

Analogous to the case of bounded loss functions, the relative type III estimation error on $\hat{\mathcal{M}}$ depends solely on $\hat{\mathcal{M}}$, while relative types I, II, and IV estimation errors depend on $\hat{\mathcal{M}}$, but also on the choice of algorithm $\mathbb{A}$ employed to learn a hypothesis $\hat{h}^{\mathbb{A}}_{\hat{\mathcal{M}}} \in \hat{\mathcal{M}}$. In this section, we consider the two algorithms in Figure 1.19, that are learning by reusing and learning with independent sample. We start with the case of an independent sample, and then briefly discuss learning by reusing.

The results stated here are rather similar to the case of bounded loss functions, with some minor modifications, and virtually all the discussion of Section 2.3 applies to this case. Hence, we state the analogous results, present a proof only when it is different from the respective result in Section 2.3, and do not discuss further the results, referring to Section 2.3 for a comprehensive discussion.

2.4.3 Learning with independent sample

Bounds for relative types I and II estimation errors, when learning on a random model with a sample independent of the one employed to compute such random model, may be obtained as in Theorem 2.10. In fact, the proof of the following bounds are the same as in that theorem, with the respective changes from estimation errors to relative estimation errors. Hence, we state the results without a proof.

Theorem 2.24. *Fix an unbounded loss function and assume P is such that (2.29) hold. Assume we are learning with an independent sample $\tilde{\mathcal{D}}_M$, and that for each $\epsilon > 0$ there exist sequences $\{B^I_{M,\epsilon} : M \geq 1\}$ and $\{B^{II}_{M,\epsilon} : M \geq 1\}$ of positive real-valued increasing functions with domain $\mathbb{Z}_+$ satisfying*

$$\lim_{M\to\infty} B^I_{M,\epsilon}(k) = \lim_{M\to\infty} B^{II}_{M,\epsilon}(k) = 0,$$

for all $\epsilon > 0$ and $k \in \mathbb{Z}_+$ fixed, such that

$$\mathbb{P}\left(\sup_{h\in\mathcal{M}}\left|\frac{L_{\tilde{\mathcal{D}}_M}(h) - L(h)}{L(h)}\right| > \epsilon\right) \leq B^I_{M,\epsilon}(d_{VC}(\mathcal{M})) \text{ and}$$

$$\mathbb{P}\left(\frac{L(\hat{h}^{\tilde{\mathcal{D}}_M}_{\mathcal{M}}) - L(h^\star_{\mathcal{M}})}{L(\hat{h}^{\tilde{\mathcal{D}}_M}_{\mathcal{M}})} > \epsilon\right) \leq B^{II}_{M,\epsilon}(d_{VC}(\mathcal{M})),$$

for all $\mathcal{M} \in \mathbb{L}(\mathcal{H})$. Let $\hat{\mathcal{M}} \in \mathbb{L}(\mathcal{H})$ be a random model learned by $\mathbb{M}_{\mathbb{L}(\mathcal{H})}$. Then, for any $\epsilon > 0$,

$$\textbf{(I)}\ \mathbb{P}\left(\sup_{h\in\hat{\mathcal{M}}}\left|\frac{L_{\tilde{\mathcal{D}}_M}(h) - L(h)}{L(h)}\right| > \epsilon\right) \leq \mathbb{E}\left[B^I_{M,\epsilon}(d_{VC}(\hat{\mathcal{M}}))\right] \leq B^I_{M,\epsilon}\left(d_{VC}(\mathbb{L}(\mathcal{H}))\right)$$

and

$$\textbf{(II)}\ \mathbb{P}\left(\frac{L(\hat{h}^{\tilde{\mathcal{D}}_M}_{\hat{\mathcal{M}}}) - L(h^\star_{\hat{\mathcal{M}}})}{L(\hat{h}^{\tilde{\mathcal{D}}_M}_{\hat{\mathcal{M}}})} > \epsilon\right) \leq \mathbb{E}\left[B^{II}_{M,\epsilon}(d_{VC}(\hat{\mathcal{M}}))\right] \leq B^{II}_{M,\epsilon}\left(d_{VC}(\mathbb{L}(\mathcal{H}))\right),$$

in which the expectations are over all samples $\mathcal{D}_N$, from which $\hat{\mathcal{M}}$ is calculated. Since $d_{VC}(\mathbb{L}(\mathcal{H})) < \infty$, both probabilities above converge to zero when $M \longrightarrow \infty$.

The convergence to zero of relative type III estimation error may be obtained, as in Theorem 2.11, by the methods used to prove Theorem 2.22. We state and prove this result, since its proof is slightly different from that of Theorem 2.11. We note that the result below also holds when learning by reusing, since does not depend on the algorithm $\mathbb{A}$.

Theorem 2.25. *Assume the premises of Theorem 2.22 are in force. Let $\hat{\mathcal{M}} \in \mathbb{L}(\mathcal{H})$ be a random model learned by $\mathbb{M}_{\mathbb{L}(\mathcal{H})}$. Then, for any $\epsilon > 0$,*

$$\begin{aligned}\textbf{(III)}\ &\mathbb{P}\left(\frac{L(h^\star_{\hat{\mathcal{M}}}) - L(h^\star)}{L(h^\star_{\hat{\mathcal{M}}})} > \frac{\epsilon}{L(\mathcal{M}^\star)}\right) \\ &\leq 2m\ \mathfrak{m}(\mathbb{L}(\mathcal{H}))\left[\hat{B}_{N,\frac{\delta'(1-\delta')}{2}}(d_{VC}(\mathbb{L}(\mathcal{H}))) + B_{N,\frac{\delta'(1-\delta')}{4}}(d_{VC}(\mathbb{L}(\mathcal{H})))\right]\end{aligned}$$

in which

$$\delta' := \frac{\epsilon \vee \epsilon^\star}{2\max\limits_{i\in\mathcal{J}} L(\mathcal{M}_i)}.$$

In particular,

$$\lim_{N\to\infty} \mathbb{P}\left(\frac{L(h^\star_{\hat{\mathcal{M}}}) - L(h^\star)}{L(h^\star_{\hat{\mathcal{M}}})} > \epsilon\right) = 0,$$

for any $\epsilon > 0$.

Proof. We show that

$$\mathbb{P}\left(L(h^\star_{\hat{\mathcal{M}}}) - L(h^\star) > \epsilon\right) \geq \mathbb{P}\left(\frac{L(h^\star_{\hat{\mathcal{M}}}) - L(h^\star)}{L(h^\star_{\hat{\mathcal{M}}})} > \frac{\epsilon}{L(\mathcal{M}^\star)}\right), \tag{2.39}$$

so from (2.19) and (2.32) will follow that

$$\mathbb{P}\left(\frac{L(h^\star_{\hat{\mathcal{M}}}) - L(h^\star)}{L(h^\star_{\hat{\mathcal{M}}})} > \frac{\epsilon}{L(\mathcal{M}^\star)}\right) \leq \mathbb{P}\left(\min_{i\in\mathcal{J}} \frac{\hat{L}(\mathcal{M}_i)}{L(\mathcal{M}_i)} \leq 1-\delta'\right) + \mathbb{P}\left(\max_{i\in\mathcal{J}} \frac{\hat{L}(\mathcal{M}_i)}{L(\mathcal{M}_i)} \geq 1+\delta'\right),$$

and the result is then direct from the proof of Theorem 2.22. But (2.39) is clearly true since

$$\frac{L(h^\star_{\hat{\mathcal{M}}}) - L(h^\star)}{L(h^\star_{\hat{\mathcal{M}}})} > \frac{\epsilon}{L(\mathcal{M}^\star)} \implies L(h^\star_{\hat{\mathcal{M}}}) - L(h^\star) > \epsilon\frac{L(h^\star_{\hat{\mathcal{M}}})}{L(\mathcal{M}^\star)} \geq \epsilon. \tag{2.40}$$

□

Finally, a bound on the rate of convergence of type IV estimation error to zero is a direct consequence of Theorems 2.24 and 2.25, and the following inequality

$$\begin{aligned}\textbf{(IV)}\ &\mathbb{P}\left(\frac{L(\hat{h}^{\tilde{D}_M}_{\hat{\mathcal{M}}}) - L(h^\star)}{L(\hat{h}^{\tilde{D}_M}_{\hat{\mathcal{M}}})} > \frac{\epsilon}{L(\mathcal{M}^\star)}\right)\\ &\leq \mathbb{P}\left(\frac{L(\hat{h}^{\tilde{D}_M}_{\hat{\mathcal{M}}}) - L(h^\star_{\hat{\mathcal{M}}})}{L(\hat{h}^{\tilde{D}_M}_{\hat{\mathcal{M}}})} > \frac{\epsilon}{2L(\mathcal{M}^\star)}\right) + \mathbb{P}\left(\frac{L(h^\star_{\hat{\mathcal{M}}}) - L(h^\star)}{L(h^\star_{\hat{\mathcal{M}}})} > \frac{\epsilon}{2L(\mathcal{M}^\star)}\right),\end{aligned}$$

which is true since $L(\hat{h}^{\tilde{D}_M}_{\hat{\mathcal{M}}}) \geq L(h^\star_{\hat{\mathcal{M}}})$.

Corollary 2.26. *Assume the premises of Theorem 2.22 and 2.24 are in force. Let $\hat{\mathcal{M}} \in \mathbb{L}(\mathcal{H})$ be a random model learned by $\mathbb{M}_{\mathbb{L}(\mathcal{H})}$. Then, for any $\epsilon > 0$,*

$$\begin{aligned}&\mathbb{P}\left(\frac{L(\hat{h}^{\tilde{D}_M}_{\hat{\mathcal{M}}}) - L(h^\star)}{L(\hat{h}^{\tilde{D}_M}_{\hat{\mathcal{M}}})} > \frac{\epsilon}{L(\mathcal{M}^\star)}\right)\\ &\leq \mathbb{E}\left[B^{II}_{M,\frac{\epsilon}{2L(\mathcal{M}^\star)}}(d_{VC}(\hat{\mathcal{M}}))\right] + 2m\,\mathfrak{m}(\mathbb{L}(\mathcal{H}))\left[\hat{B}_{N,\frac{\delta'(1-\delta')}{2}}(d_{VC}(\mathbb{L}(\mathcal{H}))) + B_{N,\frac{\delta'(1-\delta')}{4}}(d_{VC}(\mathbb{L}(\mathcal{H})))\right]\\ &\leq B^{II}_{M,\frac{\epsilon}{2L(\mathcal{M}^\star)}}(d_{VC}(\mathbb{L}(\mathcal{H}))) + 2m\,\mathfrak{m}(\mathbb{L}(\mathcal{H}))\left[\hat{B}_{N,\frac{\delta'(1-\delta')}{2}}(d_{VC}(\mathbb{L}(\mathcal{H}))) + B_{N,\frac{\delta'(1-\delta')}{4}}(d_{VC}(\mathbb{L}(\mathcal{H})))\right]\end{aligned}$$

with

$$\delta' := \frac{\epsilon/2 \vee \epsilon^\star}{2\max_{i\in\mathcal{J}} L(\mathcal{M}_i)}.$$

In particular,

$$\lim_{\substack{N\to\infty\\ M\to\infty}} \mathbb{P}\left(\frac{L(\hat{h}^{\tilde{D}_M}_{\hat{\mathcal{M}}}) - L(h^\star)}{L(\hat{h}^{\tilde{D}_M}_{\hat{\mathcal{M}}})} > \epsilon\right) = 0,$$

for any $\epsilon > 0$.

From Proposition 2.21, Theorems 2.22, 2.23, 2.24 and 2.25, and Corollary 2.26, follow the consistency of the Model Selection framework given by selecting $\hat{\mathcal{M}}$ via $\mathbb{M}_{\mathbb{L}(\mathcal{H})}$ and learning on it with an independent sample, when we consider $\hat{L}$ given by k-fold cross-validation or an independent validation sample, the loss function is unbounded, and P satisfies (2.29).

Corollary 2.27. *Assume the loss function is unbounded and P is such that (2.29) hold. The Model Selection framework given by*

- *(a) estimating $L(\mathcal{M})$ by k-fold cross validation with a fixed k or by an independent validation sample,*
- *(b) selecting $\hat{\mathcal{M}}$ via $\mathbb{M}_{\mathbb{L}(\mathcal{H})}$,*

(c) and learning with an independent sample on $\hat{\mathcal{M}}$,

is consistent.

2.4.4 Learning by reusing

When learning by reusing, a result analogous to Theorem 2.17, together with Theorem 2.22 and a result analogous to Corollary 2.26, will imply the consistency of the approach. We state this result and the consistency without proof.

Theorem 2.28. *Fix an unbounded loss function and assume* P *is such that* (2.29) *hold. Assume we are learning by reusing and that, for each* $\epsilon > 0$, *there exist sequences* $\{B^I_{N,\epsilon} : N \geq 1\}$ *and* $\{B^{II}_{N,\epsilon} : N \geq 1\}$ *of positive real-valued increasing functions with domain* $\mathbb{Z}_+$ *satisfying*

$$\lim_{N\to\infty} B^I_{N,\epsilon}(k) = \lim_{N\to\infty} B^{II}_{N,\epsilon}(k) = 0,$$

for all $\epsilon > 0$ *and* $k \in \mathbb{Z}_+$ *fixed, such that*

$$\mathbb{P}\left(\sup_{h\in\mathcal{M}}\left|\frac{L_{D_N}(h) - L(h)}{L(h)}\right| > \epsilon\right) \leq B^I_{N,\epsilon}(d_{VC}(\mathcal{M})) \text{ and}$$

$$\mathbb{P}\left(\frac{L(\hat{h}^{D_N}_{\mathcal{M}}) - L(h^\star_{\mathcal{M}})}{L(\hat{h}^{D_N}_{\mathcal{M}})} > \epsilon\right) \leq B^{II}_{N,\epsilon}(d_{VC}(\mathcal{M})),$$

for all $\mathcal{M} \in \mathbb{L}(\mathcal{H})$. *Let* $\hat{\mathcal{M}} \in \mathbb{L}(\mathcal{H})$ *be a random model learned by* $\mathbb{M}_{\mathbb{L}(\mathcal{H})}$. *Then, for any* $\epsilon > 0$,

$$\textbf{\textit{(I)}}\ \mathbb{P}\left(\sup_{h\in\hat{\mathcal{M}}}\left|\frac{L_{D_N}(h) - L(h)}{L(h)}\right| > \epsilon\right) \leq B^I_{N,\epsilon}(d_{VC}(\mathcal{M}^\star)) + \mathbb{P}\left(\hat{\mathcal{M}} \neq \mathcal{M}^\star\right)$$

and

$$\textbf{\textit{(II)}}\ \mathbb{P}\left(\frac{L(\hat{h}^{D_N}_{\hat{\mathcal{M}}}) - L(h^\star_{\hat{\mathcal{M}}})}{L(\hat{h}^{D_N}_{\hat{\mathcal{M}}})} > \epsilon\right) \leq B^{II}_{N,\epsilon}(d_{VC}(\mathcal{M}^\star)) + \mathbb{P}\left(\hat{\mathcal{M}} \neq \mathcal{M}^\star\right).$$

If conditions (2.31) *of Theorem 2.22 are satisfied, both probabilities above converge to zero when* $N \longrightarrow \infty$.

Corollary 2.29. *Assume the loss function is unbounded and* P *is such that* (2.29) *hold. The Model Selection framework given by*

(a) estimating $L(\mathcal{M})$ *by k-fold cross validation with a fixed* k *or by an independent validation sample,*

(b) selecting $\hat{\mathcal{M}}$ *via* $\mathbb{M}_{\mathbb{L}(\mathcal{H})}$,

(c) and learning by reusing on $\hat{\mathcal{M}}$,

is consistent.

2.5 Next steps

So far, we have established a Model Selection framework which is data-driven, systematic, and consistent, and that may perform better than learning via ERM directly on $\mathcal{H}$. In the next chapter, we turn to the last facet of the method, which regards its computational complexity, more specifically the existence of non-exhaustive algorithms to compute $\hat{\mathcal{M}}$ by minimizing $\hat{L}$ in $\mathbb{L}(\mathcal{H})$.

We have seen at the end of Section 2.3.1 that, even though the method may be better than learning via ERM directly on $\mathcal{H}$, there is an intrinsic computational aspect that may forbid its application when it demands an exhaustive search of a $\mathbb{L}(\mathcal{H})$ with a cardinality exponential on $d_{VC}(\mathcal{H})$. Observe that this is the case in the examples in Section 1.3.2, and will usually be the case in practical problems.

Fortunately, as we will see in the next chapter, and in the applications in Chapter 4, there are some instances in which a non-exhaustive algorithm may be employed to find the optimal solution $\hat{\mathcal{M}}$, and, even when this is not the case, the method may work well in practice, returning suboptimal solutions that, even though are not $\hat{\mathcal{M}}$, may be suitable for the application at hand. This will establish the non-exhaustiveness of the method.

Chapter 3

U-curve: properties and algorithms

As important as the consistency of the Model Selection framework via Learning Spaces, is the possibility of computing $\hat{\mathcal{M}}$ for solving real problems. As can be noted in the examples of Section 1.3, the cardinality of a Learning Space might be (more than) exponential on the number of parameters representing the hypotheses in $\mathcal{H}$, hence an exhaustive search of it is usually not practical.

In this chapter, we discuss properties that, when satisfied by a Learning Space, allow the development of non-exhaustive algorithms to compute $\hat{\mathcal{M}}$. We define the so-called U-curve properties in Section 3.2, and show in Section 3.3 that one of them is satisfied by the Partition Lattice Learning Space. In Section 3.4, we establish a sufficient condition for a U-curve property that shed light on what it actually means, by drawing a parallel to convexity. Then, in Section 3.5 we present a generic non-exhaustive algorithm to compute $\hat{\mathcal{M}}$ when a U-curve property is satisfied.

The U-curve phenomenon, formally defined here as the U-curve properties, and proved to hold in a Learning Space, is related to many heuristics and features empirically observed in learning problems, which we briefly discuss in the next section.

3.1 Occam's razor and peaking phenomenon are facets of U-curve

The U-curve phenomenon has been empirically observed as a decrease on the estimated error of a sequence of nested models with increasing complexity, up to a point when there is an inflection point, and the error starts to monotonically increase with further increment of the complexity. This idea is illustrated in Figure 3.1, where the estimated error $\hat{L}$ in a chain $\mathcal{M}_0 \subset \cdots \subset \mathcal{M}_6$ has a U-format.

A first facet of the U-curve phenomenon is the principle of parsimony, which is attributed to William of Ockham (1287-1347), and is often called Occam's razor, which states that "entities should not be multiplied beyond necessity" [45, 146]. Informally, this is

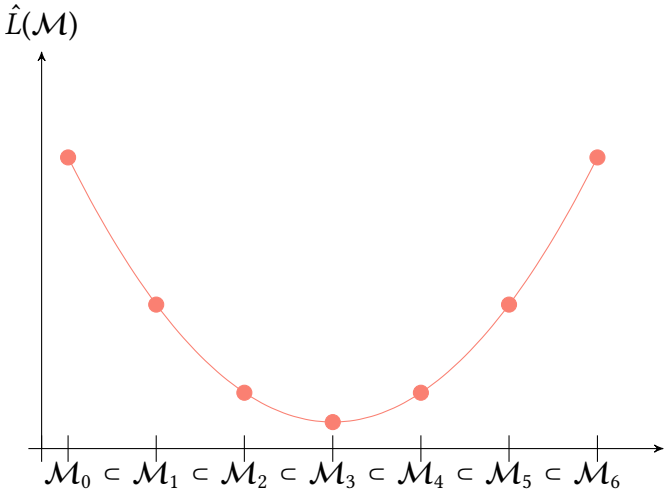

Figure 3.1: *Illustration of the U-curve phenomenon, instantiated to a chain of nested models with increasing complexity. This is the typical behavior of $\hat{L}$ on continuous chains of a Learning Space that satisfies the strong U-curve property (cf. Definition 3.1).*

interpreted as to prefer the simplest available explanation, and may not only be related to the U-curve phenomenon, but also to the method proposed in this thesis in general.

On the one hand, when an inflection is observed in the setting of Figure 3.1, we may conclude that a suitable explanation has been found, and further complicating the explanation (model) is actually harmful, as the principle dictates. On the other hand, the paradigm, presented in Figure 1.16, of seeking to learn on the simplest unbiased model $\mathcal{M}^\star$ of $\mathbb{L}(\mathcal{H})$, and the choice of $\hat{\mathcal{M}}$ as the simplest strong local minimum, are also in direct alignment with this principle.

Another facet of the U-curve phenomenon is the *peaking phenomenon*, also called *curse of dimensionality*, which is a special case of the phenomenon illustrated in Figure 3.1, in which the complexity of the models is related to the number of variables/parameters representing them (dimension), and the next model in the sequence is obtained by adding more variables/parameters to the prior model. There are some specific cases in which this principle was illustrated to hold, and we refer to [73, 74, 101, 126, 127, 162] and the references therein for a further discussion of the peaking phenomenon.

Based on these two facets of the U-curve phenomenon, specially on the peaking phenomenon, the alleged existence of an inflection point has been employed as a stopping criterion for Model Selection algorithms, and inspired the development of the U-curve algorithms [8, 55, 128, 130, 131, 133] in the context of feature selection. Actually, the U-curve phenomenon is implicitly considered in classical algorithms for feature selection such as *Sequential Backward Selection* (SBS) [97, 141], *Sequential Forward Selection* (SFS) [155], *Sequential Forward Floating Selection* (SFFS) [121], *Adaptative Floating Search* (ASFFS) [143], *Beam-Search* [140] and *branch-and-bound* [106]. See [128, Chapter 2] for a review of these algorithms and their relation to the U-curve algorithms.

However, in many instances there is a lack of theoretical results which guarantee that a phenomenon such as that of Figure 3.1 really holds, and that the inflection point may

somehow be used as a stopping criterion. In fact, it could happen that the error curve is not monotonically increasing after the inflection point, so there may exist another local minimum in the sequence, that might have a lower estimated error, hence stopping at the first local minimum leads to a suboptimal solution.

In this context, we propose a formalization of the U-curve phenomenon on Learning Spaces, establishing a sufficient condition for it to hold, and showing that it holds, in some form, for the Partition Lattice Learning Space. Moreover, we propose a generic algorithm that properly employs the phenomenon as a stopping criterion for an optimal non-exhaustive computation of $\hat{\mathcal{M}}$.

It will be clear from our proofs that the behavior in Figure 3.1 does not hold in general for a continuous chain of a Learning Space, but a weaker version of it, which considers all chains that passes through a local minimum, holds in the Partition Lattice Learning Space (cf. Proposition 3.2) and in some subsets of it (cf. Corollary 3.3), and is much more plausible to hold in other cases.

3.2 U-curve properties

The Model Selection approach based on Learning Spaces is a solution of optimization problem

$$\hat{\mathcal{M}} := \mathbb{M}_{\mathbb{L}(\mathcal{H})}(\mathcal{D}_N, \hat{L}) \in \hat{\mathcal{L}} = \underset{\mathcal{M}\in\mathbb{L}(\mathcal{M})}{\arg\min}\, \hat{L}(\mathcal{M}), \tag{3.1}$$

that is the solution in $\hat{\mathcal{L}}$ with the least VC dimension.

The main issue with problem (3.1) is that, in principle, it demands a combinatorial algorithm which exhaustively searches $\mathbb{L}(\mathcal{H})$ to compute $\hat{\mathcal{L}}$ and then select $\hat{\mathcal{M}}$ as the simplest model in it. However, due to properties of $\mathbb{L}(\mathcal{H})$ under a loss function ℓ, and the fact that we consider an estimator $\hat{L}(\mathcal{M})$ apart from the resubstitution error $L_{\mathcal{D}_N}(\hat{h}_{\mathcal{M}}^{\mathcal{D}_N})$, one may take advantage of a U-curve property to solve the problem without exhaustively searching $\mathbb{L}(\mathcal{H})$. Indeed, to find $\hat{\mathcal{M}}$ one needs only to find all strong, sup-strong, inf-strong or weak local minimums of $\mathbb{L}(\mathcal{H})$ (cf. Definition 1.17), as each global minimum is one of them, a search which may be performed more efficiently than an exhaustive one if the loss function satisfies a U-curve property.

Definition 3.1. *A Learning Space $\mathbb{L}(\mathcal{H})$ under loss function ℓ and estimator $\hat{L}$ satisfies the:*

- ***strong U-curve property*** *if every weak local minimum of a continuous chain of $\mathbb{L}(\mathcal{H})$ is a global minimum of such chain;*
- ***weak U-curve property*** *if every strong local minimum is a global minimum of all continuous chains of $\mathbb{L}(\mathcal{H})$ which contain it;*
- ***sup-weak U-curve property*** *if every sup-strong local minimum has an estimated error lesser or equal to that of all models in $\mathbb{L}(\mathcal{H})$ which contain it;*
- ***inf-weak U-curve property*** *if every inf-strong local minimum has an estimated error lesser or equal to that of all models in $\mathbb{L}(\mathcal{H})$ contained in it.*

The conditions characterizing the U-curve properties should be true with probability one, holding for all possible samples $\mathcal{D}_N$, for any value of N.

We call the properties *U-curve*, since the plot of $(i_j, \hat{L}(\mathcal{M}_{i_j})), j = 1, \ldots, k$, is U-shaped when calculated for any continuous chain $\mathcal{M}_{i_1} \subset \cdots \subset \mathcal{M}_{i_k}$, if the strong U-curve property holds (see Figure 3.1). It is straightforward that the strong implies the weak U-curve property. Since the concept of local minimums (cf. Definition 1.17) depends on $\hat{L}$, so does the U-curve properties, whose conditions, once $\hat{L}$ is fixed, should hold for any possible sample $\mathcal{D}_N$. In Figure 3.2, we present an example of a lattice which satisfies the weak U-curve property.

In Figure 3.3 is illustrated a strong and a sup-strong local minimum. On the one hand, we see in (a) that the intersection of all chains is a strong local minimum, since it is a local minimum of all continuous chains which contain it. Furthermore, it is also the global minimum of all continuous chains which contain it, presenting the behavior which characterizes the weak U-curve property[1]. On the other hand, in (b) we see that the intersection of all chains is a sup-strong local minimum and has an error lesser than the models greater than it, a behavior that characterizes the sup-weak U-curve property. The red model in (b) is also a weak local minimum of four chains that pass through it, and is the global minimum of such chains, a behavior that characterizes the strong U-curve property.

All U-curve properties allow a non-exhaustive search for $\hat{\mathcal{M}}$. On the one hand, if the strong U-curve property is satisfied, to find $\hat{\mathcal{M}}$ we do not need to exhaustively search $\mathbb{L}(\mathcal{H})$: we go through every continuous chain of $\mathbb{L}(\mathcal{H})$ until we find a weak local minimum of it so that we find every weak local minimum, and, therefore, the global minimum. Similarly, if the weak U-curve property is satisfied, to find $\hat{\mathcal{M}}$ we go through every continuous chain of $\mathbb{L}(\mathcal{H})$ until we find a strong local minimum of it so that we find every strong local minimum, and, therefore, the global minimum. Either way, $\mathbb{L}(\mathcal{H})$ is not exhaustively searched, as when we find a weak or strong local minimum we do not need to estimate the error of the remaining models (greater or lesser than the local minimum) of a continuous chain, as the strong or weak U-curve property, respectively, ensure that the found local minimum is a global minimum of the continuous chain. Something analogous to this was first done for variable selection lattices in [8, 55, 128, 130, 131, 133].

On the other hand, if the sup-weak (inf-weak) U-curve property is satisfied, to find $\hat{\mathcal{M}}$ we can go through every continuous chain of $\mathbb{L}(\mathcal{H})$ until we find a sup-strong (inf-strong) local minimum of it so that we find every sup-strong (inf-strong) local minimum and, therefore, the global minimum. In this case, $\mathbb{L}(\mathcal{H})$ is not exhaustively searched, as when we find a sup-strong (inf-strong) local minimum we do not need to estimate the error of the greater (lesser) models of a continuous chain, as the sup-weak (inf-weak) U-curve property ensures that the found local minimum has an error lesser or equal to the models greater (lesser) than it.

The U-curve properties are characterized by local features of the estimated error $\hat{L}$,

[1] Although in Figure 3.3 (a) the error is monotone before and after the strong local minimum, this is not always the case when the weak U-curve property is satisfied. Observe that in Definition 3.1 there is nothing forbidding the existence, in chains that pass through strong local minimums, of a weak local minimum with an error greater or equal to that of the strong.

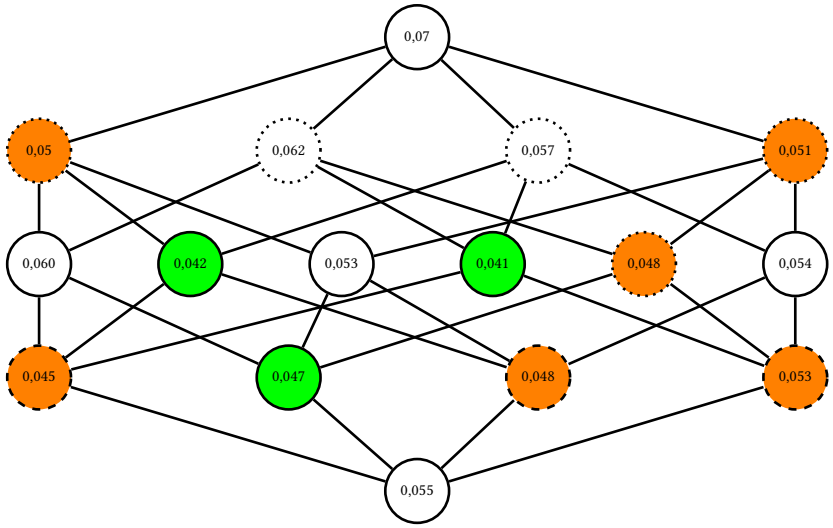

Figure 3.2: *Example of a lattice satisfying the weak U-curve property. The number inside each node $\mathcal{M}$ is $\hat{L}(\mathcal{M})$. The strong local minimums are in green, the weak local minimums are in orange, the inf-strong local minimums are dashed and the sup-weak local minimums are dotted. All strong local minimums are global minimums of all continuous chains which contain them, so this is an example of a weak U-curve property configuration. The inclusion relation $\subset$ is from the bottom to the top.*

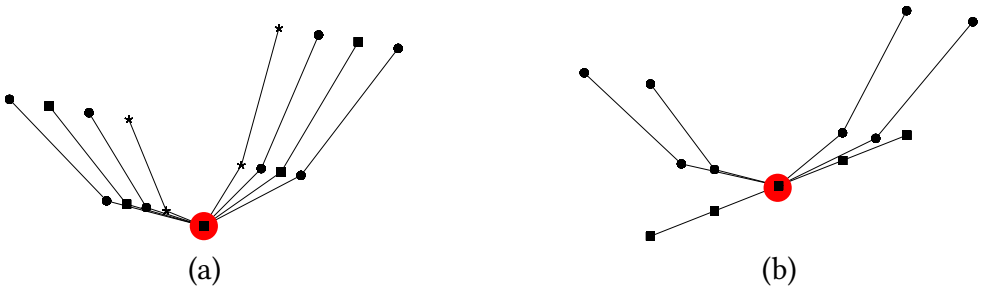

Figure 3.3: *Illustration of (a) strong and (b) sup-strong local minimums.*

when calculated for chains of a Learning Space, which imply global properties of such chains. This implication is only possible due to the existence of a structure, given by relations between the models in $\mathbb{L}(\mathcal{H})$. Therefore, not only the structure of the Learning Space is highly related to the rate of convergence to the target model, evidenced by the MDE $\epsilon^\star$, but it is also what enables the estimation of $\hat{\mathcal{M}}$ via the solution of a U-curve optimization problem.

The next step, after defining the U-curve properties, is to point Learning Spaces which satisfy them. A natural way is to establish sufficient conditions for a U-curve property, which can be verified on a given $\mathbb{L}(\mathcal{H})$ or employed to build Learning Spaces. In Section 3.3, we show that the Partition Lattice Learning Space satisfies the sup-weak U-curve property, and in Section 3.4 we establish a sufficient condition for the weak U-curve property.

3.3 U-curve on the Partition Lattice Learning Space

When $\mathcal{X}$ is finite, one could, theoretically, search the Partition Lattice Learning Space to estimate a target partition. However, since the cardinality of this space is the $|\mathcal{X}|$-Bell number [18, 21, 22], which increases more than exponentially with $|\mathcal{X}|$ (see Table 3.1 for the 30 first Bell numbers), an exhaustive search of this lattice is impractical.

However, depending on how one defines the loss function ℓ and the error estimator $\hat{L}$ of each partition, a non-exhaustive search may be performed in this space, returning a suitable partition on which to learn a good hypothesis, since the sup-weak U-curve property is satisfied. This is the result of the next proposition.

$\lvert\mathcal{X}\rvert$	Bell number	$\lvert\mathcal{X}\rvert$	Bell number	$\lvert\mathcal{X}\rvert$	Bell number
1	1	11	678,570	21	474,869,816,156,751
2	2	12	4,213,597	22	4,506,715,738,447,323
3	5	13	27,644,437	23	44,152,005,855,084,344
4	15	14	190,899,322	24	445,958,869,294,805,312
5	52	15	1,382,958,545	25	4,638,590,332,229,998,592
6	203	16	10,480,142,147	26	49,631,246,523,618,762,752
7	877	17	82,864,869,804	27	545,717,047,936,060,030,976
8	4,140	18	682,076,806,159	28	6,160,539,404,599,936,679,936
9	21,147	19	5,832,742,205,057	29	71,339,801,938,860,290,605,056
10	115,975	20	51,724,158,235,372	30	846,749,014,511,809,254,653,952

Table 3.1: *First to 30th Bell number.*

Proposition 3.2. *The Partition Lattice Learning Space under the simple loss function and $\hat{L}$ of the form (2.8) satisfies the sup-weak U-curve property.*

Proof. We first consider the case $m = 1$, that is when the sample is split into a training and validation sample. It is enough to show that, if $\mathcal{H}|_\pi$ is a sup-strong local minimum of $\mathbb{L}(\mathcal{H})$ and $\pi \preceq \pi_i$, then $\hat{L}(\hat{h}_\pi) \leq \hat{L}(\hat{h}_i)$, in which $\hat{h}_\pi$ and $\hat{h}_i$ are the ERM hypothesis of $\mathcal{H}|_\pi$ and $\mathcal{H}|_{\pi_i}$ under the training sample, and $\hat{L}$ is the empirical error under the validation sample. To aid in the understanding of the statements in this proof, one may test them in the joint frequency tables in Figure 1.13 to better comprehend their logic.

Note that, for all $\pi_j \preceq \pi_i$, it holds

$$\hat{h}_j(x) = \hat{h}_i(x) \text{ for all } x \in \bigcup_{a \in \pi_j \cap \pi_i} a.$$

This is the case because, if $a \in \pi_j \cap \pi_i$, then, for all $x \in a$,

$$\hat{h}_j(x) = \arg\max_{y \in \{0,1\}} \sum_{k=1}^{N-V_N} \frac{\mathbb{1}\{Y_k = y, X_k \in a\}}{\sum_{k=1}^{N-V_N} \mathbb{1}\{X_k \in a\}}. \tag{3.2}$$

From (3.2) follows that the value of $\hat{h}_{\pi'}(x)$ for $x \in a$ is always the same if $a \in \pi'$, whatever the partition π' that contains a or, in other words, however the points in $\mathcal{X} \setminus a$ are partitioned. Furthermore, if $\pi_j \preceq \pi_i$ and $|\pi_j| = |\pi_i| - 1$, then

$$\pi_i \setminus \left(\pi_j \cap \pi_i\right) = \{a_1, a_2\} \qquad\qquad \pi_j = \left(\pi_j \cap \pi_i\right) \bigcup \{a_1 \cup a_2\},$$

as π_i is obtained from π_j by partitioning a block of it into two blocks a_1, a_2. From (3.2), we can establish that

$$\hat{h}_i(x) = \hat{h}_j(x), \qquad \text{for all } x \in \bigcup_{a \in \pi_j \cap \pi_i} a \cup a_1 \text{ or for all } x \in \bigcup_{a \in \pi_j \cap \pi_i} a \cup a_2.$$

Indeed, if

$$y^\star := \arg\max_{y \in \{0,1\}} \sum_{k=1}^{N-V_N} \frac{\mathbb{1}\{Y_k = y, X_k \in a_1 \cup a_2\}}{\sum_{k=1}^{N-V_N} \mathbb{1}\{X_k \in a_1 \cup a_2\}}, \tag{3.3}$$

then at least one of the following equalities hold

$$\begin{aligned} y_1^\star &:= \arg\max_{y \in \{0,1\}} \sum_{k=1}^{N-V_N} \frac{\mathbb{1}\{Y_k = y, X_k \in a_1\}}{\sum_{k=1}^{N-V_N} \mathbb{1}\{X_k \in a_1\}} = y^\star \text{ or} \\ y_2^\star &:= \arg\max_{y \in \{0,1\}} \sum_{k=1}^{N-V_N} \frac{\mathbb{1}\{Y_k = y, X_k \in a_2\}}{\sum_{k=1}^{N-V_N} \mathbb{1}\{X_k \in a_2\}} = y^\star, \end{aligned} \tag{3.4}$$

as the ratio in (3.3) with $y = y^\star$ is a weighted mean of the ratios in (3.4) with $y = y^\star$, so that if it is greater than 1/2, as is the case when $y = y^\star$, then the maximum of the ratios in (3.4) is greater than 1/2 when $y = y^\star$, as the maximum is not lesser than the weighted mean. This establishes that at least one of the $y_1^\star, y_2^\star$ is equal to $y^\star$.

Assume that $\mathcal{H}|_\pi$, is a sup-strong local minimum of $\mathbb{L}(\mathcal{H})$. Then, if $\pi \preceq \pi_i$ and $|\pi| = |\pi_i| - 1$, denoting $A = \bigcup_{a \in \pi \cap \pi_i} a$ and $\hat{P}$ as the empirical measure of the validation sample, we have that

$$\begin{aligned} \hat{L}(\hat{h}_\pi) \leq \hat{L}(\hat{h}_i) &= \int_{A \times \{0,1\}} \ell((x,y), \hat{h}_i)\, d\hat{P}(x,y) + \int_{A^c \times \{0,1\}} \ell((x,y), \hat{h}_i)\, d\hat{P}(x,y) \\ &= \int_{A \times \{0,1\}} \ell((x,y), \hat{h}_\pi)\, d\hat{P}(x,y) + \int_{A^c \times \{0,1\}} \ell((x,y), \hat{h}_i)\, d\hat{P}(x,y) \end{aligned} \tag{3.5}$$

as $\hat{h}_i(x) = \hat{h}_\pi(x)$ for $x \in A$. We have that $A^c = a_1 \cup a_2$, with $a_1 \cap a_2 = \emptyset$, $a_1, a_2 \in \pi_i$, and

$$\int_{A^c \times \{0,1\}} \ell((x,y), \hat{h}_i)\, d\hat{P}(x,y) = \int_{a_1 \times \{0,1\}} \ell((x,y), \hat{h}_i)\, d\hat{P}(x,y) + \int_{a_2 \times \{0,1\}} \ell((x,y), \hat{h}_i)\, d\hat{P}(x,y) \tag{3.6}$$

so that, as $\mathcal{H}|_\pi$ is a sup-strong local minimum, by substituting (3.6) in (3.5), we have

$$\int_{a_1 \times \{0,1\}} \ell((x,y), \hat{h}_i)\, d\hat{P}(x,y) \geq \int_{a_1 \times \{0,1\}} \ell((x,y), \hat{h}_\pi)\, d\hat{P}(x,y)$$

$$\int_{a_2\times\{0,1\}} \ell((x,y),\hat{h}_i)\; d\hat{P}(x,y) \geq \int_{a_2\times\{0,1\}} \ell((x,y),\hat{h}_\pi)\; d\hat{P}(x,y), \tag{3.7}$$

with equality holding for at least one of the two inequalities by (3.4).

Since $\mathcal{H}|_\pi$ is a sup-strong local minimum, condition (3.7) holds for any $a_1 \subset b \in \pi$ by taking $a_2 = b \setminus a_1$. Indeed, let $a_1 \subset b \in \pi$ be arbitrary and consider $\pi^\star = (\pi \setminus \{b\}) \cup \{a_1, b \setminus a_1\}$. Then clearly $\pi \leq \pi^\star$ and $|\pi| = |\pi^\star| - 1$, so (3.7) follows.

To end the proof, recalling that $A = \bigcup_{a\in\pi\cap\pi_i} a$, we note that, if $\pi \leq \pi_i$, then

$$\begin{aligned}\hat{L}(\hat{h}_i) &= \int_{A\times\{0,1\}} \ell((x,y),\hat{h}_\pi)\; d\hat{P}(x,y) + \sum_{j=1}^{p} \int_{a_j\times\{0,1\}} \ell((x,y),\hat{h}_i)\; d\hat{P}(x,y) \\ &\geq \int_{A\times\{0,1\}} \ell((x,y),\hat{h}_\pi)\; d\hat{P}(x,y) + \sum_{j=1}^{p} \int_{a_j\times\{0,1\}} \ell((x,y),\hat{h}_\pi)\; d\hat{P}(x,y) = \hat{L}(\hat{h}_\pi),\end{aligned} \tag{3.8}$$

in which $\{a_1, \ldots, a_p\} = \pi_i \setminus (\pi \cap \pi_i)$ is a partition of A^c. Inequality (3.8) holds since, for all $a_j \in \pi_i$, there exists a $b_j \in \pi$ such that $a_j \subset b_j$, which follows from the fact that $\pi \leq \pi_i$, so we may apply inequality (3.7) to each parcel of the sum in (3.8).

To show the result for $m > 1$, we may repeat the proof above to each term in sum (2.8), since they each represent a pair of independent training and validation samples, to establish that if $H|_\pi$ is a sup-strong local minimum, then

$$\hat{L}^{(j)}(\hat{h}_\pi^{(j)}) \leq \hat{L}^{(j)}(\hat{h}_i^{(j)}), \qquad \forall j = 1, \ldots, m \tag{3.9}$$

for any π_i such that $\pi \leq \pi_i$. Summing (3.9) for j from 1 to m and dividing by m we have the result. □

From the proof of Proposition 3.2, it actually follows that if, instead of considering the whole Partition Lattice Learning Space, one considered a subset of it that satisfies the following property, then the sup-weak U-curve property is still satisfied.

Let $\mathcal{F} = \{\pi \,:\, \pi \text{ is a partition of } \mathcal{X}\}$ and denote, for each $\mathcal{F}' \subset \mathcal{F}$,

$$\mathbb{L}_{\mathcal{F}'}(\mathcal{H}) := \left\{ \mathcal{H}|_\pi \,:\, \pi \in \mathcal{F}' \right\} \subset \mathbb{L}(\mathcal{H}),$$

recalling that $\mathcal{H}|_\pi$ are the hypotheses in $\mathcal{H}$ which respect partition π, that is, classify points within a same block of partition π in a same class. Denote, for every pair $\pi, \pi_i \in \mathcal{F}'$ satisfying $\pi \leq \pi_i$,

$$\pi_i \setminus (\pi \cap \pi_i) := \left\{ a_1, \ldots, a_p \right\},$$

for a $p \geq 2$. For each $j = 1, \ldots, p$, let $b_j \subset \mathcal{X}$ be such that $a_j \cup b_j \in \pi$, which exists by the definition of partial order $\leq$. We show that the following condition on $\mathcal{F}'$ is sufficient for the weak U-curve property:

$$\pi, \pi_i \in \mathcal{F}',\ \pi \leq \pi_i \implies \pi_j := \left(\pi \setminus \left\{ a_j \cup b_j \right\}\right) \cup \left\{ a_j, b_j \right\} \in \mathcal{F}',\ \forall j = 1, \ldots, p. \tag{3.10}$$

This is the content of the following corollary.

Corollary 3.3. *If the subset $\mathcal{F}'$ of the Partition Lattice of $\mathcal{X}$ satisfies (3.10), then $\mathbb{L}_{\mathcal{F}'}(\mathcal{H})$ under the simple loss function and $\hat{L}$ of the form (2.8) satisfies the sup-weak U-curve property.*

Proof. In this instance, the proof of Proposition 3.2 remains true until formula (3.7), with the obvious modification that all partitions considered are in $\mathcal{F}'$. We continue with the proof after this formula.

As $\mathcal{H}|_\pi$ is a sup-strong local minimum, condition (3.7) holds for any a_1, a_2 such that $a_1 \cup a_2 \in \pi$ and $(\pi \setminus \{a_1 \cup a_2\}) \cup \{a_1, a_2\} \in \mathcal{F}'$. Hence, inequality (3.8) remains true since, by condition (3.10), for all $j = 1, \dots, p$, there exists a b_j such that $a_j \cup b_j \in \pi$ and $\left(\pi \setminus \left\{a_j \cup b_j\right\}\right) \cup \{a_j, b_j\} \in \mathcal{F}'$, so we may apply inequality (3.7) to each parcel of the sum, and (3.8) indeed follows.

From this point on, the proof remains the same as in Proposition 3.2 and the result follows. □

Learning hypotheses via the Partition Lattice Learning Space, although demands a lot of computation, has some advantages. First, if one has prior information about the partition generated by $h^\star$, he may search only the partitions which satisfy a given property, or within a partition consider only hypotheses that respect it and satisfy a given condition. Second, once a partition is selected, one may qualitatively analysis it, and the path of the U-curve algorithm until it (cf. Algorithm 4), to obtain insights about the learned hypothesis and better understand why it classifies certain inputs in an output.

In Section 4.1, we present some simulated examples of learning on the Partition Lattice Learning Space, which outline some interesting features of it, and in Section 4.4 we present a subset of the Partition Lattice Learning Space, suitable for solving image transformation tasks, which satisfies the sup-weak U-curve property due to Corollary 3.3.

3.4 Sufficient condition for the weak U-curve property

A natural sufficient condition for the weak U-curve property would be something analogous to convexity in real-valued functions, a property which implies that a local minimum is actually the only global minimum. A local minimum of a convex function of at least two variables is such that, departing from the minimum, the value of the function does not decrease in every direction, and this implies that the point is a global minimum, a feature analogous to the weak U-curve property. Hence, estimated errors which are convex, in some sense, in $\mathbb{L}(\mathcal{H})$ should satisfy the weak U-curve property. We start establishing notation, and then present a sufficient condition for the weak U-curve property.

For each $\mathcal{M} \in \mathbb{L}(\mathcal{H})$, a Lattice Learning Space, define

$$\mathcal{C}^+(\mathcal{M}) := \left\{\mathcal{M}_i \in \mathbb{L}(\mathcal{H}) \,:\, \mathcal{M} \subset \mathcal{M}_i\right\} \qquad \mathcal{C}^-(\mathcal{M}) := \left\{\mathcal{M}_i \in \mathbb{L}(\mathcal{H}) \,:\, \mathcal{M}_i \subset \mathcal{M}\right\}$$

as the models which contain or are contained in $\mathcal{M}$, respectively. Both $\mathcal{C}^+(\mathcal{M})$ and $\mathcal{C}^-(\mathcal{M})$ are complete lattices, on which $\mathcal{M}$ is the least and greatest model, respectively. We define,

for each $\mathcal{M}_i \in \mathcal{C}^+(\mathcal{M}) \setminus \{\mathcal{M}\}$, the lower immediate neighborhood of $\mathcal{M}_i$ relative to $\mathcal{M}$ as

$$N^+(\mathcal{M}_i) := \left\{ \mathcal{M}_j \in \mathcal{C}^+(\mathcal{M}) \,:\, \mathcal{M}_j \subset \mathcal{M}_i, d(\mathcal{M}_j, \mathcal{M}_i) = 1 \right\},$$

and, for each $\mathcal{M}_i \in \mathcal{C}^-(\mathcal{M}) \setminus \{\mathcal{M}\}$, the upper immediate neighborhood relative to $\mathcal{M}$ as

$$N^-(\mathcal{M}_i) := \left\{ \mathcal{M}_j \in \mathcal{C}^-(\mathcal{M}) \,:\, \mathcal{M}_i \subset \mathcal{M}_j, d(\mathcal{M}_j, \mathcal{M}_i) = 1 \right\}.$$

What differs these two sets, both composed by the models in the sub-lattice which has $\mathcal{M}$ as the greatest or least element, which are at a distance one from $\mathcal{M}_i$, is if these models contain or are contained in $\mathcal{M}_i$.

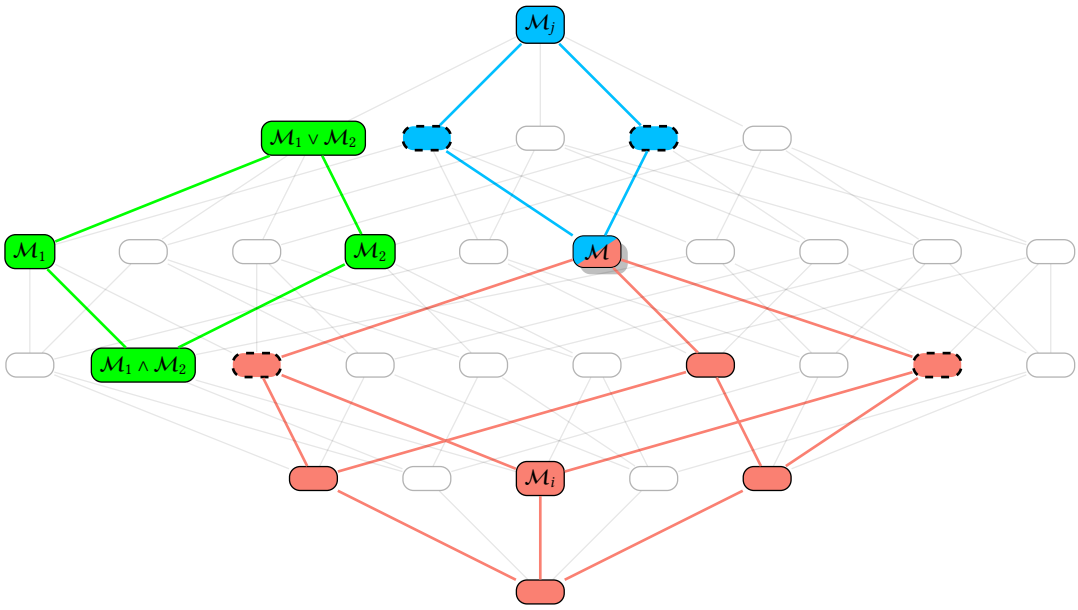

Figure 3.4: *A Learning Space isomorphic to a Boolean lattice, so it is U-curve compatible. The orange nodes represent the lattice $\mathcal{C}^-(\mathcal{M})$, and the blue nodes the lattice $\mathcal{C}^+(\mathcal{M})$, for a given $\mathcal{M}$. The orange dashed nodes are in $N^-(\mathcal{M}_i)$, and the blue dashed nodes are in $N^+(\mathcal{M}_j)$. The green nodes are an example of a pair $\mathcal{M}_1, \mathcal{M}_2$ for which the condition (3.11) of Theorem 3.4 should be satisfied.*

If $\mathcal{M}_j \in N^+(\mathcal{M}_i)$, then $\mathcal{M} \subset \mathcal{M}_j \subset \mathcal{M}_i$, and if $\mathcal{M}_j \in N^-(\mathcal{M}_i)$, then $\mathcal{M}_i \subset \mathcal{M}_j \subset \mathcal{M}$. We say that $\mathbb{L}(\mathcal{H})$ is U-curve compatible if, for every $\mathcal{M} \in \mathbb{L}(\mathcal{H})$,

$$\begin{aligned} N^+(\mathcal{M}_i) = \{\mathcal{M}\} \text{ or } |N^+(\mathcal{M}_i)| \geq 2, \qquad & \forall\, \mathcal{M}_i \in \mathcal{C}^+(\mathcal{M}) \setminus \{\mathcal{M}\} \\ N^-(\mathcal{M}_i) = \{\mathcal{M}\} \text{ or } |N^-(\mathcal{M}_i)| \geq 2, \qquad & \forall\, \mathcal{M}_i \in \mathcal{C}^-(\mathcal{M}) \setminus \{\mathcal{M}\} \end{aligned}$$

i.e., $\mathbb{L}(\mathcal{H})$ is *U-curve compatible* if, for every $\mathcal{M} \in \mathbb{L}(\mathcal{H})$, the lower (upper) immediate neighborhood of all models in $\mathcal{C}^+(\mathcal{M}) \setminus \{\mathcal{M}\}$ ($\mathcal{C}^-(\mathcal{M}) \setminus \{\mathcal{M}\}$) is equal to $\mathcal{M}$ or contain at least two distinct models. The sets defined above are illustrated in Figure 3.4, which presents a U-curve compatible Learning Space. If $\mathbb{L}(\mathcal{H})$ is U-curve compatible, then a simple property of $\hat{L}(\mathcal{M}_i)$ is sufficient for the weak U-curve property.

Theorem 3.4. *Let $\mathbb{L}(\mathcal{H})$ be a U-curve compatible Lattice Learning Space. If all $\mathcal{M}_1, \mathcal{M}_2 \in$*

$\mathbb{L}(\mathcal{H})$ *such that* $d(\mathcal{M}_i, \mathcal{M}_1 \wedge \mathcal{M}_2) = d(\mathcal{M}_i, \mathcal{M}_1 \vee \mathcal{M}_2) = 1$, $i = 1, 2$, *satisfies*

$$\hat{L}(\mathcal{M}_1 \vee \mathcal{M}_2) \geq \hat{L}(\mathcal{M}_1) + \hat{L}(\mathcal{M}_2) - \hat{L}(\mathcal{M}_1 \wedge \mathcal{M}_2), \tag{3.11}$$

with probability 1, *then the weak U-curve property holds for* $\mathbb{L}(\mathcal{H})$ *under* ℓ *and estimator* $\hat{L}$.

Proof. Let $\mathcal{M} \in \mathbb{L}(\mathcal{H})$ be a strong local minimum. We first show that if $\mathcal{M}_i, \mathcal{M}_j \in C^+(\mathcal{M}), \mathcal{M}_i \subset \mathcal{M}_j$, then $\hat{L}(\mathcal{M}_i) \leq \hat{L}(\mathcal{M}_j)$, which implies that $\mathcal{M}$ is a global minimum of $C^+(\mathcal{M})$, since it is its least element. Let $k^\star$ be the size of the greatest continuous chain in $C^+(\mathcal{M})$ which contains $\mathcal{M}$, and define for $\mathcal{M}_1 \subset \mathcal{M}_2 \in \mathbb{L}(\mathcal{H})$,

$$D(\mathcal{M}_1, \mathcal{M}_2) = \max \left\{ \text{Length of continuous chain starting in } \mathcal{M}_1 \text{ and ending in } \mathcal{M}_2 \right\}.$$

We may partition $C^+(\mathcal{M})$ by the greatest position each model occupies in a continuous chain starting in $\mathcal{M}$:

$$C^+(\mathcal{M}, k) = \left\{ \mathcal{M}_i \in C^+(\mathcal{M}) \,:\, D(\mathcal{M}_i, \mathcal{M}) = k \right\},$$

for $2 \leq k \leq k^\star$, and $C^+(\mathcal{M}, 1) = \{\mathcal{M}\}$.

By hypothesis, $\hat{L}(\mathcal{M}) \leq \hat{L}(\mathcal{M}_i)$ for all $\mathcal{M}_i \in C^+(\mathcal{M}, 2)$, as $\mathcal{M}$ is a strong local minimum. We proceed by induction. Assume, for a $k \geq 3$, that $\hat{L}(\mathcal{M}_j) \leq \hat{L}(\mathcal{M}_i)$ for all $\mathcal{M}_i, \mathcal{M}_j \in \cup_{l=1}^{k-1} C^+(\mathcal{M}, l)$ when $\mathcal{M}_j \subset \mathcal{M}_i$. Fix $\mathcal{M}_j \in C^+(\mathcal{M}, k)$ and $\mathcal{M}_j^{(1)} \in C^+(\mathcal{M}, k-1)$ with $\mathcal{M}_j^{(1)} \subset \mathcal{M}_j$. Note that $\mathcal{M}_j^{(1)} \in N^+(\mathcal{M}_j) \subset \cup_{l=1}^{k-1} C^+(\mathcal{M}, l)$ and, as $|N^+(\mathcal{M}_j)| \geq 2$, since $\mathbb{L}(\mathcal{H})$ is U-curve compatible, there exists another $\mathcal{M}_j^{(2)} \in N^+(\mathcal{M}_j)$ such that $\mathcal{M}_j = \mathcal{M}_j^{(1)} \vee \mathcal{M}_j^{(2)}$.

Therefore,

$$\hat{L}(\mathcal{M}_j) \geq \hat{L}(\mathcal{M}_j^{(1)}) + \hat{L}(\mathcal{M}_j^{(2)}) - \hat{L}(\mathcal{M}_j^{(1)} \wedge \mathcal{M}_j^{(2)}) \geq \hat{L}(\mathcal{M}_j^{(1)}) \tag{3.12}$$

by the induction hypothesis, as $\hat{L}(\mathcal{M}_j^{(2)}) - \hat{L}(\mathcal{M}_j^{(1)} \wedge \mathcal{M}_j^{(2)}) \geq 0$ since $\mathcal{M}_j^{(1)} \wedge \mathcal{M}_j^{(2)} \subset \mathcal{M}_j^{(2)}$ and both are in $\cup_{l=1}^{k-1} C^+(\mathcal{M}, l)$. From (3.12), and the induction hypothesis, it follows that $\hat{L}(\mathcal{M}_i) \leq \hat{L}(\mathcal{M}_j)$ for all $\mathcal{M}_i, \mathcal{M}_j \in C^+(\mathcal{M}), \mathcal{M}_i \subset \mathcal{M}_j$, as there is a $\mathcal{M}_j^{(1)} \in N^+(\mathcal{M}_j)$ such that $\mathcal{M}_i \subset \mathcal{M}_j^{(1)} \subset \mathcal{M}_j$ which implies $\hat{L}(\mathcal{M}_i) \leq \hat{L}(\mathcal{M}_j^{(1)}) \leq \hat{L}(\mathcal{M}_j)$.

With an analogous deduction and the inequality

$$\hat{L}(\mathcal{M}_1 \wedge \mathcal{M}_2) \geq \hat{L}(\mathcal{M}_1) + \hat{L}(\mathcal{M}_2) - \hat{L}(\mathcal{M}_1 \vee \mathcal{M}_2), \tag{3.13}$$

we can show that if $\mathcal{M}_i, \mathcal{M}_j \in C^-(\mathcal{M}), \mathcal{M}_i \subset \mathcal{M}_j$, then $\hat{L}(\mathcal{M}_i) \geq \hat{L}(\mathcal{M}_j)$, which implies that $\mathcal{M}$ is also the global minimum of $C^-(\mathcal{M})$, as it is its greatest element. □

Remark 3.5. *From the proof of Theorem 3.4, one can deduce that its premises may be loosened. Condition (3.11) need not be satisfied by all* $\mathcal{M}_1, \mathcal{M}_2 \in \mathbb{L}(\mathcal{H})$ *such that* $d(\mathcal{M}_i, \mathcal{M}_1 \wedge \mathcal{M}_2) = d(\mathcal{M}_i, \mathcal{M}_1 \vee \mathcal{M}_2) = 1, i = 1, 2$. *It is necessary only that, for every* $\mathcal{M} \in \mathbb{L}(\mathcal{H})$ *and every* $\mathcal{M}_1 \in C^+(\mathcal{M}) \setminus \{\mathcal{M}\}$ $(C^-(\mathcal{M}) \setminus \{\mathcal{M}\})$ *with* $d(\mathcal{M}, \mathcal{M}_1) > 1$, *there are* $\mathcal{M}_1^{(i)} \in C^+(\mathcal{M}) \setminus \{\mathcal{M}\}$ $(C^-(\mathcal{M}) \setminus \{\mathcal{M}\})$, $i = 1, 2$, *such that* $\mathcal{M}_1 = \mathcal{M}_1^{(1)} \vee \mathcal{M}_1^{(2)} (= \mathcal{M}_1^{(1)} \wedge \mathcal{M}_1^{(2)})$,

and condition (3.11) is satisfied by $\mathcal{M}_1^{(1)}, \mathcal{M}_1^{(2)}$. *Moreover, Theorem 3.4 may be adapted to* $\mathbb{L}(\mathcal{H})$ *which is not a lattice by adding further constraints to it.*

We see in the proof of Theorem 3.4 that (3.11) is also a sufficient condition for the inf-weak and sup-weak U-curve properties. Indeed, the proof establishes that if $\mathcal{M}$ is a sup-strong local minimum, then it is the global minimum of $\mathcal{C}^+(\mathcal{M})$, what is equivalent to the sup-weak U-curve property. Similarly, with the inequality (3.13), we can show that if $\mathcal{M}$ is a inf-strong local minimum, then it is a global minimum of $\mathcal{C}^-(\mathcal{M})$, what is equivalent to the inf-weak U-curve property. We state this result as a corollary.

Corollary 3.6. *Condition* (3.11) *is sufficient for both the sup-weak and inf-weak U-curve property.*

Remark 3.7. *If both the sup-weak and inf-weak U-curve properties are satisfied, then clearly the strong U-curve property also holds. Nevertheless, the opposite is not necessarily true, as the strong U-curve property may hold, but either the sup-weak or inf-weak may not.*

An example of a pair $\mathcal{M}_1, \mathcal{M}_2$ which should satisfy (3.11) is presented in Figure 3.4. Assuming, without loss of generality, that $\hat{L}(\mathcal{M}_1) \geq \hat{L}(\mathcal{M}_2)$, and rewriting (3.11) as

$$\hat{L}(\mathcal{M}_1 \vee \mathcal{M}_2) - \hat{L}(\mathcal{M}_1) \geq \hat{L}(\mathcal{M}_2) - \hat{L}(\mathcal{M}_1 \wedge \mathcal{M}_2),$$

we see that the increase on $\hat{L}$ when we go from $\mathcal{M}_1$ to $\mathcal{M}_1 \vee \mathcal{M}_2$ is greater than when we go from $\mathcal{M}_1 \wedge \mathcal{M}_2$ to $\mathcal{M}_2$. This feature is analogous to that observed on convex functions, that is, the increment of the function increases when its inputs increase, in which *increase in inputs* in this context is according to relation $\subset$ and the value of $\hat{L}$, when subsets are not related. Hence, we call property (3.11) *Lattice Convexity*. As is the case with convex functions, (strong) local minimums are global minimums, but, since $\mathbb{L}(\mathcal{H})$ is a poset, this is true only for the models which are related to the local minimum.

Our definition of weak U-curve property is not as restrictive as lattice convexity, since the latter implies that the chains are monotone below and above the strong minimum, as can be inferred from the proof of Theorem 3.4, while in our definition there may be, for example, another weak local minimum in a continuous chain that passes through a strong minimum, although this weak minimum has a loss greater than the strong one.

Although the Partition Lattice Learning Space satisfies the sup-weak U-curve property, it does not satisfy the weak U-curve property. We now present, as a counterexample, a sample on which a strong local minimum is not a global minimum of all chains that contain it.

In Table 3.2, we present the empirical joint frequencies of a training and validation sample. Observe that $\pi = \{\{1\}, \{2,4\}, \{3,5\}, \{6\}\}$ is a strong local minimum and that $\hat{L}(\mathcal{H}|_\pi) = 0.25$. This can be seen by noting that breaking a partition (upper neighbor), or uniting a partition (lower neighbor), does not change the estimated error, and hence it is indeed a strong local minimum. However, partition $\pi' = \{\{1,2,3,4,5,6\}\}$, which is such that $\pi' \leq \pi$, has a lesser estimated error, that is $\hat{L}(\mathcal{H}|_{\pi'}) = 0.2$, hence π is not a global minimum. This implies that the weak U-curve property does not hold in the Partition

$\mathcal{X}$	Training		Validation	
	0	1	0	1
1	0.45	0	0.2	0.25
2	0	0.1	0	0.09
3	0	0.1	0	0.09
4	0	0.1	0	0.09
5	0	0.1	0	0.09
6	0	0.15	0	0.09

Table 3.2: *Counterexample of empirical training and validation joint frequencies under which a strong local minimum of the Partition Lattice Learning is not a global minimum of all chains that pass through it. The strong local minimum is* $\pi = \{\{1\},\{2,4\},\{3,5\},\{6\}\}$*, with* $\hat{L}(\mathcal{H}|_\pi) = 0.25$*, but* $\hat{L}(\mathcal{H}|_{\pi'}) = 0.2$ *with* $\pi' = \{\{1,2,3,4,5,6\}\}$*, and* $\pi' \leq \pi$*. There are more strong local minimums which are not global minimums under these empirical joint frequencies.*

Lattice Learning Space when ℓ is the simple loss function and $\hat{L}$ is of form[2] (2.8), as for it to hold, it should be true for *any* possible sample of *any* possible distribution P.

Remark 3.8. *In the Partition Lattice Learning Space, condition* (3.11) *is not satisfied for all pair of sets* $\mathcal{M}_1, \mathcal{M}_2$ *with* $d(\mathcal{M}_i, \mathcal{M}_1 \wedge \mathcal{M}_2) = d(\mathcal{M}_i, \mathcal{M}_1 \vee \mathcal{M}_2) = 1, i = 1, 2$*. Consider the pair generated by* $\pi_1 = \{a_1, a_2, b, c\}$ *and* $\pi_2 = \{a, b_1, b_2, c\}$ *in which* $a = a_1 \cup a_2$ *and* $b = b_1 \cup b_2$*. Now, to get from* $\pi_1 \wedge \pi_2 = \{a, b, c\}$ *to* $\pi_1 \vee \pi_2 = \{a_1, a_2, b_1, b_2, c\}$ *in this case, we perform two partition breaks, each one in distinct partitions and increasing the estimated out-of-sample error by* $l_1, l_2 \in \mathbb{R}$*, independently of the order of such breaks, by the argument at inequality* (3.7)*. From this, we may conclude that, say,*

$$\hat{L}(\mathcal{H}|_{\pi_1}) = \hat{L}(\mathcal{H}|_{\pi_1 \wedge \pi_2}) + l_1 \qquad \hat{L}(\mathcal{H}|_{\pi_2}) = \hat{L}(\mathcal{H}|_{\pi_1 \wedge \pi_2}) + l_2$$

and

$$\hat{L}(\mathcal{H}|_{\pi_1 \vee \pi_2}) = \hat{L}(\mathcal{H}|_{\pi_1}) + l_2 = \hat{L}(\mathcal{H}|_{\pi_2}) + l_1 = \hat{L}(\mathcal{H}|_{\pi_1 \wedge \pi_2}) + l_1 + l_2.$$

Hence

$$\hat{L}(\mathcal{H}|_{\pi_1 \wedge \pi_2}) + \hat{L}(\mathcal{H}|_{\pi_1 \vee \pi_2}) = \hat{L}(\mathcal{H}|_{\pi_1}) + \hat{L}(\mathcal{H}|_{\pi_2}),$$

which implies the condition (3.11)*.*

On the other hand, consider partitions $\pi_1 = \{a_1 \cup a_2, a_3, b\}$ *and* $\pi_2 = \{a_1 \cup a_3, a_2, b\}$*, and denote* $a = a_1 \cup a_2 \cup a_3$*. To get from* $\pi_1 \wedge \pi_2 = \{a, b\}$ *to* $\pi_1 \vee \pi_2 = \{a_1, a_2, a_3, b\}$ *we perform two partition breaks on the same root partition block* a*. In this case, the order in which* a *is broken may influence the increment of the estimated out-of-sample error and the condition of Theorem* 3.2 *may be violated. Consider the empirical distributions obtained from a training*

[2] If there are m pairs of samples with a behavior analogous to that of the empirical frequencies in Table 3.2, then the weak U-curve property will not hold when $m > 1$.

and validation samples presented in Table 3.3. In this example,

$$\begin{cases} \hat{L}(\mathcal{H}|_{\pi_1 \wedge \pi_2}) = 0.48 \\ \hat{L}(\mathcal{H}|_{\pi_1}) = 0.44 \\ \hat{L}(\mathcal{H}|_{\pi_2}) = 0.41 \\ \hat{L}(\mathcal{H}|_{\pi_1 \vee \pi_2}) = 0.33 \end{cases} \implies \begin{cases} \hat{L}(\mathcal{H}|_{\pi_1 \wedge \pi_2}) + \hat{L}(\mathcal{H}|_{\pi_1 \vee \pi_2}) = 0.81 \\ \hat{L}(\mathcal{H}|_{\pi_1}) + \hat{L}(\mathcal{H}|_{\pi_2}) = 0.85 \end{cases}$$

so condition (3.11) is not satisfied.

When partition $\{a_1, a_2, a_2\}$ is broken into $\{a_1, a_3\}, a_2$ the error decreases 0.07, while when we perform the break of partition $\{a_1, a_2\}$ into a_1, a_2 it decreases 0.11, hence the variation of the error when we perform a break to free point a_2 depends on the order we perform the break, that is, if we free a_3 before or after freeing a_2.

$\mathcal{X}$	Training		Validation		$\hat{h}_{\pi_1 \wedge \pi_2}$	$\hat{h}_{\pi_1}$	$\hat{h}_{\pi_2}$	$\hat{h}_{\pi_1 \vee \pi_2}$
	0	1	0	1				
a_1	0.18	0.07	0.2	0.05	1	0	0	0
a_2	0.02	0.11	0.01	0.12	1	0	1	1
a_3	0.01	0.11	0.02	0.10	1	1	0	1
b	0.2	0.3	0.25	0.25	1	1	1	1

Table 3.3: *Empirical distributions of a training and validation samples when $\mathcal{X} = \{a_1, a_2, a_3, b\}$ is a set with four points, and estimated hypothesis for partitions $\pi_1 \wedge \pi_2 = \{\{a_1, a_2, a_3\}, b\}, \pi_1 = \{\{a_1, a_2\}, a_3, b\}, \pi_2 = \{\{a_1, a_3\}, a_2, b\}$ and $\pi_1 \vee \pi_2 = \{a_1, a_2, a_3, b\}$.*

3.5 A generic U-curve algorithm

The U-curve algorithm was first proposed by [133] in the context of variable selection, and showed by [128] to be NP-hard (see also [55]). However, the algorithm avoids an exhaustive search of $\mathbb{L}(\mathcal{H})$ and permits solving problems which could not be solved by exhaustive search, due to limitations on computer power. Indeed, it has been applied to a variety of problems [8, 32, 55, 130, 131]. In this section, we discuss generic U-curve algorithms to solve (3.1) when the weak or sup-weak U-curve property is satisfied.

To apply the U-curve algorithm in this context, we assume that an optimizer which, given $\mathcal{M} \in \mathbb{L}(\mathcal{H})$, computes $\hat{L}(\mathcal{M})$, is implemented. Moreover, for each $\mathcal{M} \in \mathbb{L}(\mathcal{H})$ we define

$$N(\mathcal{M}) = \left\{ \mathcal{M}_i \in \mathbb{L}(\mathcal{H}) \,:\, \mathcal{M} \subset \mathcal{M}_i, d(\mathcal{M}, \mathcal{M}_i) = 1 \right\}$$

as the models in $\mathbb{L}(\mathcal{H})$ which immediately contains $\mathcal{M}$, and

$$n(\mathcal{M}) = \left\{ \mathcal{M}_i \in \mathbb{L}(\mathcal{H}) \,:\, \mathcal{M}_i \subset \mathcal{M}, d(\mathcal{M}, \mathcal{M}_i) = 1 \right\}$$

as the models in $\mathbb{L}(\mathcal{H})$ immediately contained in $\mathcal{M}$. We call $N(\mathcal{M})$ and $n(\mathcal{M})$ the upper and lower immediate neighborhood of $\mathcal{M}$, respectively.

Next, we need auxiliary algorithms to compute if a given model $\mathcal{M}$ is a strong local minimum, and a sup-strong local minimum, of $\mathbb{L}(\mathcal{H})$ by estimating the error of all models in the immediate neighborhoods of $\mathcal{M}$, and comparing them to the estimated error of $\mathcal{M}$. The *MinimumExhausted* and *SupMinimumExhausted* algorithms are presented in Algorithms 1 and 2, and return TRUE if $\mathcal{M}$ is a strong local minimum and a sup-strong minimum, respectively, and FALSE otherwise.

Algorithm 1 *MinimumExhausted* auxiliary algorithm.

Input: $\mathcal{M}, \hat{L}(\mathcal{M})$

for $\mathcal{M}' \in N(\mathcal{M}) \cup n(\mathcal{M})$ **do**
  if $\hat{L}(\mathcal{M}') < \hat{L}(\mathcal{M})$ **then**
    return FALSE
return TRUE

Algorithm 2 *SupMinimumExhausted* auxiliary algorithm.

Input: $\mathcal{M}, \hat{L}(\mathcal{M})$

for $\mathcal{M}' \in N(\mathcal{M})$ **do**
  if $\hat{L}(\mathcal{M}') < \hat{L}(\mathcal{M})$ **then**
    return FALSE
return TRUE

Taking advantage of the weak U-curve property, the U-curve algorithm presented in Algorithm 3 solves (3.1), and is as follows. We first select a model $\mathcal{M} \in \mathbb{L}(\mathcal{H})$ and calculate $\hat{L}(\mathcal{M})$. Then, we apply the *MinimumExhausted* algorithm to $\mathcal{M}$ to establish if it is a strong local minimum. If it is a strong local minimum, we store $\mathcal{M}$ and exclude from $\mathbb{L}(\mathcal{H})$ all models which contain or are contained in $\mathcal{M}$, as we know they have an equal or greater estimated error by the weak U-curve property, obtaining a $\mathbb{L} \subset \mathbb{L}(\mathcal{H})$. If $\mathbb{L} \neq \emptyset$, we start the process again by selecting a model $\mathcal{M} \in \mathbb{L}$.

If $\mathcal{M}$ is not a strong local minimum and there exists a $\mathcal{M}' \in (N(\mathcal{M}) \cup n(\mathcal{M})) \cap \mathbb{L}$ with $\hat{L}(\mathcal{M}') < \hat{L}(\mathcal{M})$, we exclude $\mathcal{M}$ from $\mathbb{L}$, make $\mathcal{M} = \mathcal{M}'$ and start the algorithm again. Otherwise, if there is no such $\mathcal{M}'$, we exclude $\mathcal{M}$ from $\mathbb{L}$ and start the algorithm from another $\mathcal{M} \in \mathbb{L}$. We proceed this way until $\mathbb{L} = \emptyset$. Finally, we select the global minimums among the stored strong local minimums.

On the other hand, by taking advantage of the sup-weak U-curve property, the U-curve algorithm presented in Algorithm 4 solves (3.1), and is as follows. We first select a model $\mathcal{M} \in \mathbb{L}(\mathcal{H})$ and calculate $\hat{L}(\mathcal{M})$. Then, we apply the *SupMinimumExhausted* algorithm to $\mathcal{M}$ to establish if it is a sup-strong local minimum. If it is a sup-strong local minimum, we store $\mathcal{M}$ and exclude from $\mathbb{L}(\mathcal{H})$ all models which contain $\mathcal{M}$, as we know they have an equal or greater estimated error by the sup-weak U-curve property, obtaining a $\mathbb{L} \subset \mathbb{L}(\mathcal{H})$. If $\mathbb{L} \neq \emptyset$, we start the process again by selecting a model $\mathcal{M} \in \mathbb{L}$.

If $\mathcal{M}$ is not a sup-strong local minimum and there exists a $\mathcal{M}' \in N(\mathcal{M}) \cap \mathbb{L}$ with $\hat{L}(\mathcal{M}') < \hat{L}(\mathcal{M})$, we exclude $\mathcal{M}$ from $\mathbb{L}$, make $\mathcal{M} = \mathcal{M}'$ and start the algorithm again. Otherwise, if there is no such $\mathcal{M}'$, we exclude $\mathcal{M}$ from $\mathbb{L}$ and start the algorithm from another $\mathcal{M} \in \mathbb{L}$. We proceed this way until $\mathbb{L} = \emptyset$. Finally, we select the global minimums among the stored sup-strong local minimums.

The pseudocode presented in Algorithms 1, 2, 3 and 4 do not treat the technical details of an implementation of the U-curve algorithm, and present only its main ideas, as it is out of the scope of this thesis to further study U-curve algorithms.

Algorithm 3 Generic U-curve algorithm when the weak U-curve property is satisfied.

Ensure: $\mathcal{M} \in \mathbb{L}(\mathcal{H})$, Cost $\leftarrow \hat{L}(\mathcal{M})$, $\mathbb{L} \leftarrow \mathbb{L}(\mathcal{H})$, LocalMinimuns $\leftarrow \emptyset$

while $\mathbb{L} \neq \emptyset$ **do**
  if $MinimumExhausted(\mathcal{M}, Cost)$ **then**
    LocalMinimuns $\leftarrow$ LocalMinimuns $\cup \{\mathcal{M}\}$
    $\mathbb{L} \leftarrow \mathbb{L} \setminus \{\mathcal{M}_i \in \mathbb{L} : \mathcal{M}_i \subset \mathcal{M}$ or $\mathcal{M} \subset \mathcal{M}_i\}$
    if $\mathbb{L} \neq \emptyset$ **then**
      $\mathcal{M} \leftarrow \mathcal{M}_i, \mathcal{M}_i \in \mathbb{L}$
      Cost $\leftarrow \hat{L}(\mathcal{M})$
  else
    if $\exists \mathcal{M}' \in (N(\mathcal{M}) \cup n(\mathcal{M})) \cap \mathbb{L}$ s.t $\hat{L}(\mathcal{M}') < \hat{L}(\mathcal{M})$ **then**
      $\mathbb{L} \leftarrow \mathbb{L} \setminus \{\mathcal{M}\}$
      $\mathcal{M} \leftarrow \mathcal{M}'$
      Cost $\leftarrow \hat{L}(\mathcal{M}')$
    else
      $\mathbb{L} \leftarrow \mathbb{L} \setminus \{\mathcal{M}\}$
      if $\mathbb{L} \neq \emptyset$ **then**
        $\mathcal{M} \leftarrow \mathcal{M}_i, \mathcal{M}_i \in \mathbb{L}$
        Cost $\leftarrow \hat{L}(\mathcal{M})$
return LocalMinimuns

Remark 3.9. *Due to ties of error $\hat{L}$, the global minimum with the least VC dimension may not be in the set returned by Algorithm 3, since another global minimum which contains it may have been returned instead. To force the return of $\hat{\mathcal{M}}$ one has to check if the neighbors of a strong local minimum, with lesser VC dimension and same error, are also strong local minimums. We did not add this feature to Algorithm 3 to better present its main idea, but it must be regarded when implementing the algorithm for a specific case. This is not an issue in Algorithm 4, since a model with lesser VC dimension is never excluded from $\mathbb{L}(\mathcal{H})$ without either checking if it is a sup-strong local minimum, or concluding that it is not the global minimum with the least VC dimension since it contains a sup-strong local minimum.*

Remark 3.10. *Algorithm 4 may be easily modified if the inf-weak U-curve property is satisfied instead.*

Algorithm 4 Generic U-curve algorithm when the sup-weak U-curve property is satisfied.

Ensure: $\mathcal{M} \in \mathbb{L}(\mathcal{H})$, Cost $\leftarrow \hat{L}(\mathcal{M})$, $\mathbb{L} \leftarrow \mathbb{L}(\mathcal{H})$, LocalMinimuns $\leftarrow \emptyset$

while $\mathbb{L} \neq \emptyset$ **do**
  if $SupMinimumExhausted(\mathcal{M}, Cost)$ **then**
    LocalMinimuns $\leftarrow$ LocalMinimuns $\cup \{\mathcal{M}\}$
    $\mathbb{L} \leftarrow \mathbb{L} \setminus \{\mathcal{M}_i \in \mathbb{L} : \mathcal{M} \subset \mathcal{M}_i\}$
    if $\mathbb{L} \neq \emptyset$ **then**
      $\mathcal{M} \leftarrow \mathcal{M}_i, \mathcal{M}_i \in \mathbb{L}$
      Cost $\leftarrow \hat{L}(\mathcal{M})$
  else
    if $\exists \mathcal{M}' \in N(\mathcal{M}) \cap \mathbb{L}$ s.t $\hat{L}(\mathcal{M}') < \hat{L}(\mathcal{M})$ **then**
      $\mathbb{L} \leftarrow \mathbb{L} \setminus \{\mathcal{M}\}$
      $\mathcal{M} \leftarrow \mathcal{M}'$
      Cost $\leftarrow \hat{L}(\mathcal{M}')$
    else
      $\mathbb{L} \leftarrow \mathbb{L} \setminus \{\mathcal{M}\}$
      if $\mathbb{L} \neq \emptyset$ **then**
        $\mathcal{M} \leftarrow \mathcal{M}_i, \mathcal{M}_i \in \mathbb{L}$
        Cost $\leftarrow \hat{L}(\mathcal{M})$
return LocalMinimuns

3.6 Improving the U-curve algorithm

There are a handful of features which could be implemented to the generic Algorithms 3 and 4 to improve their performance. The generic U-curve algorithms will always return a result, since every time we exhaust/sup-exhaust the neighborhood of a model we delete it from $\mathbb{L}$ so, in the worst case, the algorithm would perform an exhaustive search of[3] $\mathbb{L}(\mathcal{H})$. Nevertheless, we can stop the algorithm early and apply an exhaustive search on the models remaining in $\mathbb{L}$, when the cardinality of $\mathbb{L}$ is within the reach of an exhaustive search. This may improve the algorithm since, as the cardinality of $\mathbb{L}$ decreases, it will contain less strong/sup-strong local minimums, hence the prunes of $\mathbb{L}$ due to strong/sup-strong local minimums will get rarer, and the resources employed to exhaust/sup-exhaust the neighborhood of each model in $\mathbb{L}$ could be better employed to search it exhaustively.

For example, if a model in $\mathbb{L}$ has more immediate neighbors than there are points in $\mathbb{L}$, an exhaustive search of $\mathbb{L}$ could be more efficient than continuing with the U-curve algorithm. Therefore, one manner of improving the efficiency of the U-curve algorithm is to stop it when $|\mathbb{L}| < c$, for a given constant c, and then exhaustively search $\mathbb{L}$ for its global minimums, which could then be compared with the strong/sup-strong local minimums found by the U-curve algorithm to find the global minimums of $\mathbb{L}(\mathcal{H})$, and the solution of (3.1).

Another manner of improving the efficiency of the algorithm is to delete from $\mathbb{L}$ all neighbors with greater loss, visited during the *MinimumExhausted* and *SupMinimumEx-*

[3] Actually, it would exhaust or sup-exhaust the neighborhood of all models, what is redundant and more complex than an exhaustive search of $\mathbb{L}(\mathcal{H})$.

hausted calls. Since these models have greater error than the exhausted model, they cannot be global minimums, hence may be disregarded. In some instances, it could be a good option to not stop the *MinimumExhausted* and *SupMinimumExhausted* algorithms when a model with lesser error is found, but rather calculate the error for all considered neighbors to exclude from $\mathbb{L}$ all of them with greater error, and restart the algorithm in the neighbor with the least error, when the exhausted model is not a strong/sup-strong local minimum. Algorithm 4 with all features discussed so far is applied to simulated data in Section 4.1.

Moreover, Algorithms 3 and 4 may be performed in parallel by taking advantage of a specific feature of the lattice structure of $\mathbb{L}(\mathcal{H})$, improving the speed of the algorithm (see the results in [55] for the Boolean lattice of variable selection). Also, besides being a sufficient condition to the weak U-curve property, (3.11) is also a tool for increasing the efficiency of a U-curve algorithm. Assume that $\mathcal{M}_1$ and $\mathcal{M}_2$ satisfy (3.11), and that $\min\left(\hat{L}(\mathcal{M}_1), \hat{L}(\mathcal{M}_2)\right) > \hat{L}(\mathcal{M}_1 \wedge \mathcal{M}_2)$. Then

$$\hat{L}(\mathcal{M}_1 \vee \mathcal{M}_2) \geq \hat{L}(\mathcal{M}_1) + \hat{L}(\mathcal{M}_2) - \hat{L}(\mathcal{M}_1 \wedge \mathcal{M}_2) > \hat{L}(\mathcal{M}_1 \wedge \mathcal{M}_2),$$

so after visiting the models $\mathcal{M}_1 \wedge \mathcal{M}_2$, $\mathcal{M}_1$ and $\mathcal{M}_2$, and noting the increase in the estimated error, one does not need to visit $\mathcal{M}_1 \vee \mathcal{M}_2$, since the estimated error will increase. This fact may be employed to establish search strategies for U-curve algorithms.

The efficiency of the algorithms may be greatly improved if one employs them to find suboptimal solutions. For example, one may apply a stochastic U-curve algorithm, in which we sample a model in $\mathbb{L}(\mathcal{H})$ every time we restart the algorithm and one does not perform prunes in $\mathbb{L}(\mathcal{H})$, but rather stop the algorithm when a "sufficient" number of strong/sup-strong local minimums, or a "good enough" local minimum, is found. A version of this algorithm was implemented for the Partition Lattice Learning Space in [32] where more details can be found, and is applied to simulated data in Section 4.1. Furthermore, in such a stochastic algorithm, one could sample a certain number of neighbors and consider as a strong/sup-strong local minimum any model which has a cost lesser than the sampled ones, what would speed up the algorithm at the cost of considering as strong/sup-strong minimums models which are not.

Apart from features such as the ones discussed above, there are two main theoretical considerations when developing a U-curve algorithm which can impact its efficiency. The first one is how to prune the Learning Space when one finds a local minimum, i.e., find all models which contain or are contained in a local minimum and store this information in some way to not visit these models. This is an important question, since $\mathbb{L}(\mathcal{H})$ may be too great to be stored.

For example, the Partition Lattice Learning Space with 100 points in the domain $\mathcal{X}$ has a cardinality of the order 10^{115}, so it is not possible to store it. Hence, one needs a computationally cheap manner of determining if a $\mathcal{M} \in \mathbb{L}(\mathcal{H})$ is in $\mathbb{L}$, or if $\mathbb{L} = \emptyset$, in a given step of the algorithm, apart from storing $\mathbb{L}$ and testing the inclusion $\mathcal{M} \in \mathbb{L}$ or if $\mathbb{L} = \emptyset$. When the Learning Space is a Boolean lattice, the inclusion and emptiness test may be explicitly computed via efficient representations of the nodes in the lattice, such as *Reduced Ordered Binary Decision Diagram* (ROBDD) [31]. However, an efficient

representation of the nodes in general lattices, and in special in Partition Lattices, is missing, and the instantiation of optimal Algorithm 4 to practical problems of interest passes through the development of these representations for non-Boolean lattices.

The second consideration is which neighbor of a model to visit first in the *MinimumExhausted* and *SupMinimumExhausted* algorithms: can one search $\mathbb{L}(\mathcal{H})$ more efficiently by going down certain paths? We leave both these considerations as open problems, since technical details of the U-curve algorithm are out of the scope of this thesis.

3.7 Next steps

We have concluded the formalization of our Model Selection framework, which we have established to be data-driven systematic, consistent and non-exhaustive. It remains now to illustrate how the method may be employed to solve real learning problems, and to discuss the implications and perspectives of it.

In the first section of Chapter 4, we simulate data sets to learn via the Partition Lattice Learning Space, to outline some of its properties established in Chapters 2 and 3. Then, in Section 4.2 we illustrate how the method may be employed to forecast a binary sequence under a Markov chain framework, and apply it to develop an investment strategy for bitcoin.

In Section 4.3, we propose a multilayer W-operator to learn handwritten digits which is estimated via a suboptimal U-curve algorithm, and then applied to recognize the zero digit in the MNIST data set [89]. Finally, in Section 4.4, we instantiate the method to learn interval Boolean functions, which are specially suitable to transform images, and propose a Learning Space for it that satisfies the sup-weak U-curve property.

In Chapter 5, we discuss the main implications of the proposed method, its qualities and pitfalls, and a myriad of promising topics and open problems for future researches.

Chapter 4

Applications

In this chapter, we illustrate how one may take advantage of a Learning Space to solve practical problems. In Section 4.1, we present simulations of learning on the Partition Lattice Learning Space, which illustrate some interesting theoretical results of Chapters 2 and 3. In Section 4.2, we propose a suboptimal U-curve algorithm to learn a classifier suitable for forecasting a sequence of binary values under a Markov chain framework, and apply it to obtain an investment strategy for bitcoin.

In Section 4.3, we propose a multilayer W-operator for recognizing handwritten digits and apply it to the MNIST data set [89]. Finally, in Section 4.4, we instantiate learning via Learning Spaces for interval Boolean functions, which are suitable for problems involving image transformation and classification, and show that the subset of the Partition Lattice Learning Space considered satisfies the sup-weak U-curve property.

We expect with these applications to illustrate the potential of Learning Space based techniques, rather than develop definitive algorithms to solve practical problems, an endeavor we leave for future researches.

Code and data availability

The code and data sets used in this chapter are available at https://github.com/dmarcondes/PhDthesis. The application in Section 4.1 was implemented in **R** [122] and the code is organized in the **R** package **partitionUcurve**. The application in Section 4.2 was also implemented in **R**, and the code is organized in the **R** package **MarkovLS**. The application in Section 4.3 was implemented in **C**, and the results of the algorithm were analyzed with **R**. All code and **R** packages are in the GitHub repository. The code of applications in Sections 4.1 and 4.2 ran in a computer with a Intel Core i7-8565U CPU @ 1.80GHz × 8 processor and 16GB of RAM memory. The code of application in Section 4.3 ran in a server with a Intel(R) Xeon(R) Gold 6144 CPU @ 3.50GHz x 32 processor and 512GB of RAM memory.

4.1 Learning via the Partition Lattice Learning Space

In order to illustrate learning via the Partition Lattice Learning Space, and some theoretical results of Chapters 2 and 3, we simulate the learning with samples from four joint distributions, for $\mathcal{X} = \{1, \dots, 8\}$ and $\mathcal{Y} = \{0, 1\}$, which are presented in Table 4.1, that we call Examples 1 to 4. These joint distributions are such that X is uniformly distributed and $L(h^\star) = 0.23125$, when considering the simple loss function, but $\epsilon^\star$ increases from Example 1 to 3, and Example 4 has the same $\epsilon^\star$ as Example 2.

The joint distributions have increasing Conditional Entropy [94], from Example 1 to 4, which is defined as

$$H(Y|X) = -\sum_{x \in \mathcal{X}, y \in \mathcal{Y}} \mathbb{P}(X = x, Y = y) \log \frac{\mathbb{P}(X = x, Y = y)}{\mathbb{P}(X = x)},$$

that is a measure of concentration of the conditional distributions of Y given $X = x$, $x \in \mathcal{X}$. As more concentrated the conditional distributions are, i.e., as lower the Conditional Entropy, more *distant* are, in average, the conditional probabilities of 0 and of 1 given X, hence fewer samples are needed to decide which probability is greater[1], what is equivalent to decide the value of $h^\star$ at a point x.

x	Example 1		Example 2		Example 3		Example 4	
	$p(0,x)$	$p(1,x)$	$p(0,x)$	$p(1,x)$	$p(0,x)$	$p(1,x)$	$p(0,x)$	$p(1,x)$
1	0.00625	0.11875	0.00625	0.11875	0.00625	0.11875	0.00625	0.11875
2	0.00250	0.12250	0.00250	0.12250	0.00250	0.12250	0.00250	0.12250
3	0.06375	0.06125	0.06500	0.06000	0.07000	0.05500	0.08750	0.03750
4	0.06375	0.06125	0.06500	0.06000	0.07000	0.05500	0.08750	0.03750
5	0.06375	0.06125	0.06500	0.06000	0.07000	0.05500	0.08750	0.03750
6	0.11250	0.01250	0.11250	0.01250	0.11250	0.01250	0.11250	0.01250
7	0.10000	0.02500	0.10000	0.02500	0.10000	0.02500	0.06500	0.06000
8	0.12375	0.00125	0.12000	0.00500	0.10500	0.02000	0.08750	0.03750
$\epsilon^\star$	0.00250		0.00500		0.01500		0.00500	
CE	0.40711		0.42419		0.49252		0.61935	
$L(h^\star)$	0.23125		0.23125		0.23125		0.23125	

$p(0,x) = \mathbb{P}(Y = 0, X = x)$, $p(1,x) = \mathbb{P}(Y = 1, X = x)$, CE: Conditional Entropy

Table 4.1: *Joint distributions considered in each example. All of them have a same $L(h^\star)$, but are, from Example 1 to 4, of increasing Conditional Entropy and $\epsilon^\star$, except for Example 4 which has the same $\epsilon^\star$ as Example 2.*

For each example, we simulated 100 samples of each size in $\{64, 96, 128, 160, 192, 224, 256\}$, which were then divided into a training sample (1/2), validation sample (1/4) and indepen-

[1] This can be established from a statistical perspective by noting that, as more distant these probabilities are from each other, or equivalently from 0.5, fixed a confidence level, fewer samples are needed to obtain a confidence interval for them which do not contain 0.5, what is enough to determine the value of $h^\star$ with a high rate of success relative to the fixed confidence level.

dent sample to learn on $\hat{\mathcal{M}}$ (1/4). For each sample, we learned hypothesis $\hat{h}$ via ERM with the whole sample (union of training, validation and independent samples), and applied two variations of Algorithm 4:

- **Optimal**: Algorithm 4 was applied until an exclusion of models from $\mathbb{L}$ due to a sup-strong local minimal implied $|\mathbb{L}| < 1,000$. Then, $\mathbb{L}$ was exhaustively searched for models with estimated error equal or lesser than that of the found sup-strong local minimums. The global minimums with the least VC dimension among the sup-strong local minimums, stored and found from the exhaustive search of $\mathbb{L}$, were returned as $\hat{\mathcal{M}}$, and hypotheses were learned on $\hat{\mathcal{M}}$ with the independent sample. This algorithm is optimal due to the sup-weak U-curve property that is satisfied in the Partition Lattice Learning Space (cf. Proposition 3.2).
- **Suboptimal**: Algorithm 4 was applied, but without excluding models from $\mathbb{L}(\mathcal{H})$ due to sup-strong local minimums, until the first sup-strong local minimal was found after 100 models have been exhausted. The "global minimums" $\hat{\mathcal{M}}$ were returned as the found sup-strong local minimums with the least estimated error and VC dimension, and hypotheses were learned on $\hat{\mathcal{M}}$ with the independent sample. This algorithm is suboptimal, since there is no guarantee that a global minimum with the least VC dimension was found after exhausting only 100 models.

In both algorithms, when a sup-strong local minimal is found, the search restarts from a model sampled uniformly from $\mathbb{L}$ and $\mathbb{L}(\mathcal{H})$, for the optimal and suboptimal algorithms, respectively. The optimal algorithm is essentially Algorithm 4 with an early stop when $\mathbb{L}$ is within the reach of an exhaustive search, considered here as having lesser than $1,000$ models. The suboptimal algorithm does not prune the partition lattice, and search it for sup-strong local minimums, restarting from a model sampled from $\mathbb{L}(\mathcal{H})$, until it has visited (exhausted) at least 100 models.

For each example and sample size, we present in Table 4.3 some metrics summarizing the results of the 100 simulated samples under the optimal and suboptimal algorithms. The optimal algorithm exhausted in general between 300 to 450 models in each case, although there are cases where it exhausted as few as 50, and as many as $1,600$, models, and had, in general, an execution time 7 times that of the suboptimal algorithm, that exhausted around 100 models in each case by design.

We see, in general, an increase on the estimated error $\hat{L}(\hat{\mathcal{M}})$ of $\hat{\mathcal{M}}$ with the sample size towards the error $L(\mathcal{M}^\star)$, that is 0.23125 in all examples. However, we do not see a great effect of the sample size on the real errors[2] of $\hat{\mathcal{M}}$, of the ERM hypothesis learned directly on $\mathcal{H}$ with the whole sample, and of the hypothesis estimated from $\hat{\mathcal{M}}$ with the independent sample.

An interesting feature of learning via Learning Spaces that can be observed in these simulations is the one depicted in Figure 1.18, that is the existence of scenarios where

[2] In the case $\hat{\mathcal{M}}$ (sup-strong local minimums with the least error and VC dimension), $\hat{h}$ (ERM hypotheses of $\mathcal{H}$ under the whole sample) or $\hat{h}_{\hat{\mathcal{M}}}^{\tilde{\mathcal{D}}_M}$ (hypotheses that minimize the empirical error under the independent sample in $\hat{\mathcal{M}}$) are not unique, we consider as their real error the lesser error among the models/hypotheses returned as them.

learning via Learning Spaces with independent sample, by dividing a sample of size $N + M$ into a training, validation and independent sample, is better than learning directly on $\mathcal{H}$ with the whole sample of size $N + M$ via ERM, that is the classical learning framework of VC theory (cf. Appendix A). We see in Table 4.3 that the median error of the hypothesis learned via Learning Spaces is almost always equal or lesser than the respective median error of the ERM hypothesis of $\mathcal{H}$ under the whole sample.

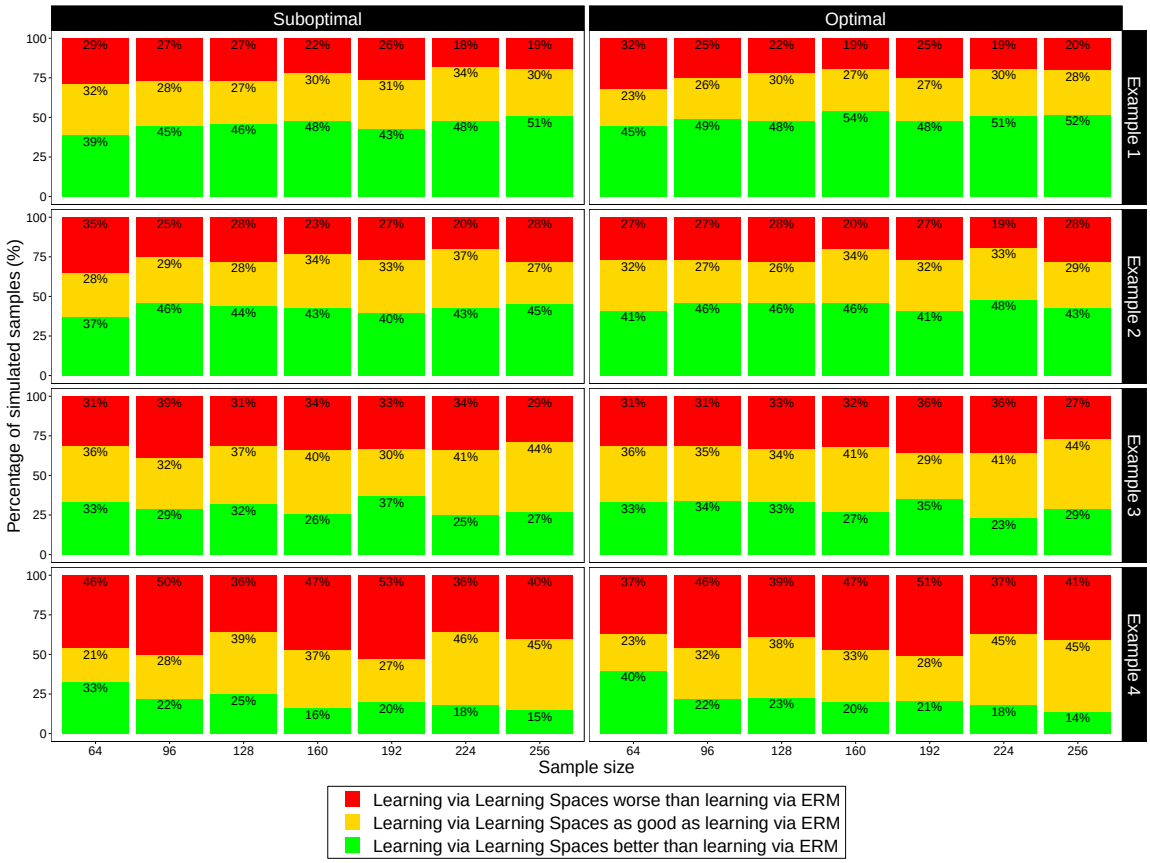

Figure 4.1: *Percentage of the simulations in which the error of the ERM hypothesis in $\mathcal{H}$ were lesser, equal, or greater than the error of the hypothesis learned on $\hat{\mathcal{M}}$, i.e., via Learning Spaces, with the independent sample for each example, sample size and algorithm.*

Further studying this feature, we present in Figure 4.1, for each example, sample size and algorithm, the percentage of the samples where learning via Learning Space was better, worse and as good as learning via ERM on $\mathcal{H}$ with the whole sample, meaning that the real error of the hypothesis learned via Learning Spaces was lesser, greater, and equal, respectively, to the real error of the hypothesis learned via ERM.

In Example 1, learning via Learning Spaces was better in around 39 to 54% of the samples, a number which decreased in Examples 2 to 4, respectively, with 37 to 48%, 23 to 37% and 14 to 40%, while the percentage of samples where it was better to learn via ERM increased from Example 1 to Example 4, respectively, with 18 to 32%, 19 to 35%, 27 to 39% and 36 to 53%. Hence, in all cases of Example 1 to 3 learning via Learning Spaces is better or as good as learning via ERM in at least 61% of the samples, a number which gets as high as 82% in some cases. In Example 4, there were instances in which learning directly

on $\mathcal{H}$ was better in around 53% of the cases, but there are instances in which learning via Learning Spaces is as good or better in around 74% of the cases.

The quality of learning via the Partition Lattice Learning Space relative to learning directly on $\mathcal{H}$ seems to be related to the Conditional Entropy of the joint distribution, so it is better to learn via Learning Space when the Conditional Entropy is smaller, since this is the feature of the joint distributions that increases from Example 1 to 4, as the error of $h^\star$ is constant and $\epsilon^\star$ is the same in Examples 2 and 4. This difference in the quality is quite interesting, since the Conditional Entropy is closely related to the learning complexity [56].

The simulated results hint at the possibility of obtaining better bounds for type IV estimation error than for type II in $\mathcal{H}$ (cf. Figure 1.18) also in a distribution dependent framework based on the Conditional Entropy or an equivalent measure, a topic we leave for future researches.

Ex.	Size	Suboptimal better	Suboptimal as good	Suboptimal worse	Ex.	Size	Suboptimal better	Suboptimal as good	Suboptimal worse
1	64	11	61	28	3	64	10	66	24
1	96	13	72	15	3	96	9	71	20
1	128	6	85	9	3	128	9	76	15
1	160	5	79	16	3	160	7	80	13
1	192	6	79	15	3	192	9	81	10
1	224	9	82	9	3	224	8	83	9
1	256	3	93	4	3	256	3	89	8
2	64	7	76	17	4	64	9	67	24
2	96	8	79	13	4	96	10	71	19
2	128	6	86	8	4	128	11	81	8
2	160	8	76	16	4	160	7	76	17
2	192	8	84	8	4	192	8	83	9
2	224	6	79	15	4	224	6	88	6
2	256	7	88	5	4	256	4	89	7

Table 4.2: *The percentage of simulated samples in which the hypothesis returned by the suboptimal algorithm was better, worse and as good as the hypothesis returned by the optimal algorithm, for each example and sample size. When more than one hypothesis is returned, we consider the real error of the hypothesis with the least error when comparing the algorithms.*

Another interesting feature observed in the simulations is that the solution via the optimal and suboptimal algorithms are as good in the majority of cases. Table 4.2 presents, for each example and sample size, the percentage of samples where the suboptimal algorithm returned a hypothesis which was better, worse and as good as the hypothesis returned by the optimal algorithm. The algorithms were as good in at least 61% of the samples in all cases, a number which increased with the sample size and was as high as 93%, evidencing that suboptimal algorithms may properly work in practice and, since are much more efficient than optimal algorithms based on Algorithm 4, may enhance the possibility of employing Learning Spaces in real learning problems. Indeed, the applications in Sections 4.2 and 4.3 rely on suboptimal algorithms based on Learning Spaces that returned hypotheses which were suitable for the problems at hand.

Although the simulated examples are simple, with only eight points in the domain $\mathcal{X}$, they illustrate some features of the learning via the Partition Lattice Learning Space.

First, we saw that learning via this Learning Space was in general, in more than half of the cases, equal or better than learning via ERM directly on $\mathcal{H}$ with the whole sample, specially when the Conditional Entropy of P is low, evidencing the tighter bounds we have for type IV estimation error when compared to the bounds for type II estimation error of learning directly on $\mathcal{H}$ (cf. Figure 2.3).

The simulations also illustrated that suboptimal algorithms may be as good as the optimal in the Partition Lattice Learning Space. This fact should be due to the existence of a lot of redundancies in $\mathbb{L}(\mathcal{H})$ as a consequence of the lattice structure under the partial order given by $\subset$. Indeed, if $h^\star \in \mathcal{M}$, then $h^\star \in \mathcal{M}_i$ whenever $\mathcal{M} \subset \mathcal{M}_i$, and maybe for some $\mathcal{M}_i$ satisfying $\mathcal{M}_i \subset \mathcal{M}$, so, even if $\hat{\mathcal{M}}$ is not equal to $\mathcal{M}^\star$, it may contain $\mathcal{M}^\star$ or have the same error as it, and hence the hypothesis estimated via ERM with the independent sample may be as good as $h^\star$.

The theoretical results of Chapters 2 and 3, together with these simulations, support that the Partition Lattice Learning Space is a promising tool for learning hypotheses with finite domain, which, at the cost of computational power, may perform better than the classical VC theory learning framework. Furthermore, it can be suitable to consider suboptimal algorithms which allow the instantiation of the method to more complex problems. Suboptimal algorithms together with prior information about the problem at hand, which allows dropping models from $\mathbb{L}(\mathcal{H})$, decreasing its complexity (see Section 4.4 for an example), are the tools to decrease the computational complexity, making learning via the Partition Lattice Learning Space computationally possible for solving real learning problems.

Hence, in this context, the study of how to incorporate prior information into $\mathbb{L}(\mathcal{H})$ in specific problems, and the development of suboptimal algorithms that, for instance, perform a stochastic search of $\mathbb{L}(\mathcal{H})$, following, for example, ideas in [32], are promising topics for future researches that ought to be further investigated.

Example	Algorithm	Size	Exhausted	$\hat{L}(\hat{\mathcal{M}})$	$L(\hat{\mathcal{M}})$	$L(\hat{h})$	$L(\hat{h}_{\hat{\mathcal{M}}}^{\tilde{\mathcal{D}}_M})$	Time (min)
1	Optimal	64	439.5 (175.93,1071.72)	0.125 (0,0.3125)	0.23375 (0.23125,0.35125)	0.23375 (0.23125,0.23875)	0.23125 (0.23125,0.33387)	0.61008 (0.2078,1.06498)
1	Suboptimal	64	101 (100,102)	0.125 (0,0.3125)	0.23375 (0.23125,0.43762)	0.23375 (0.23125,0.23875)	0.23125 (0.23125,0.33625)	0.06536 (0.04964,0.09943)
1	Optimal	96	385.5 (105.65,1422.5)	0.14583 (0.04167,0.29167)	0.23375 (0.23125,0.35006)	0.23375 (0.23125,0.23875)	0.23125 (0.23125,0.30756)	0.44029 (0.1264,1.08991)
1	Suboptimal	96	101 (100,103)	0.16667 (0.04167,0.29167)	0.23375 (0.23125,0.35125)	0.23375 (0.23125,0.23875)	0.23125 (0.23125,0.30875)	0.05922 (0.04646,0.07904)
1	Optimal	128	419.5 (119.65,936.72)	0.15625 (0.0625,0.26641)	0.23375 (0.23125,0.31256)	0.23375 (0.23125,0.23875)	0.23125 (0.23125,0.30625)	0.4709 (0.14567,0.82049)
1	Suboptimal	128	101 (100,103.52)	0.15625 (0.0625,0.28125)	0.23375 (0.23125,0.34137)	0.23375 (0.23125,0.23875)	0.23125 (0.23125,0.30625)	0.06027 (0.04797,0.08416)
1	Optimal	160	412 (79.18,1121.45)	0.175 (0.075,0.26312)	0.23125 (0.23125,0.32556)	0.23375 (0.23125,0.23875)	0.23125 (0.23125,0.30875)	0.46198 (0.101,0.8937)
1	Suboptimal	160	100 (100,103)	0.175 (0.08688,0.26312)	0.23375 (0.23125,0.34375)	0.23375 (0.23125,0.23875)	0.23125 (0.23125,0.30875)	0.05854 (0.04527,0.08511)
1	Optimal	192	367.5 (82.03,808.37)	0.1875 (0.125,0.25)	0.23375 (0.23125,0.32687)	0.23375 (0.23125,0.23875)	0.23125 (0.23125,0.23756)	0.42227 (0.10226,0.75539)
1	Suboptimal	192	101 (100,103)	0.1875 (0.125,0.25)	0.23375 (0.23125,0.34637)	0.23375 (0.23125,0.23875)	0.23125 (0.23125,0.27419)	0.05925 (0.04556,0.07136)
1	Optimal	224	323.5 (98.42,1571.87)	0.17857 (0.09777,0.26786)	0.23375 (0.23125,0.31006)	0.23375 (0.23125,0.23875)	0.23125 (0.23125,0.23756)	0.36373 (0.1182,1.23092)
1	Suboptimal	224	101 (100,103)	0.17857 (0.09777,0.26786)	0.23375 (0.23125,0.23875)	0.23375 (0.23125,0.23875)	0.23125 (0.23125,0.23625)	0.06174 (0.04595,0.08732)
1	Optimal	256	411 (108.97,1314.45)	0.1875 (0.125,0.24258)	0.23375 (0.23125,0.23756)	0.23375 (0.23125,0.23875)	0.23125 (0.23125,0.23625)	0.49402 (0.13248,1.0115)
1	Suboptimal	256	101 (100,102.52)	0.1875 (0.125,0.24258)	0.23375 (0.23125,0.23875)	0.23375 (0.23125,0.23875)	0.23125 (0.23125,0.23756)	0.06207 (0.04963,0.09074)
2	Optimal	64	397 (133.75,1233.4)	0.125 (0,0.3125)	0.23625 (0.23125,0.3565)	0.23625 (0.23125,0.24625)	0.23125 (0.23125,0.33125)	0.54798 (0.19845,1.03148)
2	Suboptimal	64	100 (100,103)	0.125 (0,0.3125)	0.23625 (0.23125,0.3875)	0.23625 (0.23125,0.24625)	0.23125 (0.23125,0.34256)	0.06362 (0.0495,0.08937)
2	Optimal	96	430 (134.4,997.65)	0.16667 (0.04167,0.29167)	0.23625 (0.23125,0.34637)	0.23625 (0.23125,0.24625)	0.23125 (0.23125,0.34019)	0.5112 (0.17869,0.93167)
2	Suboptimal	96	101 (100,102)	0.16667 (0.04167,0.29167)	0.23625 (0.23125,0.35387)	0.23625 (0.23125,0.24625)	0.23125 (0.23125,0.31937)	0.06277 (0.04927,0.07301)
2	Optimal	128	411.5 (132.5,1410.9)	0.15625 (0.0625,0.25)	0.23625 (0.23125,0.32412)	0.23625 (0.23125,0.24625)	0.23125 (0.23125,0.30888)	0.47564 (0.14816,1.25028)
2	Suboptimal	128	100 (100,103)	0.15625 (0.0625,0.25)	0.23625 (0.23125,0.34519)	0.23625 (0.23125,0.24625)	0.23125 (0.23125,0.27537)	0.06135 (0.04916,0.07239)
2	Optimal	160	388 (70.18,1188.62)	0.175 (0.1,0.26312)	0.23625 (0.23125,0.3165)	0.23625 (0.23125,0.24625)	0.23125 (0.23125,0.30888)	0.43426 (0.07852,0.96457)
2	Suboptimal	160	101 (100,104)	0.175 (0.1,0.26312)	0.23625 (0.23125,0.31888)	0.23625 (0.23125,0.24625)	0.23125 (0.23125,0.30625)	0.06064 (0.04694,0.07729)
2	Optimal	192	313.5 (111.9,1448.9)	0.16667 (0.08333,0.27083)	0.23625 (0.23125,0.34756)	0.23625 (0.23125,0.24625)	0.23125 (0.23125,0.24625)	0.38237 (0.13069,1.15636)
2	Suboptimal	192	101 (100,103)	0.17708 (0.08333,0.27083)	0.23625 (0.23125,0.33094)	0.23625 (0.23125,0.24625)	0.23125 (0.23125,0.24625)	0.06134 (0.04889,0.09514)
2	Optimal	224	358 (109.4,1532.12)	0.17857 (0.10714,0.26786)	0.23625 (0.23125,0.24125)	0.23625 (0.23125,0.24625)	0.23125 (0.23125,0.24125)	0.39643 (0.13844,1.12924)
2	Suboptimal	224	101 (100,103)	0.17857 (0.10714,0.26786)	0.23625 (0.23125,0.31388)	0.23625 (0.23125,0.24625)	0.23125 (0.23125,0.24387)	0.06033 (0.04629,0.06858)
2	Optimal	256	415.5 (146.38,1245.82)	0.1875 (0.10938,0.26562)	0.23625 (0.23125,0.28037)	0.23625 (0.23125,0.24625)	0.23125 (0.23125,0.24625)	0.46672 (0.20011,1.05611)
2	Suboptimal	256	101 (100,103)	0.1875 (0.10938,0.26562)	0.23625 (0.23125,0.28037)	0.23625 (0.23125,0.24625)	0.23125 (0.23125,0.24625)	0.06266 (0.05129,0.08921)
3	Optimal	64	411.5 (183,1103.6)	0.125 (0,0.25)	0.23125 (0.23125,0.40656)	0.24625 (0.23125,0.3265)	0.23125 (0.23125,0.33125)	0.57118 (0.23227,0.95668)
3	Suboptimal	64	101 (100,103)	0.125 (0,0.25)	0.24625 (0.23125,0.42912)	0.24625 (0.23125,0.3265)	0.23125 (0.23125,0.34019)	0.06184 (0.04612,0.08108)
3	Optimal	96	398 (137,1170.4)	0.16667 (0.04167,0.31354)	0.24625 (0.23125,0.35675)	0.24625 (0.23125,0.26912)	0.23125 (0.23125,0.33125)	0.47412 (0.1493,1.01176)
3	Suboptimal	96	101 (100,102.52)	0.16667 (0.04167,0.31354)	0.24625 (0.23125,0.35519)	0.24625 (0.23125,0.26912)	0.23125 (0.23125,0.33125)	0.06193 (0.04614,0.07192)
3	Optimal	128	446.5 (104.5,1144.65)	0.1875 (0.0625,0.3125)	0.24625 (0.23125,0.36269)	0.24625 (0.23125,0.26912)	0.23125 (0.23125,0.33125)	0.50952 (0.11727,1.00228)
3	Suboptimal	128	101 (100,102.52)	0.1875 (0.0625,0.3125)	0.24625 (0.23125,0.42662)	0.24625 (0.23125,0.26912)	0.23125 (0.23125,0.33625)	0.06336 (0.05193,0.07428)
3	Optimal	160	372 (98.83,974.82)	0.175 (0.1,0.3)	0.24625 (0.23125,0.34769)	0.24625 (0.23125,0.26125)	0.23125 (0.23125,0.31625)	0.4612 (0.12758,0.84868)
3	Suboptimal	160	101 (100,103)	0.175 (0.1,0.31312)	0.24625 (0.23125,0.35162)	0.24625 (0.23125,0.26125)	0.23125 (0.23125,0.3115)	0.06232 (0.04846,0.08458)
3	Optimal	192	362 (52.4,1399.47)	0.1875 (0.08333,0.27083)	0.24625 (0.23125,0.3415)	0.24625 (0.23125,0.26125)	0.23125 (0.23125,0.3265)	0.40503 (0.06986,1.05268)
3	Suboptimal	192	101 (100,103)	0.1875 (0.08333,0.26094)	0.23875 (0.23125,0.35125)	0.24625 (0.23125,0.26125)	0.23125 (0.23125,0.31887)	0.06058 (0.04856,0.08049)
3	Optimal	224	350 (94.65,1248)	0.19643 (0.125,0.26786)	0.24625 (0.23125,0.32125)	0.24625 (0.23125,0.26125)	0.24625 (0.23125,0.292)	0.40018 (0.12196,0.94924)
3	Suboptimal	224	101 (100,103)	0.19643 (0.125,0.26786)	0.24625 (0.23125,0.32594)	0.24625 (0.23125,0.26125)	0.24625 (0.23125,0.27625)	0.06118 (0.04788,0.07371)
3	Optimal	256	402.5 (111.28,856.07)	0.1875 (0.1168,0.28125)	0.23125 (0.23125,0.33781)	0.24625 (0.23125,0.26125)	0.23125 (0.23125,0.31937)	0.43975 (0.13895,0.82268)
3	Suboptimal	256	101 (100,104)	0.1875 (0.1168,0.28125)	0.23125 (0.23125,0.34412)	0.24625 (0.23125,0.26125)	0.23125 (0.23125,0.30625)	0.06245 (0.04928,0.08368)
4	Optimal	64	425.5 (95.28,937.32)	0.125 (0,0.3125)	0.23625 (0.23125,0.45425)	0.23625 (0.23125,0.33125)	0.23125 (0.23125,0.33125)	0.59825 (0.12546,0.85661)
4	Suboptimal	64	101 (100,103)	0.125 (0,0.3125)	0.28125 (0.23125,0.46375)	0.23625 (0.23125,0.33125)	0.23125 (0.23125,0.33781)	0.06399 (0.05014,0.08263)
4	Optimal	96	390.5 (121.42,792.05)	0.16667 (0.08333,0.29167)	0.23625 (0.23125,0.43625)	0.23625 (0.23125,0.28625)	0.23625 (0.23125,0.33125)	0.49891 (0.1675,0.77862)
4	Suboptimal	96	101 (100,102.52)	0.16667 (0.08333,0.29167)	0.23625 (0.23125,0.47294)	0.23625 (0.23125,0.28625)	0.23625 (0.23125,0.33781)	0.06373 (0.05005,0.07397)
4	Optimal	128	408.5 (116.4,1613.77)	0.1875 (0.0625,0.29766)	0.23375 (0.23125,0.38625)	0.23125 (0.23125,0.28625)	0.23125 (0.23125,0.33125)	0.5301 (0.18958,1.25509)
4	Suboptimal	128	101 (100,103)	0.1875 (0.0625,0.29766)	0.23375 (0.23125,0.34769)	0.23125 (0.23125,0.28625)	0.23125 (0.23125,0.33125)	0.06355 (0.04791,0.08046)
4	Optimal	160	374 (85.65,1077.53)	0.175 (0.1,0.3)	0.23625 (0.23125,0.35125)	0.23125 (0.23125,0.28387)	0.23125 (0.23125,0.33125)	0.43648 (0.12366,0.85135)
4	Suboptimal	160	100 (100,102.52)	0.175 (0.1,0.3)	0.23625 (0.23125,0.38781)	0.23125 (0.23125,0.28387)	0.23125 (0.23125,0.30987)	0.06095 (0.04559,0.08828)
4	Optimal	192	324 (119.33,1290.67)	0.20833 (0.10417,0.27083)	0.23625 (0.23125,0.36962)	0.23125 (0.23125,0.28625)	0.23375 (0.23125,0.33387)	0.39365 (0.13274,1.03296)
4	Suboptimal	192	101 (100,103)	0.20833 (0.10417,0.27083)	0.23625 (0.23125,0.34637)	0.23125 (0.23125,0.28625)	0.23375 (0.23125,0.33125)	0.06104 (0.04598,0.07876)
4	Optimal	224	350 (63.85,925.95)	0.19643 (0.08839,0.28571)	0.23125 (0.23125,0.34769)	0.23125 (0.23125,0.28125)	0.23125 (0.23125,0.28625)	0.38664 (0.08393,0.81463)
4	Suboptimal	224	101 (100,103)	0.19643 (0.08839,0.28571)	0.23125 (0.23125,0.33125)	0.23125 (0.23125,0.28125)	0.23125 (0.23125,0.30987)	0.06105 (0.04562,0.07597)
4	Optimal	256	436 (99.55,1665.75)	0.20312 (0.125,0.28125)	0.23625 (0.23125,0.34375)	0.23125 (0.23125,0.28125)	0.23125 (0.23125,0.33125)	0.48169 (0.13446,1.4123)
4	Suboptimal	256	101 (100,103)	0.20312 (0.125,0.28125)	0.23375 (0.23125,0.34019)	0.23125 (0.23125,0.28125)	0.23125 (0.23125,0.33625)	0.06124 (0.04724,0.07514)

Table 4.3: *Results of the simulations for each example, sample size and type of algorithm (optimal or suboptimal). We present the number of models exhausted; the estimated error $\hat{L}(\hat{\mathcal{M}})$ of $\hat{\mathcal{M}}$; the real error $L(\hat{\mathcal{M}})$ of $\hat{\mathcal{M}}$; the real error $L(\hat{h})$ of $\hat{h}$, the ERM hypothesis in $\mathcal{H}$ of the whole sample (union of training, validation and independent sample); the real error $L(\hat{h}_{\hat{\mathcal{M}}}^{\tilde{\mathcal{D}}_M})$ of the hypothesis estimated from $\hat{\mathcal{M}}$ with the independent sample; and the execution time of the algorithm in minutes. For each quantity, we present the median, and within parentheses the percentiles 2.5% and 97.5%, of the* 100 *samples.*

4.2 Forecasting variable order Markov chains

An important class of classification problems is that of forecasting the future values of a sequence based on past observed data, in which the sample is formed by past values of the sequence and the goal of learning is to find a classifier that, based on the past until a time t, predicts (forecast) the value of the sequence at the time $t + 1$.

In Statistics, there is a whole area, called Time Series Analysis [27, 62], which deals with modeling the dependence between the values in the sequence to infer, from a statistical perspective, about future values. In Machine Learning, models that solve this problem are within the class of sequence models [43], very important to tasks such as speech recognition [40, 115] and financial forecasting [85].

In this section, we illustrate how learning via Learning Spaces may be applied to forecast a binary sequence by predicting the next value based on the last k, for a $k \geq 1$. In order to do this, we consider classifiers under a variable order Markov chain framework [19, 96] learned via a subset of the respective Partition Lattice Learning Space. In the next section, we present the main ideas and definitions of the method, while in Section 4.2.2, we present a possible algorithm to learn it via a Learning Space. In Section 4.2.3, we apply the proposed method to forecast the daily variation (positive or negative) of bitcoin to develop an investment strategy to it.

4.2.1 Main ideas and definitions

We start defining variable order Markov chains from a probabilistic perspective. Consider a sequence $W_1, \dots, W_N$ of random variables taking values in $\{0, 1\}$ with the conditional distribution of each variable given the previous satisfying

$$\begin{aligned} &\mathbb{P}\big(W_t = 1 | W_{t-1} = w_{t-1}, \dots, W_1 = w_1\big) \\ &\quad = \mathbb{P}\left(W_t = 1 | W_{t-1} = w_{t-1}, \dots, W_{t-k} = w_{t-k}\right) \\ &\quad = \mathbb{P}\left(W_{k+1} = 1 | W_k = w_{t-1}, \dots, W_1 = w_{t-k}\right) := p(1 | w_{t-1} \dots w_{t-k}), \end{aligned} \tag{4.1}$$

for all $t = k + 1, \dots, N$ and all $w_j \in \{0, 1\}, j = 1, \dots, N$. A sequence satisfying (4.1) is called a homogeneous order k Markov chain, in which homogeneous means that the conditional distribution of W_t is independent of t (second equality in (4.1)) and being an order k Markov chain means that the conditional distribution of W_t given all past depends actually only on the past from time $t - k$ (first equality in (4.1)).

A special case of order k Markov chains are those in which the order of the dependence of W_t conditional distribution on past values changes depending on the past. As an example, consider the conditional distribution in Table 4.4 (A) of an order 3 Markov chain. Although the conditional distribution of W_t given the past depends solely on the values of W_{t-1}, W_{t-2} and W_{t-3}, the dependence may be actually only on W_{t-1} or on W_{t-1}, W_{t-2}, depending on their values. Indeed, if $W_{t-1} = 0$, then

$$p(1|0 w_2 w_3) = 0.7$$

whatever the values of w_2 and w_3, which means that in this instance the order of the chain

is only 1. Likewise, if $W_{t-1} = 1$ and $W_{t-2} = 0$, then

$$p(1|10w_3) = 0.8$$

whatever the value of w_3, so the order of the chain is only 2. However, if $W_{t-1} = W_{t-2} = 1$ then the value of W_{t-3} is important to determine the conditional distribution of W_t since

$$0.75 = p(1|110) \neq p(1|111) = 0.65$$

so in this instance the order of the Markov chain is 3.

(A)

$w_1 w_2 w_3$	$p(1\|w_1, w_2, w_3)$
000	0.7
001	0.7
010	0.7
011	0.7
100	0.8
101	0.8
110	0.75
111	0.65

(B)

Context A	$p(1\|A)$
0	0.7
10	0.8
110	0.75
111	0.65

(C)

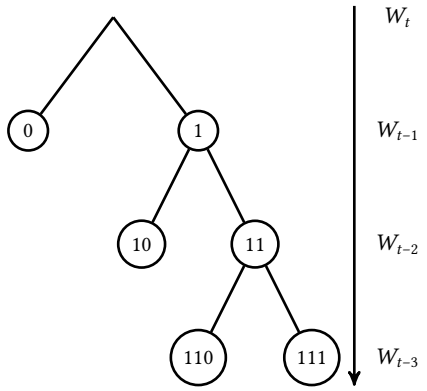

Table 4.4: *(A) Conditional distribution of an order* 3 *Markov chain, which actually represents a variable order Markov chain with contexts and conditional probabilities in (B). (C) Represents the context tree of the variable order Markov chain with conditional distribution (B).*

When a property such as that depicted in Table 4.4 (A) is satisfied, we say the Markov chain has variable order. In this example, if the last value is 0 it is not necessary to go further into the sequence to know the conditional distribution of the next variable; the same is true when the last two values are 10 or when the last three values are 110 or 111.

These sequences, which completely define the conditional distribution, are called the *contexts* of the variable order Markov chain. Since the contexts are better represented via a tree, in which the root is the value of W_t and the leaves are the contexts, variable order Markov chains are defined by their context trees, as that in Table 4.4 (C), and conditional

distribution given the contexts, as in Table 4.4 (B).

The concept of context trees has its roots in compression algorithms, such as [134], and variable order Markov chains have been applied to a variety of problems, ranging from mail spam filtering [28] to human mobility prediction [10, 158]. Since having a data distribution which is that of a variable order Markov chain is common in many applications, one could take into account this prior information and consider as hypotheses space in a problem of forecasting only hypotheses *compatible* with such chains. We now present variable order Markov chains from a forecasting in Machine Learning perspective.

From the sequence $W_1, \dots, W_N$, for a $k < N$ fixed, build the random vectors $(X_{k+1}, Y_{k+1}), \dots, (X_N, Y_N)$ as

$$X_t = (W_{t-1}, \dots, W_{t-k}) \text{ and } Y_t = W_t,$$

for $t = k+1, \dots, N$, that is, Y_t is the value of the sequence at the time t and X_t is the vector of values of the sequence from time $t-k$ to $t-1$. The problem of forecasting the sequence W_t can be in principle solved by learning a classifier in

$$\mathcal{H} = \left\{ h : \{0,1\}^k \mapsto \{0,1\} \right\}$$

with sample $\mathcal{D}_N = \{(X_{k+1}, Y_{k+1}), \dots, (X_N, Y_N)\}$. In order to constraint $\mathcal{H}$ to consider only hypotheses compatible with variable order Markov chains, we consider a subset of the partition lattice of $\mathcal{X} = \{0,1\}^k$, as follows.

Denote $\mathcal{F} = \{\pi : \pi \text{ is a partition of } \mathcal{X}\}$. We say that $\pi = \{a_1, \dots, a_{|\pi|}\} \in \mathcal{F}$ is a context partition if, for every $j = 1, \dots, |\pi|$, there exists a $1 \leq \ell_j \leq k$ and $A_j \in \{0,1\}^{l_j}$ such that

$$a_j = \left\{ x \in \mathcal{X} : (x_1, \dots, x_{l_j}) = A_j \right\}, \tag{4.2}$$

so the blocks of context partitions are formed by vectors which start with contexts $A_1, \dots, A_{|\pi|}$.

Denote $\mathcal{C} = \{\pi \in \mathcal{F} : \pi \text{ is a context partition}\}$ and recall that $\mathcal{H}|_\pi$ are the hypotheses in $\mathcal{H}$ which respect partition π, i.e., classify every point within a same block of π into the same class (cf. Example 1.12). Consider the hypotheses space

$$\mathcal{H}_C := \bigcup_{\pi \in \mathcal{C}} \mathcal{H}|_\pi,$$

containing the hypotheses that respect at least one context partition, and its Learning Space

$$\mathbb{L}(\mathcal{H}_C) := \{\mathcal{H}|_\pi : \pi \in C\}.$$

The set $\mathbb{L}(\mathcal{H}_C)$ is a Learning Space of $\mathcal{H}_C$ since it covers $\mathcal{H}_C$ by definition, and property (ii) of Learning Spaces (cf. Definition 1.10) is inherited from the Partition Lattice Learning Space of $\mathcal{H}$, of which $\mathbb{L}(\mathcal{H}_C)$ is a subset.

The hypotheses in $\mathcal{H}_C$ are those compatible with variable order Markov chains, of order at most k. This means that, if the conditional distribution which generated sequence

W_t represents a variable order Markov chain, then

$$h^\star := \arg\min_{h \in \mathcal{H}} L(h) \cap \mathcal{H}_C \neq \emptyset, \tag{4.3}$$

hence one can learn on $\mathcal{H}_C$ instead of on $\mathcal{H}$ without adding any bias to the learning process. To see that (4.3) is true, observe that, if the contexts of the chain are $A_1, \ldots, A_p$, then $h^\star \cap \mathcal{H}|_\pi \neq \emptyset$ in which π is a context partition, $|\pi| = p$ and each block of π is defined as in (4.2) with sets $A_1, \ldots, A_p$. Furthermore, if π is a context partition, then $A = \{A_1, \ldots, A_{|\pi|}\}$ forms the leaves of a context tree of a variable order Markov chain, so there is a bijection between $\mathcal{C}$ and the space of all context trees of variable order Markov chains of order at most k.

We conclude that, if there is prior information about the data generating process which leads one to believe that it generates samples from a variable order Markov chain, then the forecast of the sequence may be performed by learning via Learning Space $\mathbb{L}(\mathcal{H}_C)$. We highlight that this prior information was inserted into $\mathbb{L}(\mathcal{H})$, the Partition Lattice Learning Space of $\mathcal{H}$, to drop models and obtain $\mathbb{L}(\mathcal{H}_C)$, which is a Learning Space of $\mathcal{H}_C$. Hence, the Partition Lattice Learning Space played the role of a language to express prior information, which implied a constraint in $\mathcal{H}$, generating $\mathcal{H}_C$. In the next section, we discuss a suboptimal algorithm to learn on $\mathbb{L}(\mathcal{H}_C)$.

Remark 4.1. *We remark that the sample $\mathcal{D}_N$ is not formed by independent random vectors, hence the convergence results of Chapter 2 do not immediately apply to this case. Informally, to establish the consistency of learning via Learning Space in this instance, the results of Chapter 2 have to be extended to cases in which the sample points are dependent, but the dependence between (X_t, Y_t) and (X_{t+j}, Y_{t+j}) "decreases" when j increases. This is the case, for example, when the sample comes from an ergodic process, for which the almost sure convergence of type I estimation error to zero when $d_{VC}(\mathcal{H}) < \infty$ has been established in [2]. We leave the study of the consistence of learning via Learning Spaces under ergodic processes as a problem for future researches.*

4.2.2 Suboptimal algorithm

Although $\mathbb{L}(\mathcal{H}_C)$ has much lesser models than the whole Partition Lattice Learning Space of $\mathcal{H}$, no U-curve property is satisfied on it, and it is not possible to carry out an exhaustive search for mild values of k. Nevertheless, we may efficiently find a suboptimal model by imposing suitable stopping criteria beyond finding local minimums. In this section, we present a suboptimal algorithm which we use to solve a practical problem in Section 4.2.3.

The main idea of the algorithm is similar to that of optimal Algorithm 3: start from a model with low VC dimension and go through $\mathbb{L}(\mathcal{H}_C)$ by, at each step, jumping to the neighbor of the current model with the least estimated error. The main differences from Algorithm 3 to the suboptimal algorithm are:

(1) When the neighbors with the least estimated error have the same estimated error as the current model, the algorithm does not stop, but rather sample one of these neighbors to continue the search;

(2) Neighbors with estimated error equal to that of the current model, and lesser VC dimension, are always discarded. Among neighbors with equal estimated error, only those with greater VC dimension are considered at each step;

(3) The algorithm stops when all neighbors have an estimated error greater than that of the current model or when the VC dimension reaches a predetermined value;

(4) When one of the conditions above is met, the algorithm restarts from another initial model;

(5) After the algorithm started on all predetermined initial models, it returns the models among the ones on which it stopped at each run with the least estimated error.

The main characteristic of the estimated error on $\mathbb{L}(\mathcal{H}_C)$ is that there are many ties between neighboring models, so most models are strong local minimums, not because they have lesser estimated errors than their neighbors, but because they all have the same one. This empirical fact was the main guide to establish the properties above. For instance, we consider (1) so that the algorithm does not stop on the initial model, which usually has the same error as all its neighbors. But then, we have to consider (3) for otherwise the algorithm may not stop until it gets to the model with the greatest VC dimension.

Observe that, if we do not stop the algorithm when a strong local minimum is found, but continue the search on models with greater VC dimension, but same estimated error, we lose the implicit regularization of the U-curve algorithm, hence, as we will see in the application in Section 4.2.3, the second condition in (3) will work as the regularization. Finally, we consider (4) since, if we started the algorithm only once, then some paths along the Learning Space would not be reachable as a consequence of (2), which implies that paths going through models with same estimated error, but lower VC dimension, are not possible. We considered (2) to enhance the efficiency and simplicity of the algorithm, avoiding searching loops, and this condition is not strictly necessary. After running the algorithm from all predetermined initial values, it returns the models with the least estimated error found in these runs.

After the maximum order k of the variable order Markov chain, the next parameter of the suboptimal search are the initial models. Fixing a $l \leq k$, we consider l initial models, with context partitions defined as in (4.2) with

$$\{A_1, \dots, A_{2^j}\} = \{0,1\}^j, \qquad j = 1, \dots, l. \tag{4.4}$$

We denote the context partition formed by (4.4) as π_j. These are the models with the hypotheses compatible with variable order Markov chains with order at most j, $j = 1, \dots, l$.

Another parameter of the suboptimal search is the maximum VC dimension d where the algorithm should be stopped when no model with estimated error lesser than that of its neighbors have been found yet, as described in (3). Proceeding in this way, the search is actually on

$$\mathbb{L}_d(\mathcal{H}_C) := \{\mathcal{H}|_\pi : |\pi| \leq d\} \subset \mathbb{L}(\mathcal{H}_C),$$

which implies that the estimated hypotheses will not necessarily be in $\mathcal{H}_C$, but rather in

its subset

$$\mathcal{H}'_C := \bigcup \{\mathcal{H}|_\pi : |\pi| \leq d\}.$$

This is a constraint that may add a bias to the learning process if $h^\star \in \mathcal{H}_C \setminus \mathcal{H}'_C$, but the algorithm may nevertheless return a suitable suboptimal solution.

We assume that $\hat{L}$ is of form (2.8) with only one pair of training and validation sample. Furthermore, we assume that, if a sequence $W_1, \ldots, W_T$ is available, then the training sample, which we denote by $\mathcal{D}_N$, is formed by (X_t, Y_t) with $t = k+1, \ldots, N$,for $N < T$, and the validation sample is formed by (X_t, Y_t) with $t = N+1, \ldots, T$. This means that the evolution of the sequence until time N is used for training, and its evolution from $N+1$ until T for validation.

The framework for learning will be an adaptation of learning by reusing (cf. Figure 1.19), in which the estimator is

$$\hat{h}^{\mathcal{D}_N}_{\hat{\mathcal{M}}} := \arg\min_{h \in \hat{\mathcal{M}}} L_{\mathcal{D}_N}(h),$$

that is the ERM hypotheses on $\hat{\mathcal{M}}$ according to training sample $\mathcal{D}_N$, instead of the respective ERM hypotheses according to the sample given by the union of the training and validation samples, as is the case when learning by reusing.

Recall the definitions of the immediate neighborhoods of a model $\mathcal{M} \in \mathbb{L}(\mathcal{H}_C)$:

$$N(\mathcal{M}) = \{\mathcal{M}_i \in \mathbb{L}(\mathcal{H}_C) : \mathcal{M} \subset \mathcal{M}_i, d(\mathcal{M}, \mathcal{M}_i) = 1\}$$
$$n(\mathcal{M}) = \{\mathcal{M}_i \in \mathbb{L}(\mathcal{H}_C) : \mathcal{M}_i \subset \mathcal{M}, d(\mathcal{M}, \mathcal{M}_i) = 1\}.$$

Recall that π_j is the context partition formed by (4.4). The generic suboptimal algorithm to learn hypotheses compatible with variable order Markov chains in presented in Algorithm 5. The maximum order k of the Markov chain is implicit in the algorithm in the choice of $\mathcal{D}_N$ and $\hat{L}$, and by taking $d \leq 2^k$.

For each initial model, the algorithm determine its upper neighbors with equal error, and its upper and lower neighbors with lesser error. If there are neighbors with lesser error, we sample one of them uniformly and restart the algorithm. If there are no neighbors with lesser error, but there are some upper neighbors with equal error, we sample one of them uniformly and restart the algorithm. If there are no neighbors with lesser error and no upper neighbors with equal error, or if the current model has the maximum VC dimension d, we store the model and restart the algorithm with the next initial model. We repeat this process for all initial models, and return all the models stored. From these models, we may compute $\hat{\mathcal{M}}$ as the models with the least VC dimension among the ones with the least estimated error.

Instead of storing the current model $\mathcal{M}$, we actually store all models $\mathcal{M}_i \in \mathbb{L}(\mathcal{H}_C)$ with minimum VC dimension satisfying $\hat{h}^{\mathcal{D}_N}_{\mathcal{M}} \in \mathcal{M}_i$. Since we do not stop the algorithm at strong local minimums, it may happen that, when the algorithm stops in a model, its estimated error is tied with many models contained in it. Actually, not only the estimated error is the same, but the estimated hypothesis is also the same. Hence, instead of storing the model $\mathcal{M}$, we store the models which contain $\hat{h}^{\mathcal{D}_N}_{\mathcal{M}}$, but have minimum VC dimension

Algorithm 5 Generic suboptimal algorithm to learn models compatible with variable order Markov chains.

Ensure: $l \leftarrow$ max order initial models, $d \leftarrow$ max VC dimension,
$\mathbb{M} \leftarrow \emptyset$
for $j = 1, \ldots, l$ **do**
  $\mathcal{M} \leftarrow \mathcal{H}|_{\pi_j}$
  $Neq \leftarrow \left\{ \mathcal{M}_i \in N(\mathcal{M}) \,:\, \hat{L}(\mathcal{M}_i) = \hat{L}(\mathcal{M}) \right\}$
  $Nle \leftarrow \left\{ \mathcal{M}_i \in N(\mathcal{M}) \cup n(\mathcal{M}) \,:\, \hat{L}(\mathcal{M}_i) < \hat{L}(\mathcal{M}) \right\}$
  while $d_{VC}(\mathcal{M}) < d$ and $|Nle \cup Neq| > 0$ **do**
    if $|Nle| > 0$ **then**
      $\mathcal{M} \leftarrow$ sample from Nle
    else
      $\mathcal{M} \leftarrow$ sample from Neq
    $Neq \leftarrow \left\{ \mathcal{M}_i \in N(\mathcal{M}) \,:\, \hat{L}(\mathcal{M}_i) = \hat{L}(\mathcal{M}) \right\}$
    $Nle \leftarrow \left\{ \mathcal{M}_i \in N(\mathcal{M}) \cup n(\mathcal{M}) \,:\, \hat{L}(\mathcal{M}_i) < \hat{L}(\mathcal{M}) \right\}$
  $\mathcal{M} \leftarrow \text{Arg min} \left\{ d_{VC}(\mathcal{M}_i) \,:\, \mathcal{M}_i \in \mathbb{L}(\mathcal{H}_C), \hat{h}_{\mathcal{M}}^{D_N} \in \mathcal{M}_i \right\}$
  $\mathbb{M} \leftarrow \mathbb{M} \cup \{\mathcal{M}\}$
return $\mathbb{M}$

under this constraint.

Algorithm 5 has a stochastic nature, since neighbors with the same minimal estimated error are sampled as the next model. Hence, the algorithm is clearly suboptimal, not only by its stochasticity, but also because there is no guarantee that $\hat{\mathcal{M}}$ is the minimizer of $\hat{L}$ on $\mathbb{L}(\mathcal{H}_C)$, or that $\hat{h}_{\hat{\mathcal{M}}}^{D_N}$ is close to $h^\star$ with high probability if the sample size is great enough. However, it is more efficient than an exhaustive search of $\mathbb{L}(\mathcal{H}_C)$ and may in some cases return suitable hypotheses for the problem at hand, as we illustrate with the application in the next section.

Remark 4.2. *The proposed algorithm can be greatly improved, both from the point of view of efficiency and of decreasing the bias of learning actually on a subset of $\mathcal{H}_C$. For example, disregarding lower neighbors with same estimated error and considering only initial models related to context partitions of form (4.4), are constraints which simplify the algorithm, but that, if dropped, may increase the quality of the learning. We leave the refinement of Algorithm 5 as a topic for future researches.*

4.2.3 Investment strategy for bitcoin

The bitcoin is a digital currency invented in 2008, which since 2009 has been used as exchange for other currencies and goods [26], and that employs the blockchain technology [104] to manage and record transactions. The price of bitcoin in US Dollars is extremely volatile and has, for example, increased around 486 times from its close value in April 2013 to its peak so far in November 2021. Hence, besides its function as a currency, the bitcoin has also been considered as an investment [70, 138], and has been subject of

speculation.

In this section, we apply Algorithm 5 to develop an investment strategy for bitcoin based on the prediction of the days on which it will have a positive or a negative variation, from which one can decide the days he should stay on the market, leave it by selling all his bitcoin, and come back to it by buying bitcoin again.

We consider as the training sample the daily bitcoin close value from April 30th 2013 to December 31st 2020, as the validation sample the daily value from January 1st 2021 to January 31st 2021, and as the test sample the daily value from February 1st 2021 to April 6th 2022. Figure 4.2 presents the daily bitcoin close value for the period considered in the samples. For each day, we calculate the variation of the bitcoin as the difference between its close and open value in the day, and consider as positive days those in which this variation is great or equal to zero, and as negative days those in which this difference is negative. In the training sample, around 54% of the days are positive. We consider the simple loss function, so the error is the classification error.

From the investor perspective, due to the relative good liquidity of bitcoin, it is optimal to own bitcoin during the positive days, sell at the end of a day before a negative day, and buy back at the beginning of the first following positive day. Hence, if one can properly forecast the positive and negative days based only on the variation of the preceding days, then he can employ such a strategy hoping to get a spread over the current value of bitcoin, having above market gains, i.e., having bigger gains compared to the strategy of staying on the market every day.

We now apply Algorithm 5 to learn a hypothesis that forecast the variation (positive or negative) of the bitcoin based on the variation on the past leading up to the respective day. We consider a maximum order of the chain $k = 30$, the maximum VC dimension as seven different values ($d = 8, 16, 32, 64, 128, 256, 512$) and the number of initial models as $l = \log_2 d$. We estimate seven hypotheses, one for each value of d, to illustrate the regularization role of d in the learning process.

The results of the seven learned hypotheses are in Table 4.5. We first see that, although the validation error decreases with d, what is expected since a greater Learning Space is searched when d increases, the test error is actually better for small values of d, being minimal for $d = 32$. This fact illustrates the regularization role of d, since when it is too big, the learning process might be actually overfitting the validation sample (low validation error), so the error is great on non-observed instances (high test error).

Even though the processing time increases exponentially with d, it is relatively low for small d, which are actually the values of d with best test error. Hence, a suitable solution to the problem at hand may be obtained quite efficiently by taking small values of d. However, even when the test error is minimal, it is quite large (0.453), what is common when forecasting financial series, specially when there is high volatility, since it is not a task that can usually be performed with absolute low error.

Nevertheless, even though the error is absolutely high, the learned model may be suitable for the problem at hand, that is to develop an efficient strategy to invest on bitcoin. Indeed, even with errors around 0.45, there are models with d equal or lesser to 128 that attain a maximum spread in the test sample of more than 100%, and all of them have a

d	Time (min)	Order	$d_{VC}(\hat{\mathcal{M}})$	Error		Spread (%)		
				Validation	Test	Min	Max	Final
8	0.007	4	7	0.516	0.465	0	70.4	55.5
16	0.037	7	13	0.452	0.463	-5.2	67.2	65.2
32	0.340	11	28	0.419	0.453	-5.2	100.5	80.1
64	2.591	8	62	0.323	0.488	-31.7	57.8	41.9
128	27.314	12	117	0.290	0.465	-9.5	112.6	95
256	255.931	11	251	0.194	0.500	-46.7	9.8	5.4
512	2550.394	13	487	0.161	0.505	-52.7	-1.3	-14.6

Table 4.5: *Results of the models estimated via Algorithm 5 to forecast the variation of bitcoin. For each model, it is presented the maximum VC dimension d considered, the time in minutes it took to run Algorithm 5, the maximum order of its contexts, the VC dimension of $\hat{\mathcal{M}}$, the classification error on the validation and test sample, and the minimum, maximum and final spread obtained in the test period by applying the strategy based on the learned classifier. The training, validation and test sample sizes of all models are, respectively,* 2774, 31 *and* 430.

spread of at least 41% at the end of the test period. This illustrates that, even if the model has a great absolute test error, and is not an optimal solution, it may still be useful for developing an investment strategy.

In each plot in Figure 4.2 (B), we present the daily balance of two accounts, which started with $1,000$ dollars in bitcoin in the first day of the test sample, and that followed two strategies, respectively, stay on the market all days and stay only on days in which the learned classifier for the respective value of d predicts as positive. We see that the models for $d = 256$ and $d = 512$ perform badly, evidencing that they are indeed overfitting the validation sample, on which their error is 0.19 and 0.16, respectively. The strategy of the model with $d = 64$ struggles at the beginning of the test period, but recovers and ends the period with a spread of almost 42%. The strategy based on the models with $d = 8, 16, 32$ and 128 behave similarly, although the model with $d = 8$ seems to perform better on the short term, while the other models work better at the long term, specially the model with $d = 128$ which has the biggest final spread of 95%.

One of the qualities of the method is that the learned hypothesis is completely interpretable and one knows exactly the scenarios which it classifies as positive and negative days. For instance, see the learned hypothesis for $d = 8$ presented in Table 4.6. According to it, if the prior day was negative (0), then one should enter the market, as a positive day is predicted. In all other scenarios one should enter/stay in the market, except when the prior day was positive, but the previous two days followed a positive day, but were both negative (1001); or when the prior three days were positive, but the day before was negative (1110). This completely characterizes the investment strategy, and one knows exactly what the learned hypothesis is doing.

This is an example of an application on which suboptimal learning via Learning Spaces efficiently returns a suitable solution to the problem at hand. Although the method worked fairly well to develop an investment strategy for bitcoin, it may not work for other financial series, since it relies on the assumption that the data is generated by a variable order Markov

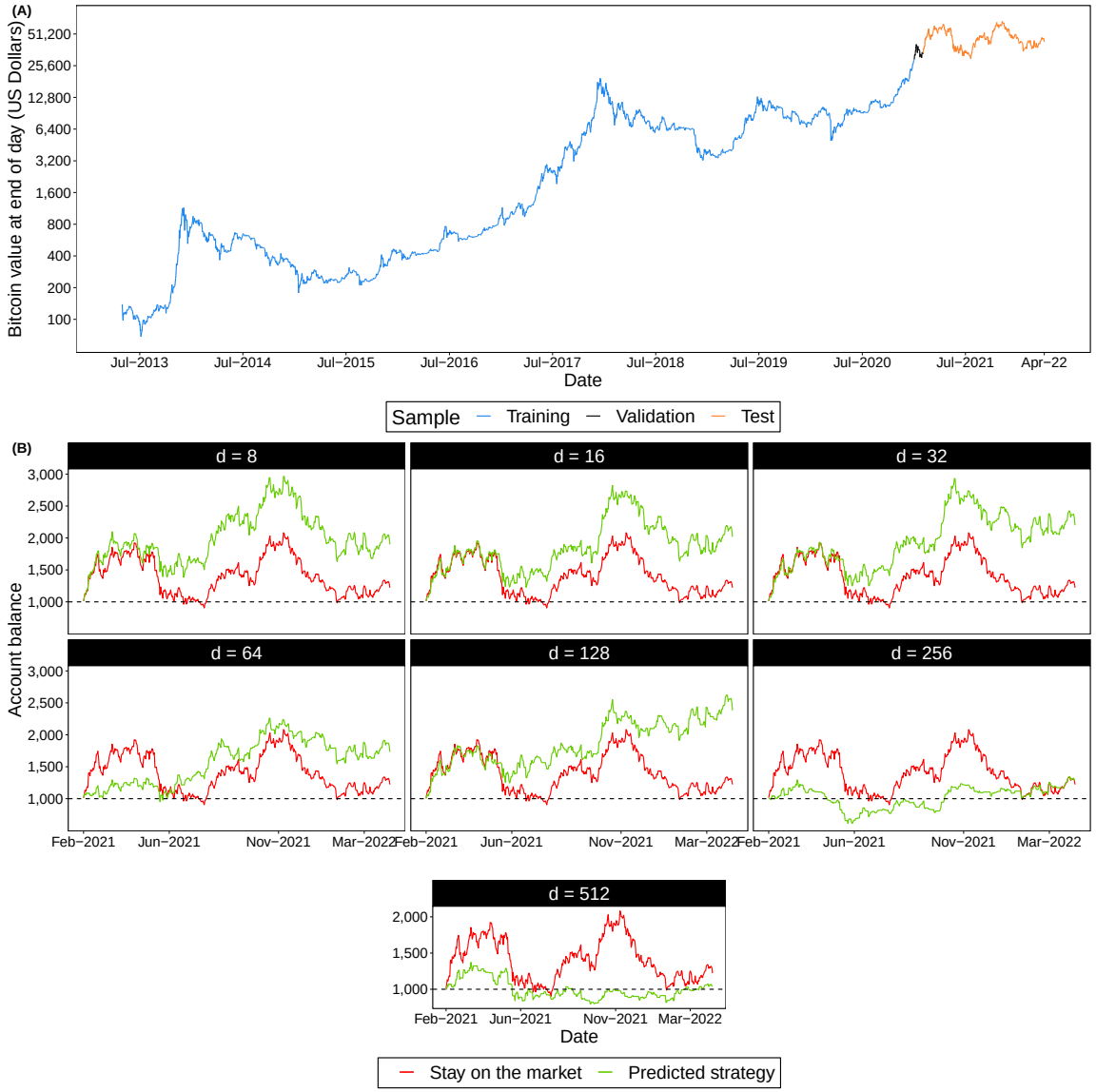

Figure 4.2: *(A) Daily bitcoin value in US Dollars from April 30th 2013 to April 6th 2022, which are the days considered in the training, validation and test samples according to the colors. (B) The balance of two accounts which started with* $1,000$ *US Dollars of bitcoin in February 1st 2021, and which followed, respectively, the strategy of staying on the market every day (red), and staying on the market only on days in which the learned hypothesis, for the respective value of* d*, predicts as positive days (green).*

chain. Moreover, even in this case, if $L(h^\star)$ is too great, then the learned hypothesis may not be useful however it is learned. Nevertheless, it is an interpretable, simple and efficient procedure to forecast binary sequences, and we leave further applications of it as a topic for future researches.

4.3 Multilayer W-operator

A classical problem in Machine Learning is the recognition of handwritten digits, and a canonical benchmark for learning methods is the MNIST data set [89], consisting of

Context A	$\hat{h}^{\mathcal{D}_N}_{\hat{\mathcal{M}}}(A)$
0	1
1000	1
1001	0
101	1
110	1
1110	0
1111	1

Table 4.6: *Estimated hypothesis for $d = 8$.*

70,000 gray-scale images of handwritten digits, which should be recognized as one of the ten digits. In this section, we will adapt Model Selection via Learning Spaces to solve this classical problem. We first present the main ideas of the method in Section 4.3.1, then formally define it in Section 4.3.2, and present its results on MNIST in Section 4.3.3.

4.3.1 Main ideas

A square sized black and white image may be seen as a matrix in the set $\mathcal{X} = \{0,1\}^{p\times p}$, $p \geq 2$, in which the value of each coordinate represents if the respective pixel in the image is white (0) or black (1). In Figure 4.3, we have an example of handwritten digits two, zero, and three.

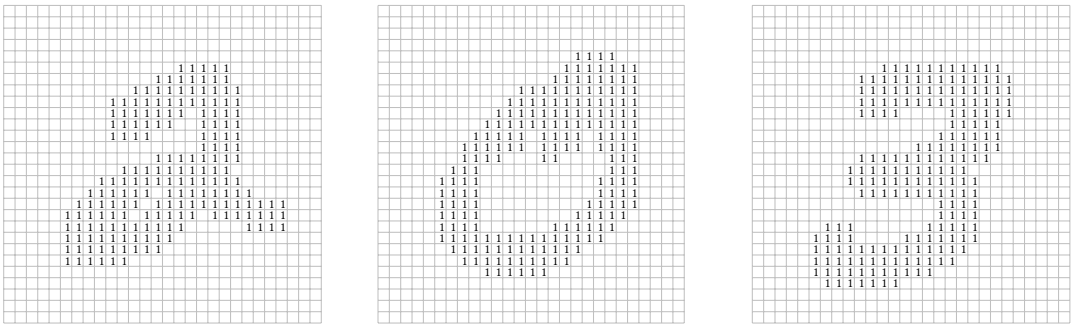

Figure 4.3: *Matrix representation of black and white handwritten digits in the MNIST data set [89]. The gray pixels (value greater than zero) were considered as black (value one). The zero values are omitted for a better visualization.*

The learning problem of digit recognition seeks a classifier which, given an image, returns 1 if a fixed digit is in the image, and 0 otherwise. Formally,

$$\mathcal{H} = \{h : \mathcal{X} \mapsto \{0,1\}\}$$

in which, for example, $h^\star(x)$, $x \in \mathcal{X}$, should equal 1 if digit zero is in x, and 0 otherwise. In this case, considering the simple loss function, we expect that

$$\min_{h\in\mathcal{H}} L(h) = L(h^\star) = 0,$$

since the joint distribution P of the image X and the presence of digit zero Y is deterministic.

This is the case since

$$\mathbb{P}(Y = 1|X = x) = \begin{cases} 1, & \text{if digit zero is in } x \\ 0, & \text{if digit zero is not in } x \end{cases},$$

hence $h^\star(x)$ equals one if this probability is 1, and equals zero otherwise, and, therefore, has zero error.

The main issue with this problem is that $d_{VC}(\mathcal{H}) = 2^{p^2}$, that is the number of possible images of size $p \times p$, so any sample of size lesser than this number will have zero empirical error, so $\hat{h}$ will not be a reliable estimator of $h^\star$. Nevertheless, there is a suitable solution to this issue, that is to actually consider a subset $\mathcal{M} \subset \mathcal{H}$, with $d_{VC}(\mathcal{M}) \ll d_{VC}(\mathcal{H})$, where it is possible to obtain a $\hat{h}_\mathcal{M}$ that well approximates $h^\star_\mathcal{M}$, hoping that $L(h^\star_\mathcal{M})$ is not too greater than zero.

The task of digit recognition has been performed on the MNIST data set with increasing success throughout the years. The first solutions to this problem involved linear classifiers, k-nearest neighbors, non-linear classifiers, support vector machines and shallow neural networks [90]. Since then, the best performing classifiers have been obtained via deep convolutional neural networks [76, 88, 123, 124] which attained a record of 0.23% test error on this data set [35]. We note that some of these methods consider a combination of ten binary classifiers, one for each digit, while others obtain one classifier with output in $\{0, \dots, 9\}$.

All methods applied to solve this problem do not consider the whole hypotheses space $\mathcal{H}$, but rather a restricted subset of it with hypotheses that have some properties expected to be satisfied by $h^\star$. For example, if a same handwritten digit is centered at one image, or a bit dislocated to the right at another, both of them should be classified together since translating a digit does not change its value. Hence, $h^\star$ should be translation invariant, and one could consider only hypotheses in $\mathcal{H}$ with this property.

These restrictions on $\mathcal{H}$ are not made explicitly, but are rather implicit from a parametric representation of the hypotheses, that is, one would consider $\mathcal{M} = \{h_\theta : \theta \in \Theta\} \subset \mathcal{H}$, containing all hypotheses represented as h_θ with parameter $\theta \in \Theta$. This is done, for instance, in DNN in which the hypotheses space is that of the hypotheses representable by a fixed architecture rather than all hypotheses with domain $\mathcal{X}$ and output $\mathcal{Y}$. This can be seen in Example 1.6, where the hypotheses space generated by an architecture $\mathcal{A}$ (cf. (1.8)) may not be the greatest functional hypotheses space for the given problem.

In order to solve the handwritten digit recognition problem via Learning Spaces, we will consider an implicitly determined subset $\mathcal{M}$ of $\mathcal{H}$ via the composition of W-operators [11, 12, 13, 29, 30, 46, 48, 49, 67, 68, 98, 129]. Informally, a W-operator of a binary image is a filter given by relabeling each pixel of the image according to the values of the pixels in a neighborhood of it. This concept is exemplified in Figure 4.4, and is as follows.

A W-operator ψ is determined by a structuring element, or window, W and an operator $f_\psi : \{0,1\}^W \mapsto \{0,1\}$, which classifies each point in $\{0,1\}^W$ in $\{0,1\}$. In this example, W is a subset of $\{0,1\}^{5\times5}$, more specifically, equals zero at all coordinates but at five, which form a symmetric cross in the middle of the 5×5 matrix. The idea behind the W-operator

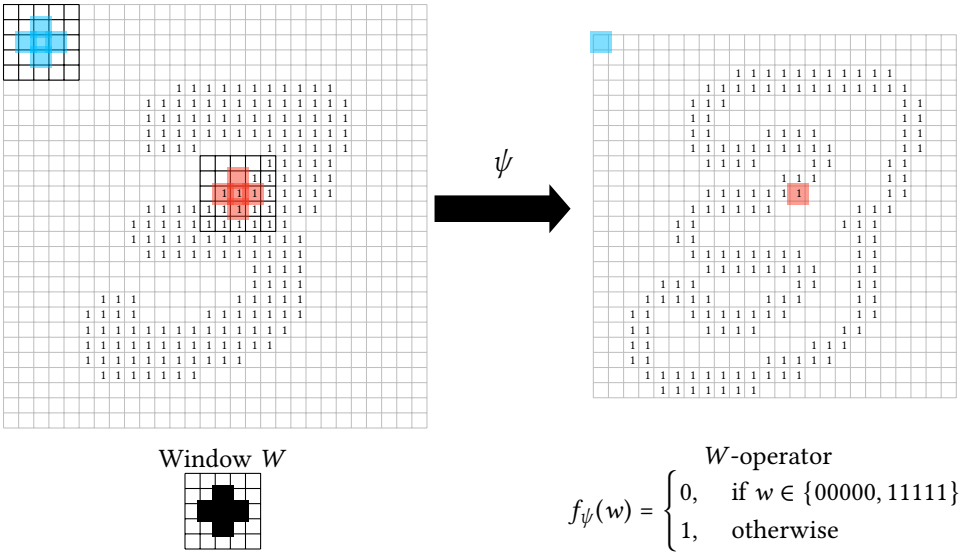

Figure 4.4: *Example of a W-operator filter ψ which recognizes the boundary of a digit. The window W is a subset of $\{0,1\}^{5\times5}$ and the W-operator equals zero if all considered neighbors of a pixel are equal, and one otherwise. The window W is centered at every possible pixel of $x \in \mathcal{X}$, that are all but the ones at the first and last two rows and columns, and the W-operator is calculated for this pixel. Going through every pixel of the image, results in the image on the right-hand side. The zero values are omitted for a better visualization.*

is to center the 5×5 matrix at each pixel in the image x, see the values of its neighbors that are within W, and apply f_ψ to relabel this pixel, from its value to the output of f_ψ.

To exemplify this operation, consider the window W centered at two pixels in Figure 4.4, one at roughly the middle, and the other at the top left of the image. When we center the window at the point in the middle we see that its W-neighborhood, from left to right, top to bottom, is 01111, so when we apply f_ψ to it, we obtain the value 1, which is highlighted in the image at the right-hand side. On the other hand, when we center the window W at the top left we see that its W-neighborhood is 00000, so when we apply f_ψ we obtain the value 0, also highlighted in the image at the right-hand side.

When we center the window at every possible pixel of x and apply f_ψ to its W-neighborhood, that is, when we apply W-operator ψ, we obtain the image at the right-hand side. Observe that this image has four rows and four columns lesser than the original one, since it is not possible to center the window at the points in the first two, and last two, rows and columns. We note that what is centered is the 5×5 matrix of which the one valued coordinates represent the W-neighborhood.

The choice of W and f_ψ was such that the result of ψ is the boundary of the original image. Hence, this W-operator is a filter that recognizes boundaries. In order to obtain other filters, one can change the structuring element W or the function f_ψ. There is a whole area of Mathematics, called Mathematical Morphology, devoted to, among other things, the development of such filters and the deduction of their properties. We refer to [15, 47, 100, 105, 142] for an introduction to Mathematical Morphology and the learning of morphological operators.

The implicit subset of $\mathcal{H}$ we will consider is formed by hypotheses which can be represented as the composition of W-operators. As an example, assume the images have size 29×29, so that $\mathcal{X} = \{0,1\}^{29\times 29}$, and let

$$\theta := \{(W_1, f_{\psi_1}), \cdots, (W_7, f_{\psi_7})\}$$

be a fixed sequence of seven W-operators, in which the structuring elements are not necessarily equal. Then, let

$$h_\theta(x) := (\psi_7 \circ \psi_6 \circ \psi_5 \circ \psi_4 \circ \psi_3 \circ \psi_2 \circ \psi_1)(x) \tag{4.5}$$

be the function given by composing the seven W-operators. Since when applying each W-operator the image loses four rows and four columns, after applying the composition of seven W-operators to a 29×29 image x we obtain a number, that is, $h_\theta(x) \in \{0,1\}$, so

$$\mathcal{M} := \{h_\theta : \theta \in \Theta\},$$

is a subset of $\mathcal{H}$, in which Θ is a collection of sets formed by seven W-operators. We call hypotheses of the form (4.5) multilayer W-operators.

The idea behind considering $\mathcal{M}$ is that, to classify an image, one has to apply some filters to extract relevant features from it. For example, the W-operator in Figure 4.4 extracts an important feature of the image, its boundary, and, once the boundary is known, the interior is redundant information, and hence the image has been filtered off an irrelevant feature. After applying six W-operators, which in principle are designed to remove irrelevant features, the size of the image is now 5×5, so the last W-operator is actually a classifier $\psi_7 : \{0,1\}^{W_7} \mapsto \{0,1\}$, with $W_7 \subset \{0,1\}^{5\times 5}$.

The use of filters to extract relevant features is the essence of image processing with the aim of pattern recognition. Nevertheless, in the discrete case, the sequence of filters is usually hand-tailored by the researcher, or are learned from data in a hybrid context, requiring a lot of prior information about the problem at hand [11, 12, 13, 29, 30, 46, 48, 49, 67, 68, 98, 129], and only the last classifier is purely learned from data. The method proposed here seeks to learn not only the last classifier, but also the filters, in a manner analogous to what is done in the continuous case by convolutional neural networks [160]. In fact, we borrow some ideas from the learning of DNN parameters and adapt them to the discrete case, by developing algorithms which search Learning Spaces, and lattices representing the domain of each W-operator, to learn a sequence of filters and a classifier to learn a fixed handwritten digit.

In the next section, we formalize the method and discuss the estimation algorithm, that is a lattice version of the gradient descent algorithm [137], which can be applied to obtain a suboptimal estimator of $h^\star_{\mathcal{M}}$. In Section 4.3.3, we apply the method to learn the zero digit in the MNIST data set, where we see that the test error of the method, although not as low as the ones obtained via convolutional neural networks, is fairly small, and the method has the advantage of a better understanding of the estimated hypothesis.

4.3.2 Notation and definitions

Let $\mathcal{X} = \{0,1\}^{p\times p}$, $p \geq 2$, be the set of black and white square images of size p, and let (X, Y) be a random vector, defined on $(\Omega, \mathcal{S}, \mathbb{P})$, such that X has support in $\mathcal{X}$ and

$$Y(\omega) = \mathbb{1}\{\text{digit zero is in } X(\omega)\}$$

for $\omega \in \Omega$. Considering $\mathcal{H}$ as all functions from $\mathcal{X}$ to $\{0,1\}$, and the simple loss function, it follows that, independently of the distribution of X,

$$L(h^\star) = 0.$$

Furthermore, since the joint distribution of (X, Y) is deterministic, we have that

$$L_{\mathcal{D}_N}(\hat{h}^{\mathcal{D}_N}) = 0, \tag{4.6}$$

for any possible sample $\mathcal{D}_N$. Hence, to properly estimate $h^\star$ with a relatively small sample, one should consider a subset $\mathcal{M} \subset \mathcal{H}$ with *low* VC dimension satisfying

$$L(h^\star_{\mathcal{M}}) \approx 0,$$

on which (4.6) does not hold. This will be done via multilayer W-operators, as follows.

Fix a $d_W \geq 3$ odd such that $(p-1)/(d_W-1) \in \mathbb{Z}_+$, and let $W \in \{-\lfloor d_W/2\rfloor, \dots, \lfloor d_W/2\rfloor\}^2$ be a structuring element, or window. Actually, W is not any element in this set, but has the constraint of being a connected set. This means that, for every $w, w' \in W$, there exists a sequence $w_0, \dots, w_r \in W$, $r \geq 1$, such that $w_0 = w$, $w_r = w'$ and

$$\|w_i - w_{i+1}\|_\infty = 1,$$

for all $i = 0, \dots, r-1$. Denote

$$\mathscr{C} = \left\{ W \in \{-\lfloor d_W/2\rfloor, \dots, \lfloor d_W/2\rfloor\}^2 \,:\, W \text{ is connected} \right\},$$

so the structuring element is such that $W \in \mathscr{C}$.

For each $W \in \mathscr{C}$, let

$$\mathcal{F}_W = \{f \,:\, \{0,1\}^W \mapsto \{0,1\}\}$$

be the set of all binary functions with domain $\{0,1\}^W$, and define

$$\mathcal{F} = \{(W, f) \,:\, W \in \mathscr{C}, f \in \mathcal{F}_W\}$$

as the collection of W-operators with window in $\mathscr{C}$, which are completely defined by a $W \in \mathscr{C}$ and a $f \in \mathcal{F}_W$. Finally, denote $l = (p-1)/(d_W-1) \in \mathbb{Z}_+$ and let

$$\Theta = \prod_{i=1}^{l} \mathcal{F}$$

be the Cartesian product of l copies of $\mathcal{F}$. Observe that an element θ in Θ is actually a

sequence of l W-operators with window in $\mathscr{C}$, which we denote by

$$\theta = \{(W_1, f_{\psi_1}), \dots, (W_l, f_{\psi_l})\}.$$

Denote each $x \in \{0,1\}^{p' \times p'}$, with $d_W \leq p' \leq p$, by its coordinates as $x = (x_{i,j})_{i,j}$, with $i, j = 1, \dots, p'$, and define

$$\psi(x) := \left(f_\psi \left(x_{i+\frac{d_W-1}{2}, j+\frac{d_W-1}{2}} + W \right) \right)_{i,j},$$

for $i, j = 1, \dots, p' - (d_W - 1)$, in which

$$x_{i+\frac{d_W-1}{2}, j+\frac{d_W-1}{2}} + W = \left\{ x_{i+\frac{d_W-1}{2}+w_1, j+\frac{d_W-1}{2}+w2} \, : \, (w_1, w_2) \in W \right\} \in \{0,1\}^W.$$

We highlight that the dimension of the W-operator domain is $p' \times p'$, while the dimension of its output is $p' - (d_W - 1) \times p' - (d_W - 1)$.

For every $\theta \in \Theta$, define

$$h_\theta(x) := (\psi_l \circ \cdots \circ \psi_1)(x). \tag{4.7}$$

We call h_θ a multilayer W-operator. Since after applying each W-operator the dimension of the output reduces by $d_W - 1$, and since $p - 1 = l(d_W - 1)$, after applying l W-operators in sequence, the dimension of the output is $p - l(d_W - 1) = 1$, hence h_θ has output in $\{0,1\}$. Therefore,

$$\mathcal{M} := \{h_\theta \, : \, \theta \in \Theta\} \subset \mathcal{H},$$

i.e., the set of functions represented by multilayer W-operators of form (4.7) is a subset of all functions from $\{0,1\}^{p \times p}$ to $\{0,1\}$.

In order to solve the handwritten digit recognition problem, we will consider the hypotheses space generated by multilayer W-operators (4.7). The first step, once the hypotheses space is fixed, is to choose a Learning Space of it. For each $\boldsymbol{W} = \{W_1, \dots, W_l\} \in \mathscr{C}^l$ let

$$\Theta_{\boldsymbol{W}} := \left\{ \{(W_1, f_{\psi_1}), \dots, (W_l, f_{\psi_l})\} \, : \, f_{\psi_i} \in \mathcal{F}_{W_i}, i = 1, \dots, l \right\}$$

be all sequences of W-operators with windows $\boldsymbol{W}$, and denote

$$\mathcal{M}_{\boldsymbol{W}} := \{h_\theta \, : \, \theta \in \Theta_{\boldsymbol{W}}\}$$

as the multilayer W-operators with windows $\boldsymbol{W}$. We consider the following Learning Space of $\mathcal{M}$:

$$\mathbb{L}(\mathcal{M}) = \left\{ \mathcal{M}_{\boldsymbol{W}} \, : \, \boldsymbol{W} \in \mathscr{C}^l \right\} \tag{4.8}$$

whose models are the multilayer W-operators with each possible sequence of windows $\boldsymbol{W}$.

The first issue in this framework is to minimize the empirical error L_{D_N} in a fixed $\mathcal{M}_{\boldsymbol{W}}$ since, in principle, it is necessary to calculate this error for every classifier in it, what is not possible. For instance, if $|W_i| = d_W$ for all $W_i \in \boldsymbol{W}$, then $|\mathcal{M}_{\boldsymbol{W}}| = \left(2^{d_W}\right)^l$, which is an enormous number even for small values of l and d_W, such as seven and five, respectively, when it is of order 10^{10}. Observe that, since the maximum size of each window is $(d_W)^2$, a

window with d_W points is really small, so even in simple cases the number of classifiers is too great.

Instead of minimizing the empirical error, we propose a suboptimal algorithm which searches a lattice for a *locally good* hypothesis according to sample $\mathcal{D}_N$, which, although is not an ERM hypothesis, may be suitable for the application at hand.

Fix a window $W \in \mathscr{C}$, denote $\mathscr{L}_W = \{0,1\}^{\{0,1\}^W}$, and observe there is a bijection between $\mathcal{F}_W$ and $\mathscr{L}_W$. Let $(\mathscr{L}_W, \leq)$ be a Boolean lattice, in which

$$w \leq w' \iff w_i \leq w'_i,\ \forall i = 1, \dots, 2^{|W|}.$$

From the bijection above, it follows that $\mathcal{F}_W$ is isomorphic to a Boolean lattice.

Now, in the same manner, fix $\boldsymbol{W} \in \mathscr{C}^l$, denote $\mathscr{L}_{\boldsymbol{W}} = \prod_{i=1}^{l} \{0,1\}^{W_i}$, and mind the bijection between $\mathscr{L}_{\boldsymbol{W}}$ and $\mathcal{M}_{\boldsymbol{W}}$. Let $(\mathscr{L}_{\boldsymbol{W}}, \leq)$ be a Boolean lattice, in which

$$\boldsymbol{w} \leq \boldsymbol{w}' \iff w_i \leq w'_i\ \forall i = 1, \dots, l,$$

i.e., a $\boldsymbol{w}$ in the Cartesian product is lesser or equal to a $\boldsymbol{w}'$ if, and only if, each of its elements is lesser or equal to the corresponding element of $\boldsymbol{w}'$ in the respective Boolean lattice. To easy notation, we use the same symbol $\leq$ to mean the partial order in $\mathscr{L}_W$ and $\mathscr{L}_{\boldsymbol{W}}$, for any $W \in \mathscr{C}$ and $\boldsymbol{W} \in \mathscr{C}^l$, and which order we mean is clear from the context. By bijection, consider the Boolean lattice $(\mathcal{M}_{\boldsymbol{W}}, \leq)$.

We define the strong local minimums of $(\mathcal{M}_{\boldsymbol{W}}, \leq)$, a concept analogous to that of Definition 1.17, but which considers the in-sample error L_{D_N}, and a lattice of hypotheses, instead of a Learning Space, that is a collection of models. In the next definition, d means the distance in the acyclic directed graph $(\mathcal{M}_{\boldsymbol{W}}, \leq)$ (cf. Appendix B).

Definition 4.3. *A hypothesis h is a strong local minimum of $\mathcal{M}_{\boldsymbol{W}}$ if*

$$L_{D_N}(h) \leq \min \left\{ L_{D_N}(h') \,:\, h' \in \mathcal{M}_{\boldsymbol{W}}, d(h,h') = 1 \right\}.$$

The idea is to approximate $h^\star_{\mathcal{M}_{\boldsymbol{W}}}$ by a $\hat{h}^{\mathbb{A}}_{\mathcal{M}_{\boldsymbol{W}}}$, in which the algorithm $\mathbb{A}$ performs a greedy search of $(\mathcal{M}_{\boldsymbol{W}}, \leq)$, that at each step looks for the neighbor of a given hypothesis with the least empirical error. If such neighbor has an error lesser than the current hypothesis, the algorithm restarts from it. Otherwise, if the hypothesis is a strong local minimum, the algorithm stops and returns it. This algorithm is analogous to the U-curve algorithm (cf. Algorithm 3), with the difference that it stops when it finds a strong local minimum of $\mathcal{M}_{\boldsymbol{W}}$.

This algorithm is much more simple than an exhaustive search of $\mathcal{M}_{\boldsymbol{W}}$, since each hypothesis has *only* $\sum_{i=1}^{l} 2^{|W_i|}$ neighbors[3]. Although there is no guarantee that the found strong local minimum is an ERM hypothesis, it is a *locally good* hypothesis, since by changing the output of one point of any of the W-operators, the empirical error does not

[3] Each neighbor is obtained by changing the output of a point in the domain of one of the l W-operators. Since each operator has $2^{|W_i|}$ points in the domain, each model has $\sum_{i=1}^{l} 2^{|W_i|}$ neighbors.

decrease. This is a suboptimal way to solve an impossible problem which can be suitable for the application at hand, as we will see when we apply this method to MNIST.

This algorithm is analogous to the gradient descent algorithm to minimize differentiable losses with parameters in $\mathbb{R}^d$ [137]. Indeed, gradient descent searches the hypotheses space in a greedy way by, at each step, going in the direction in the parametric space which has the greatest decrease in error, which, by an elementary deduction, is the direction opposed to its gradient. The algorithm continues this search until it is confident that it found a local minimum, that, in general, is not a global minimum, but which is a suitable hypothesis for the problem at hand [3].

In our instance, the parameters of the hypotheses, which are the outputs of each W-operator, are actually in a lattice, and the idea of, at each step, going in the direction that minimizes the loss until a strong local minimum is found, is analogous to that of the gradient descent algorithm. Hence, we call it the lattice gradient descent algorithm.

The lattice gradient descent needs an auxiliary algorithm which determines if a given hypothesis is a strong local minimum of $(\mathcal{M}_W, \leq)$. The *LatticeMinimumExhausted* algorithm is presented in Algorithm 6, and returns TRUE if h is a strong local minimum, and FALSE otherwise. The lattice gradient descent is presented in Algorithm 7, and returns a hypothesis $\hat{h}^{\mathbb{A}}_{\mathcal{M}_W}$ that is the first strong local minimum it finds after starting the algorithm at a $h \in \mathcal{M}_W$.

Algorithm 6 *LatticeMinimumExhausted* auxiliary algorithm.

Input: $h, L_{D_N}(h)$

1: **for** $h' \in \mathcal{M}_W \,:\, d(h', h) = 1$ **do**
2: **if** $L_{D_N}(h') < L_{D_N}(h)$ **then**
3: **return** FALSE
4: **return** TRUE

Algorithm 7 Lattice gradient descent algorithm for learning on $\mathcal{M}_W$.

Ensure: $h \in \mathcal{M}_W$, Cost $\leftarrow L_{D_N}(h)$
1: **while** $!MinimumExhausted(h, Cost)$ **do**
2: $h \leftarrow h'$ s.t. $L_{D_N}(h') = \min\{L_{D_N}(g) \,:\, g \in \mathcal{M}_W, d(g, h) = 1\}$
3: Cost $\leftarrow L_{D_N}(h')$
4: $\hat{h}^{\mathbb{A}}_{\mathcal{M}_W} \leftarrow h$
5: **return** $\hat{h}^{\mathbb{A}}_{\mathcal{M}_W}$

In this framework, the error of each model in $\mathbb{L}(\mathcal{M})$ will be defined as

$$\hat{L}(\mathcal{M}_W) := \hat{L}(\hat{h}^{\mathbb{A}}_{\mathcal{M}_W}),$$

in which the error in the right-hand side is the empirical error of $\hat{h}^{\mathbb{A}}_{\mathcal{M}_W}$ under a validation sample, independent of $\boldsymbol{D}_N$. To easy notation, we assume that the available sample has

already been split into $\mathcal{D}_N$ and a validation sample, where the former is used to calculate $\hat{h}^{\mathbb{A}}_{\mathcal{M}_W}$, and the latter to estimate its error.

The solution of Model Selection via Learning Space in this instance would be

$$\hat{\mathcal{M}} := \underset{\mathcal{M}_W \in \mathbb{L}(\mathcal{M})}{\arg\min}\, \hat{L}(\mathcal{M}_W). \tag{4.9}$$

We note that, not only the number of models in $\mathbb{L}(\mathcal{M})$ is enormous, but the computational time to calculate $\hat{h}^{\mathbb{A}}_{\mathcal{M}_W}$ is also meaningful, so a search of $\mathcal{M}_W$ to return an optimal solution of (4.9) is impossible, even if a U-curve property was satisfied.

In order to circumvent this issue, we apply the U-curve algorithm in Algorithm 3 until we find the first strong local minimum, and return it. We denote this first strong local minimum by $\tilde{\mathcal{M}}$. Although there is no guarantee that $\tilde{\mathcal{M}}$ is a global minimum, it may have a small error and may be suitable for the problem at hand.

In summary, the learned hypothesis of $\mathcal{M}$ will be $\hat{h}^{\mathbb{A}}_{\tilde{\mathcal{M}}}$, which is suboptimal when compared, for example, with $\hat{h}^{\mathcal{D}_N}_{\hat{\mathcal{M}}}$ obtained when learning by reusing (cf. (1.21)). Nevertheless, as we see when applying the method to MNIST in the next section, it is not possible to calculate $\hat{h}^{\mathcal{D}_N}_{\hat{\mathcal{M}}}$ within a reasonable time, and $\hat{h}^{\mathbb{A}}_{\tilde{\mathcal{M}}}$ has actually a small test error, so the method is useful to solve the practical problem of digit recognition.

Remark 4.4. *Due to redundancies, the collection $\mathbb{L}(\mathcal{M})$ defined in (4.8) may not satisfy the property (ii) of Learning Spaces (cf. Definition 1.10), since changing one point in a window may not change the model generated by h_θ. The main issue in this instance would be the waste of resources to calculate the error of a same model twice, hence is not a big problem from a theoretical perspective. We leave for future researches a further study of the algebraic aspect of the Learning Space in this instance to assert if it indeed does not satisfy (ii) and the implications of it.*

4.3.3 MNIST results

The MNIST data set is composed of 70,000 handwritten digits, divided in a training (60,0000) and test (10,000) sample. The training images were further divided into two samples: 45,000 were used as sample $\mathcal{D}_N$, and 15,000 were used for validation, to calculate the error of each model as the validation error of the hypothesis returned by Algorithm 7 after processing the sample $\mathcal{D}_N$. We considered the image as the input, and the presence of digit zero in it as the output of the classifier, so the objective of learning is to predict the zero digit. Algorithm 7 ran for roughly six months and, in this period, exhausted only 18 models in (4.8), without finding any strong local minimum.

When the algorithm was stopped, it had last exhausted a model with validation error of 0.02126. The hypothesis estimated in this model was applied to the test sample and generated the confusion matrix in Table 4.7. We see a greater percentage of false negatives (121 out of 980, 12.3%) than of false positives (98 out of 9020, 1.08%), what in the whole test sample implies a test error of 0.0219.

Figure 4.5 presents the seven windows of the model on which the algorithm was stopped. All windows have six points, except for the first and last, which have nine and

Observed	Predicted	
	Not zero	Zero
Not zero	8922	98
Zero	121	859

Table 4.7: *Confusion matrix of the multilayer W-operator learned to predict the zero digit in the MNIST data set. The test error is* 0.0219.

seven points, respectively. The seven layers suffered modifications from their initial value, which was the centered five point cross.

The learning of a multilayer W-operator in this instance has three features. First, although not state-of-the-art, a test error of order 2% for a first implementation of the method is not at all bad, since it has room for improvement from both a theoretical and algorithmic perspective. Second, an ordinary implementation of Algorithm 7 is not suitable for application, since it took six months to attain an error of 2% in a numerical server, so it is imperative to develop more efficient algorithms, which may take advantage of Graphical Processing Units (GPU). Finally, we see in Figure 4.5 that we have the whole specification of the classifier, given by the windows in it and the operators associated to them, which permit to better understand the classification process given by filtering the input image.

We aimed with this example to explore the possibility of developing a discrete object analogous to neural networks in a continuous setting, by making use of Learning Spaces and Mathematical Morphology techniques. This example illustrated that the method may work, since fairly low test errors were obtained when solving a real problem of interest, but there is much work to be done to transform it into a practical method.

4.4 Interval Boolean functions

A more general problem than that of digit recognition is that of image classification, which seeks to classify images according to some feature present in them. Many important applications, such as handwritten digit recognition (cf. Section 4.3), fingerprint recognition [4] and face recognition [84], are special cases of image classification. In this section, we discuss how one can apply Model Selection via Learning Spaces to solve this problem, by restricting the Partition Lattice Learning Space to consider only interval Boolean functions [34, 86].

4.4.1 Main ideas

As discussed in Section 4.3, a square sized black and white image may be seen as a matrix in the set $\mathcal{X} = \{0,1\}^{p\times p}$, $p \geq 2$, in which the value of each coordinate represents if the respective pixel in the image is white (0) or black (1) (see Figure 4.3 for an example). The problem of image classification seeks a classifier which, given an image, returns one if this image has some feature of interest, and zero otherwise, so the hypotheses space in this case is

$$\mathcal{H} = \{h : \mathcal{X} \mapsto \{0,1\}\}$$

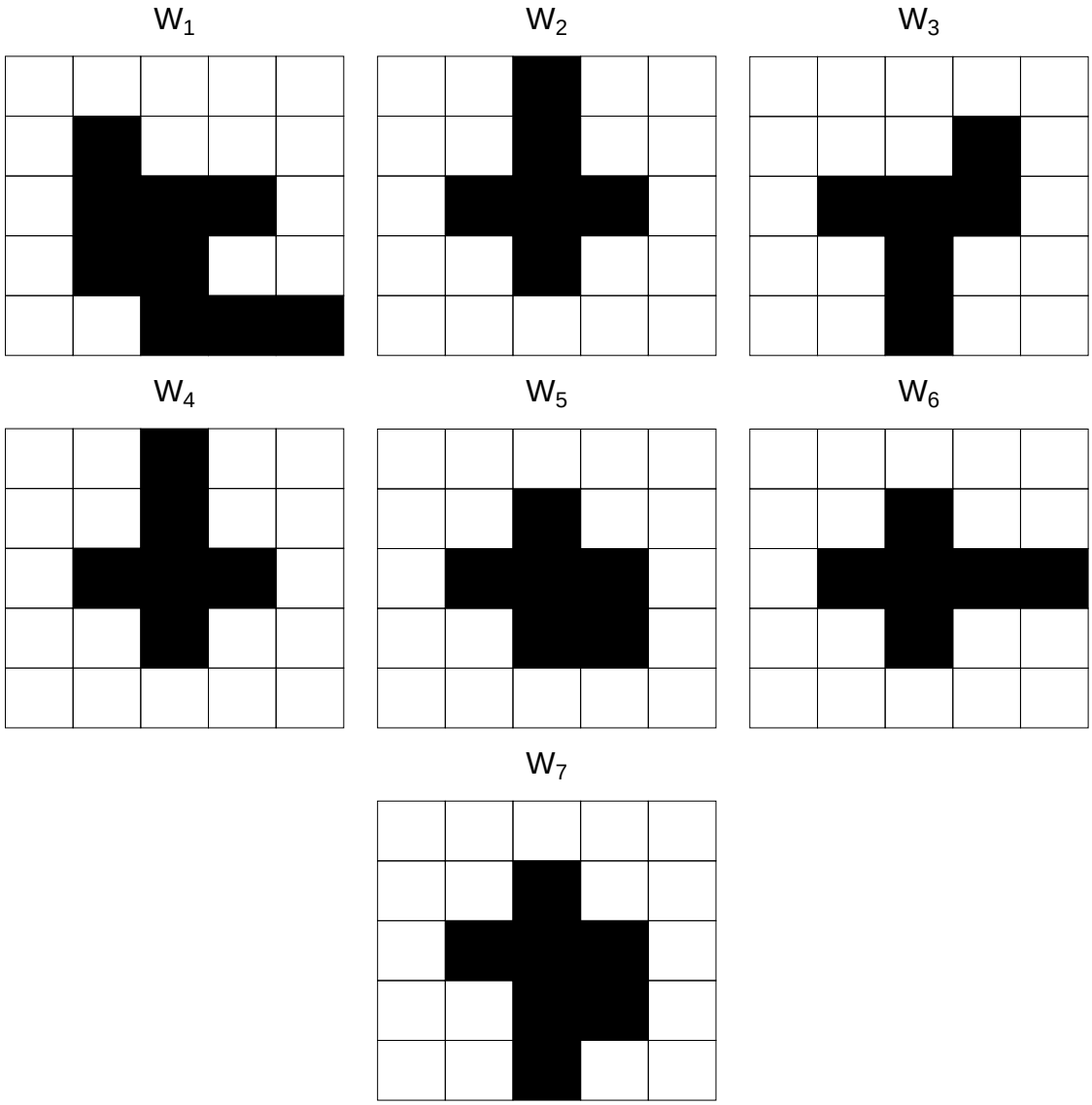

Figure 4.5: *The windows of the multilayer W-operator estimated to predict the zero digit in the MNIST data set.*

and

$$\{x \in \mathcal{X} \,:\, h^\star(x) = 1\}$$

contains all images which have the feature of interest. When we consider the simple loss function, we expect that

$$\min_{h \in \mathcal{H}} \; L(h) = L(h^\star) = 0,$$

since the presence of the feature is deterministic: it is either in the image, or not.

Common features of interest in this scenario are the presence of some entity (a dog, a tree, a vase, etc...), the presence of a handwritten character or the presence of someone's face or fingerprint, what characterizes the problems of object, handwritten character, face and fingerprint recognition, respectively. Although each of these problems have specific solvers which take advantage of particular properties of the feature being recognized, we

will present a general manner of learning a classifier which can in theory be applied to any of these problems.

Images have a scale property, in the sense that increasing or decreasing a form in it does not change its *nature*. This can be formally defined by considering the intervals of the Boolean lattice $(\mathcal{X}, \preceq)$, as follows. Recall the partial ordering in $\mathcal{X}$ given by

$$x \preceq y \iff x_i \leq y_i \text{ for all } i = 1, \ldots, p^2,$$

and, for $x \preceq y$, define the interval with limits x and y by

$$[x, y] := \{w \in \mathcal{X} : x \preceq w \preceq y\},$$

that are the images greater or equal to x and lesser or equal to y.

For each $h \in \mathcal{H}$ define

$$B_1(h) := \{x \in \mathcal{X} : h(x) = 1\}$$

as the points it classifies in 1. We assume that

$$B_1(h^\star) = \bigcup_j A_j \tag{4.10}$$

in which $A_j = [x_j, y_j]$ are disjoint intervals with at least 2 points, i.e., $|A_j| \geq 2$ and $A_j \cap A_{j'} = \emptyset$ for $j \neq j'$. We say that A_j are disjoint non-degenerate intervals.

This is a reasonable assumption, since if one changes the value of one pixel in the image, its value according to $h^\star$ will not change in the great majority of cases, and actually thousands of pixels can be changed, and the image classification can remain the same. As an example, consider the dogs in Figure 4.6, which are such that $w \in [x, y]$. Observe that any image in the interval $[x, y]$ represents a dog, hence if $h^\star$ recognizes dogs it should equal 1 in any image in this interval.

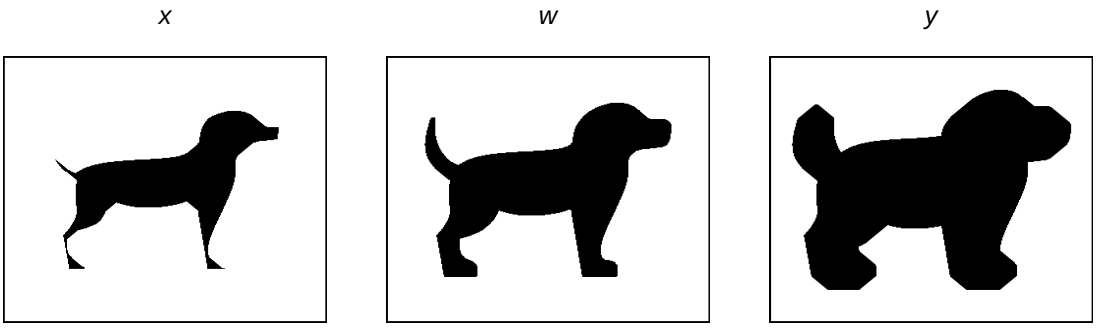

Figure 4.6: *Black and white images of size* 300×300 *of dogs, such that* $x \preceq w \preceq y$. *Image* x *has* $49,044$, *image* w *has* $65,774$ *and image* y *has* $75,427$ *black pixels. Hence, there are* $2^{26,383}$ *images in* $[x, y]$.

Assuming that $h^\star$ satisfies (4.10), one can consider a subset of $\mathcal{H}$ of candidate hypotheses which contains all hypotheses satisfying (4.10). Functions such that $B_1(h)$ can

be partitioned into non-degenerate intervals are examples of interval Boolean functions, which we formally define in Definition 4.7 in the next section. Hence, instead of $\mathcal{H}$, one could learn on

$$\mathcal{H}_2 = \{h \in \mathcal{H} \, : \, h \text{ is an interval Boolean function}\},$$

which is such that $h^\star \in \mathcal{H}_2$ by assumption (4.10).

Actually, one can go further and assume that the sets A_j in (4.10) have size at least k, i.e., $|A_j| \geq k$, with $k > 2$, so one can consider only the size k interval Boolean functions. Among these functions, there are the ones which $B_1(h)$ can be partitioned into intervals of size at least k. Therefore, one could learn on

$$\mathcal{H}_k = \{h \in \mathcal{H} \, : \, h \text{ is a size k interval Boolean function}\},$$

and it is still reasonable that $h^\star \in \mathcal{H}_k$ even for great values of k. For example, the interval $[x, y]$ represented in Figure 4.6 has $2^{26,383}$ images.

There are great advantages of learning on $\mathcal{H}_k$. First, one can consider a subset of the Partition Lattice Learning Space $\mathbb{L}(\mathcal{H})$ of $\mathcal{H}$ containing only the models whose partition is formed by disjoint unions of size k intervals, except for one block, as the Learning Space $\mathbb{L}(\mathcal{H}_k)$. Second, the VC dimension of $\mathbb{L}(\mathcal{H}_k)$ is bounded above by $\lfloor 2^{p^2}/k \rfloor + 1$ (cf. Proposition 4.8), and hence it is more efficient to learn via Learning Spaces on $\mathbb{L}(\mathcal{H}_k)$ than on $\mathbb{L}(\mathcal{H})$. Finally, the sup-weak U-curve property is also satisfied on $\mathbb{L}(\mathcal{H}_k)$ (cf. Proposition 4.10), so the learning on it is non-exhaustive.

This is a nice example of how prior information about the problem at hand may be incorporated into the Learning Space to have a more efficient learning, from the point of view of estimation errors and of computational complexity. Nevertheless, more prior information could be incorporated to enhance even more the efficiency of learning.

For instance, images are also translation, and in some cases rotation, invariant, since, for example, rotating or translating the dogs in Figure 4.6 do not change their nature: they are still dogs. On the one hand, these properties could be incorporated implicitly by considering only the hypotheses in $\mathcal{H}_k$ which are rotation and translation invariant. On the other hand, one could consider only interval Boolean functions in each layer when learning W-operators (cf. Section 4.3), adding the interval restriction to a model which considers the translation invariance.

As discussed in Section 3.6, implementing an efficient optimal search on the Partition Lattice Learning Space and its subsets is a challenge when $|\mathcal{X}|$ is not very small, which is outside the scope of this thesis. Hence, we leave the application of the learning of interval Boolean functions via Learning Spaces as a topic for future researches.

In Section 4.4.2, we formally define the interval Boolean functions and the Learning Space $\mathbb{L}(\mathcal{H}_k)$, and in Section 4.4.3 we show that a U-curve property is satisfied on $\mathbb{L}(\mathcal{H}_k)$, establishing the second example of this thesis where a U-curve property holds.

4.4.2 Notation and definitions

Let $\mathcal{X} = \{0,1\}^{p\times p}$, $p \geq 1$, be the space of black and white square images of size p and $\mathcal{H} = \{h : \mathcal{X} \mapsto \{0,1\}\}$ be the set of all Boolean functions of $p \times p$ variables. We denote the elements of $\mathcal{X}$ by $x := (x_i)_i$ and $y := (y_i)_i$, and consider the following partial order on $\mathcal{X}$:

$$x \leq y \iff x_i \leq y_i, \text{ for all } i = 1, \dots, p^2.$$

In $(\mathcal{X}, \leq)$, an image x is lesser or equal to an image y if the values of its pixels are lesser or equal to the values of the respective pixels in the image y. We define the intervals of $(\mathcal{X}, \leq)$.

Definition 4.5. *Let $x, y \in \mathcal{X}$ be such that $x \leq y$. The interval with limits x and y is defined as*

$$[x, y] := \{w \in \mathcal{X} : x \leq w \leq y\}.$$

We denote the number of elements in an interval by

$$S([x, y]) := |[x, y]|.$$

If an interval has at least two elements, it is said to be non-degenerate.

Recall the definition of $\mathcal{F} := \{\pi : \pi \text{ is a partition of } \mathcal{X}\}$, the Partition Lattice of $\mathcal{X}$ (cf. Example 1.12), and denote the partitions in $\mathcal{F}$ by $\pi := \{a_1, \dots, a_{|\pi|}\}$. Consider the set valued function $B : \mathcal{H} \mapsto \mathcal{P}(\mathcal{F})$ given by

$$B(h) := \{\pi \in \mathcal{F} : h \in \mathcal{H}|_\pi\},$$

recalling that $\mathcal{H}|_\pi$ are the hypotheses in $\mathcal{H}$ which respect partition π, that is, classify every point in a block of π in the same category. Observe that $B(h)$ contains all partitions respected by h.

Among the partitions in $\mathcal{F}$, there are those formed by $|\pi| - 1$ blocks which are disjoint unions of intervals, and a block $a_{|\pi|}$, which may not satisfy this property. These are the interval partitions of $\mathcal{X}$.

Definition 4.6. *Fix a $k \in \mathbb{Z}_+$, $k > 1$. A partition $\pi \in \mathcal{F}$ is said to be a size k interval partition if, for $i = 1, \dots, |\pi| - 1$ and $l_i \geq 1$,*

$$a_i = \bigcup_{j=1}^{l_i} [x_{ij}, y_{ij}]$$

in which $[x_{ij}, y_{ij}]$ are intervals with $S([x_{ij}, y_{ij}]) \geq k$ and $[x_{ij}, y_{ij}] \cap [x_{ij'}, y_{ij'}] = \emptyset$ if $j \neq j'$. Define

$$\mathcal{F}_k := \{\pi \in \mathcal{F} : \pi \text{ is a size } k \text{ interval partition}\}.$$

We are now in position to define the size k interval Boolean functions. These are functions which respect at least one size k interval partition.

Definition 4.7. *For each $k \in \mathbb{Z}_+$, $k > 1$, the space of the size k interval Boolean functions is*

defined as

$$\mathcal{H}_k := \{h \in \mathcal{H} \,:\, B(h) \cap \mathcal{F}_k \neq \emptyset\} = \bigcup_{\pi \in \mathcal{F}_k} \mathcal{H}|_\pi.$$

If we assume $h^\star$ satisfies (4.10), then $h^\star \in \mathcal{H}_2$, since the two block partition

$$\pi = \left\{ \bigcup_j A_j, \mathcal{X} \setminus \bigcup_j A_j \right\}$$

is in $B(h^\star)$, and is a size 2 interval partition. Analogously, if we assume each A_j is an interval with size at least k, then $h^\star \in \mathcal{H}_k$, hence, if there is prior information stating that $h^\star$ satisfies (4.10) with intervals of size at least k, then one can learn on $\mathcal{H}_k$ without inserting any bias into the learning process, since $h^\star$ is not lost by this constraint.

A possible Learning Space of $\mathcal{H}_k$ is the subset of the Partition Lattice Learning Space of $\mathcal{H}$ given only by the models generated by k interval partitions, that is

$$\mathbb{L}(\mathcal{H}_k) = \{\mathcal{H}|_\pi \,:\, \pi \in \mathcal{F}_k\}\,. \tag{4.11}$$

It is indeed a Learning Space, since *(ii)* in Definition 1.10 is inherited from the Partition Lattice Learning Space of $\mathcal{H}$, and it covers $\mathcal{H}_k$ by definition of the k interval Boolean functions.

The main advantage, from the perspective of estimation errors, of learning on $\mathcal{H}_k$ is that the VC dimension of $\mathbb{L}(\mathcal{H}_k)$ may be much lesser than that of $\mathcal{H}$ if k is great, as stated in the next proposition.

Proposition 4.8. *For every $k \geq 2$ it holds*

$$d_{VC}(\mathbb{L}(\mathcal{H}_k)) \leq \left\lfloor \frac{2^{p^2}}{k} \right\rfloor + 1.$$

Proof. By the arguments in the proof of Proposition 1.15, it follows that

$$d_{VC}(\mathbb{L}(\mathcal{H}_k)) = \max_{\pi \in \mathcal{F}_k} |\pi|.$$

Now, if $\pi \in \mathcal{F}_k$, then all of its blocks, but the last one, must have at least k points. The maximum number $|\pi|$ of blocks such that at most one has lesser than k points is bounded by the number of blocks obtained when the first $|\pi| - 1$ blocks have exactly k points, and the last one may have lesser than k points, that are

$$\left\lfloor \frac{2^{p^2}}{k} \right\rfloor + 1$$

blocks.

Observe that $\lfloor 2^{p^2}/k \rfloor$ is the maximum number of blocks of k points that can be generated by 2^{p^2} points, and the additional block is the last one containing the remaining points

when 2^{p^2} is not divisible by k. The $d_{VC}(\mathbb{L}(\mathcal{H}_k))$ is not always equal to this value, since there may not exist $\lfloor 2^{p^2}/k \rfloor$ disjoint size k intervals in $\mathcal{X}$, depending on the values of k and p. This can be seen by taking for example $p = 2$ and $k = 3$, since there is actually no interval of 3 points in the Boolean lattice of 4 variables. □

Remark 4.9. *Our definition of interval Boolean function is more comprehensive than that of [34, 86], since it comprehends, for instance, hypotheses which classify the points in the* $|\pi| - 1$ *union of intervals into* 0 *and the points in* $a_{|\pi|}$*, which may not be an interval, into* 1*. We chose this definition so that we may consider as* $\mathbb{L}(\mathcal{H})$ *a subset of the Partition Lattice Learning Space* $\mathbb{L}(\mathcal{H})$ *of* $\mathcal{H}$*, without having to exclude any hypothesis from the models remaining in* $\mathbb{L}(\mathcal{H}_k)$*, as can be established from the definition of* $\mathbb{L}(\mathcal{H}_k)$ *(cf. (4.11)). Proceeding in this way, if there is an implementation of an algorithm to estimate the error in each model in* $\mathbb{L}(\mathcal{H})$*, it can also be employed without changes to estimate the error of the models in* $\mathbb{L}(\mathcal{H}_k)$*. Moreover, we will show that* $\mathbb{L}(\mathcal{H}_k)$ *defined in this manner satisfies a U-curve property.*

4.4.3 U-curve property

The main advantage, from the computational perspective, of learning on $\mathcal{H}_k$ is that, besides the fact that there are lesser models in $\mathbb{L}(\mathcal{H}_k)$, the sup-weak U-curve property is satisfied on it under the simple loss function, when $\hat{L}$ is of the form (2.8). The proof of this result follows from Corollary 3.3.

Proposition 4.10. *The Learning Space* $\mathbb{L}(\mathcal{H}_k)$*, defined in* (4.11)*, under the simple loss function and* $\hat{L}$ *of form* (2.8)*, satisfies the sup-weak U-curve property.*

Proof. It is enough to show that $\mathcal{F}_k$ satisfies condition (3.10), so the result follows from Corollary 3.3.

Let $\pi, \pi_i \in \mathcal{F}_k$, with $\pi \preceq \pi_i$, and denote

$$\pi_i = (\pi \cap \pi_i) \cup \{a_1, \ldots, a_l\},$$

in which $\{a_1, \ldots, a_l\}$ are the blocks in π_i, but not in π. For each a_j, we claim that there exists a $b_j \in \mathcal{P}(\mathcal{X})$ such that

$$\pi_j := (\pi \setminus \{a_j \cup b_j\}) \cup \{a_j, b_j\}$$

is in $\mathcal{F}_k$, that is condition (3.10). Since $\pi \preceq \pi_i$, there exists a block in π that contains a_j, which we may write as $a_j \cup b_j$ and it remains to show that $\pi_j \in \mathcal{F}_k$. Observe that if a_j is the last block of π_i, then b_j is necessarily a union of disjoint size k intervals.

If a_j is a union of (possibly just one) disjoint size k intervals, then clearly $\pi_j \in \mathcal{F}_k$, since either b_j is a union of disjoint size k intervals, or $a_j \cup b_j$ is the last block of π and b_j is the last block of π_j. If a_j is not a union of size k intervals, then it is the last block of π_i and $a_j \cup b_j$ is the last block of π, what implies that b_j is a union of disjoint size k intervals, hence $\pi_j \in \mathcal{F}_k$ and a_j is its last block. □

Chapter 5

Discussion

5.1 Main results and implications

We proposed a systematic data-driven framework for Model Selection consisting of selecting the simplest global minimum, under an estimator $\hat{L}$, of a Learning Space, and then learning a hypothesis on it, seeking to approximate a target hypothesis of $\mathcal{H}$. The main novelty of the method is the concept of Learning Spaces, that are structured collections of candidate models, which cover $\mathcal{H}$ and contain only maximal models in the VC dimension sense: if $\mathcal{M}_1 \subset \mathcal{M}_2$ then $d_{VC}(\mathcal{M}_1) < d_{VC}(\mathcal{M}_2)$. Once a model $\hat{\mathcal{M}}$ is selected from $\mathbb{L}(\mathcal{H})$, we proposed two methods to learn on $\hat{\mathcal{M}}$, namely, with independent sample, when a sample independent of that used to select the model is used to learn on it, and by reusing, in which the same sample used to select the model is employed to learn on it.

Two important features emerged as a consequence of the proposed method:

- **Target model**: Fixed a Learning Space, the best scenario would be to learn on the simplest model in it that contains a target hypothesis, since no bias is introduced, as the error of this model is the same of $\mathcal{H}$, and, under this constraint, this is the model which will require fewer samples to have *low* estimation errors.
- **Better learn via $\mathbb{L}(\mathcal{H})$ than directly on $\mathcal{H}$**: It was theoretically and empirically evidenced that learning on an estimator $\hat{\mathcal{M}}$ of the target model may be better than learning directly on $\mathcal{H}$ via ERM, so learning via Learning Spaces is a new learning paradigm under which one can better learn by estimating the target model first.

We conclude from these features that

the lack of data may be mitigated by high computational power

since, by employing high computational power to search $\mathbb{L}(\mathcal{H})$ to estimate the target model $\mathcal{M}^\star$, one may better learn with a sample of fixed size when compared to learning directly on $\mathcal{H}$. The formalization of this fact is the main implication of learning via Learning Spaces.

A pragmatic manner of applying the abstract theory is to choose a concrete parametric representation of the hypotheses in $\mathcal{H}$. Then, within a suitable algebraic structure of such

a representation, one defines a Learning Space generator, that is a rule of how to obtain, via constraints in the parameters, a family of subsets of $\mathcal{H}$, partially ordered by inclusion, with the same algebraic structure considered for the parameters. A canonical example of this abstract system is the Boolean lattice for variable selection, in which the hypotheses are represented by the variables they depend on; the Boolean lattice algebra of the subsets of variables is considered; and the Learning Space generator associates each subset of variables with the hypotheses which depend solely on variables in the set.

Although abstract and general, and even if, at first sight, it may not be clear how to construct Learning Spaces, they emerge naturally on applications on which a *meaningful* parametric representation is available. By meaningful, we mean that each parameter is identifiable with some concrete concept, that is, the parameters are interpretable. This is the case on Examples 1.11 to 1.13, where the parameters represent variables or points of the classifier domain. Besides facilitating the development of Learning Spaces, the interpretability of the parameters has at least another two properties.

First, since one knows what each parameter means, it is possible to translate prior information into constraints in these parameters in order to (a) define a Learning Space generator, (b) replace $\mathcal{H}$ by a subset of it and (c) drop models from a Learning Space. Second, one may re-parametrize the hypotheses, or consider another constraint in the same parameters, to add other kinds of prior information, thus, generating another Learning Space for the same hypotheses space $\mathcal{H}$, as discussed in Example 1.13. Hence, on the one hand, the Learning Space is not unique and there is no general abstract formulae to build just one that would work for all applications, what would be a "canonical Learning Space". However, on the other hand, this flexibility when defining Learning Spaces makes it a customizable method, which can be instantiated for a large family of learning problems.

Still in this context, the Learning Space is a tool to express prior information, which can be of two types. On the one hand, there may be prior information that, although is inserted into a Learning Space, generates a constraint in $\mathcal{H}$, so the constrained Learning Space is actually of a $\mathcal{M} \subsetneq \mathcal{H}$. This is the case, for example, of the Learning Space for interval Boolean functions of Section 4.4, where the prior information inserted into the Boolean Partition Lattice Learning Space generated the hypotheses space $\mathcal{M}$ of the interval Boolean functions.

On the other hand, there may be prior information that, although exclude models from a $\mathbb{L}(\mathcal{H})$, do not generate any constraint in $\mathcal{H}$, since the new Learning Space still covers $\mathcal{H}$. This is the case, for example, when the Variable Selection Learning Space is obtained as a subset of the Partition Lattice Learning Space, as illustrated in Figure 1.13, when there is prior information that $h^\star$ does not depend on all variables. In this example, the hypotheses space $\mathcal{H}$ remains the same, but the structure of the Learning Space will be *favorable* to estimate $h^\star$ if it really does not depend on all variables.

Therefore, the Learning Spaces adds to the state-of-the-art in Machine Learning in what concerns translating prior information about the problem at hand into constraints to the learning process. Observe that, when one learns directly on $\mathcal{H}$ via ERM, it is not possible to insert prior information that does not constraint $\mathcal{H}$. This insertion of prior information is not new, since it is performed by variable selection methods and by certain constrained optimization algorithms in $\mathcal{H}$. Nevertheless, the Learning Space is an abstract object, of

which many Model Selection approaches are special cases, that can be instantiated in many domains, and may ease the representation of prior information.

The first facet of our data-driven systematic approach to Model Selection is its consistency, defined in terms of the convergence of the estimated model to the target one, and of the estimation errors to zero. We established the consistency for bounded and unbounded loss functions. The case of bounded loss functions was treated with the usual tools of VC theory, while the case of unbounded loss functions required some new technical results, which were established in Appendix A. We introduced the maximum discrimination error $\epsilon^\star$ and evidenced the importance of properly embedding all prior information into the Learning Space under the paradigm, supported by the theoretical and empirical results of this thesis, of better learning with a fixed sample size by properly modeling the Learning Space seeking to (a) have $\mathcal{M}^\star$ with small VC dimension and (b) have a great MDE $\epsilon^\star$.

More interesting than the consistency, which is expected from a probabilistic perspective, is that the rate of the convergences, specially of type IV estimation error, evidence scenarios in which it may be better to learn via Learning Spaces than directly on $\mathcal{H}$. This is a sub-product of the method, that was developed seeking the consistency and not aiming to beat ERM methods in $\mathcal{H}$.

The final facet of our data-driven, systematic and consistent approach to Model Selection is its non-exhaustiveness. We presented what is, to the best of our knowledge, the first formalization of the U-curve phenomenon on general lattices through the definition of the U-curve properties. We presented Learning Spaces which satisfy a U-curve property, and established an intuitive sufficient condition for the weak U-curve property.

Although the U-curve phenomenon is used as a stop criterion in many learning tasks, based on heuristics related to Occam's razor, the peaking phenomenon and the curse of dimensionality, a proof that it leads to optimal solutions is missing in many cases. For instance, it is missing in the Variable Selection Learning Space, on which it has been extensively used as a stop criterion in the literature of U-curve algorithms. Hence, after the concept of Learning Spaces, the second main contribution of this thesis is the formalization of the U-curve properties on Learning Spaces.

A consequence of the U-curve properties are the U-curve algorithms, which perform non-exhaustive searches of Learning Spaces to minimize an estimated error $\hat{L}$ when a U-curve property is satisfied, asserting the non-exhaustiveness of the method in this instance. A downside of the optimal U-curve algorithms is that they are highly complex, despite being non-exhaustive. However, we illustrated with applications that suboptimal U-curve algorithms are quite efficient, and may estimate hypotheses as good as an optimal algorithm, but with a fraction of the processing time.

Hence, the method is also non-exhaustive when a U-curve property is not satisfied, since a suboptimal algorithm may be suitable for the application at hand. For instance, suboptimal algorithms have been applied successfully to variable selection problems in the past. Even though the detailed study of U-curve algorithm is out of the scope of this thesis, we presented a couple of generic algorithms to perform non-exhaustive searches on $\mathbb{L}(\mathcal{H})$ when U-curve properties are satisfied that can be the starting point for more sophisticated algorithms.

After establishing that our data-driven systematic approach to Model Selection is consistent and non-exhaustive, we instantiated it to solve practical problems. With the simulated learning on a Partition Lattice Learning Space we illustrated some of its properties developed throughout the thesis, and observed some others, such as the fact that it may be better to learn via $\mathbb{L}(\mathcal{H})$ than directly on $\mathcal{H}$, that suboptimal algorithms may be as good as optimal algorithms, and that the quality of the approach may be related to the Conditional Entropy of the data distribution.

Then, in the application to forecasting sequences of binary values generated by variable order Markov chains, we observed that the U-curve algorithms have an embedded regularization, which is lost when the algorithm does not stop at a strong local minimum. Furthermore, we observed that suboptimal algorithms may estimate suitable hypotheses to solve the problem at hand, and that learning via Learning Space can be quite interpretable, since the inspection of the estimated hypothesis allows a fully understanding of the investment strategy generated by it.

The multilayer W-operators application was an attempt of defining a discrete learning framework, which is analogous to neural networks in a continuous setting, that can be implemented via a Learning Space. It outlined that the method may work even when the target hypothesis of each model is estimated by a suboptimal algorithm, and is not an ERM hypothesis, and illustrated how the solution may be interpretable via the inspection of the estimated windows. However, the estimation of multilayer W-operators is still too computationally complex, and it is necessary to further investigate how this method may be efficiently implemented and if it can attain competitive test errors to solve problems of interest. Finally, we defined a Learning Space for learning interval Boolean functions, which are specially suitable for image transformation tasks. This was the second example of the thesis of a Learning Space that satisfies a U-curve property.

The proposed framework has many important implications and countless topics for future researches, which we discuss in the remaining of this chapter.

5.2 Learning Spaces and penalized loss functions

Penalized Model Selection in Statistical Learning, as described in [99, Chapter 8], is a special case of the proposed framework given by considering the model error estimator

$$\hat{L}(\mathcal{M}) := L_{D_N}(\hat{h}_{\mathcal{M}}^{D_N}) + \text{pen}(\mathcal{M}) \qquad \mathcal{M} \in \mathbb{L}(\mathcal{H}), \tag{5.1}$$

that is the penalized resubstitution error, in which $\text{pen} : \mathbb{L}(\mathcal{H}) \mapsto \mathbb{R}_+$ is a penalty function. Observe that SRM methods also fit the scheme in (5.1) when $\mathbb{L}(\mathcal{H})$ is formed by a single continuous chain and the penalization is given by VC theory bounds.

The study of penalized Model Selection under the Learning Space framework would consist of proving results such as those in Chapters 2 and 3. First, one would have to establish the statistical consistency of the method, by showing the convergence of $\hat{\mathcal{M}}$ to $\mathcal{M}^\star$ with probability one, and the convergence in probability to zero of the estimation errors. This could be performed by applying very well-studied oracle and concentration

inequalities for Model Selection with penalized loss functions (see [6, 16, 82, 83, 99] and the references therein).

Second, one would have to establish conditions on the loss ℓ, the Learning Space $\mathbb{L}(\mathcal{H})$ and the penalty function under which a U-curve property is satisfied, through the existence of lattice convexity, for example. Observe that, in this case, the lattice convexity condition (3.11) reduces to

$$\begin{aligned}&\text{pen}(\mathcal{M}_1 \vee \mathcal{M}_1) + \text{pen}(\mathcal{M}_1 \wedge \mathcal{M}_1) - \text{pen}(\mathcal{M}_1) - \text{pen}(\mathcal{M}_2) \geq \\ &L_{D_N}(\hat{h}^{D_N}_{\mathcal{M}_1}) + L_{D_N}(\hat{h}^{D_N}_{\mathcal{M}_2}) - L_{D_N}(\hat{h}^{D_N}_{\mathcal{M}_1 \wedge \mathcal{M}_2}) - L_{D_N}(\hat{h}^{D_N}_{\mathcal{M}_1 \vee \mathcal{M}_2}),\end{aligned}$$

an inequality which associates the variation of the minimum empirical error within the considered models, with the variation of the penalty function on these models. The existence of a U-curve property in this scheme could be a tool to enhance the computational efficiency of Model Selection with penalization, hence be a great contribution to this field.

We leave this important case of penalized loss functions for future researches, since this thesis aimed to present the Learning Space framework not from the perspective of penalization, but rather as a general and consistent scheme for Model Selection based on resampling techniques such as cross-validation, so one does not have to *choose* a penalty function.

5.3 Decreasing the approximation error

The proposed framework may also be applied to try to reduce the approximation error, which is as follows. Let $\mathcal{H}^\star$ be the set of all measurable functions with domain $\mathcal{X}$, the support of X, and image $\mathcal{Y} \subset \mathbb{R}^d$, $d \geq 1$, which, fixed a loss function ℓ, is possibly a hypotheses space with infinite VC dimension. Denote

$$h_{Bayes} = \arg\min_{h \in \mathcal{H}^\star} L(h)$$

so that $L(h_{Bayes})$ is the Bayes error, the least error we can commit in $\mathcal{H}^\star$. Note that h_{Bayes} may or may not be in a fixed $\mathcal{H} \subset \mathcal{H}^\star$, and when it is not, we commit the error

$$L(h^\star) - L(h_{Bayes})$$

which is called approximation error (see [42, Chapter 12]). This error is, in general, not controllable and, to decrease it, one must increase $\mathcal{H}$, which in turn increases the risk of *overfitting* if the sample size is not great enough. This scenario is depicted in Figure 5.1 (a).

However, with the method presented in this paper, we may increase $\mathcal{H}$ mitigating the risk of *overfitting*, so we expect to be able to reduce the approximation error. In a perfect scenario, one would expect the scheme presented in Figure 5.1 (b): we choose a $\mathcal{H}$ highly complex, so it contains h_{Bayes}, but we actually learn on a relatively simple model $\hat{\mathcal{M}}$ which also contains h_{Bayes}. Even if h_{Bayes} is not in our complex $\mathcal{H}$, which we actually do not know,

we may expect the approximation error to be smaller, as, theoretically, we chose a $\mathcal{H}$ more complex than we would if we were unable to control *overfitting* nor search $\mathbb{L}(\mathcal{H})$.

This possible reduction on the approximation error is a topic that ought to be investigated from both a theoretical and empirical point of view in future researches.

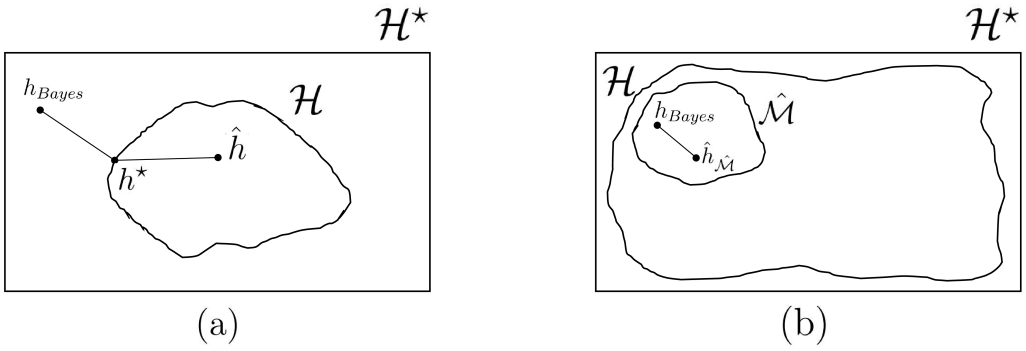

Figure 5.1: *The approximation error of hypotheses spaces.*

5.4 Perspectives in neural networks

Learning via Learning Spaces does not necessarily compete with neural networks, and may be employed to improve them. A given architecture of a neural network generates a hypotheses space $\mathcal{H}$, formed by the hypotheses obtained wandering all over its parameters' domain. Of course, such a $\mathcal{H}$ has subsets, and families of them satisfy the two axioms of Learning Spaces. Now, if one could develop a Learning Space generator which associates constraints in the architecture to models in a Learning Space, the abstract method proposed here could be applied to select architectures, since the constraints in the parameters would generate sub-architectures, so the Learning Space would represent a family of architectures, and the learning on it would be the learning of architectures. This instantiation has potential to become a relevant contribution to the field of Neural Architecture Search, since it would constitute a systematic and general approach to architecture learning, which could have major implications to the future of neural networks research.

However, since each parameter of a neural network is not exactly interpretable by itself, due to over-parametrization [41, 111], and the VC dimension of $\mathcal{H}$ is not exactly known in general for neural networks [144], the instantiation of the method in this case is not immediate and requires a further understanding of the concepts proposed here for neural networks. In order to develop a $\mathbb{L}(\mathcal{H})$ for Neural Architecture Search, it is necessary first to better understand how restrictions on an architecture imply restrictions on the hypotheses space $\mathcal{H}$ generated by it. Due to great redundancies in neural networks architectures, a restriction on its parameters does not necessarily imply a restriction on its hypotheses space, and a general description of the relation between these restrictions is an open problem.

In order to better understand the relation between these restrictions, it is necessary to revisit the vast literature about the approximation of classes of functions by neural

networks [39, 63, 64, 66, 71, 72, 78, 80, 91, 92, 93, 116, 117, 118, 157, 159, 161], among others, from another perspective. These results, in general, assert that, to represent all hypotheses in a space $\mathcal{H}$, it is necessary an architecture with certain features, that is, assert that a class of neural networks is a universal approximator of $\mathcal{H}$. Under the Learning Space, we need to study this problem from another perspective: given a neural network, what is the hypotheses space it can represent? From the answer to this question, it is possible to investigate which hypotheses space is obtained when the architecture is restricted. We leave this important study as a relevant topic for future research.

5.5 Limitations

A limitation of learning via Learning Spaces is that the quality of the method depends on the choice of Learning Space, hence requires a careful design of it. Indeed, there is no canonical Learning Space that solves well "all problems". Moreover, the Learning Space is not self-adaptative to the problem at hand, as some modern learning algorithms, so there is a need for the specification of it by the researcher. If this specification is badly performed (low MDE $\epsilon^\star$, no U-curve property, bad suboptimal U-curve algorithm, $\mathcal{M}^\star$ with high VC dimension,...), the method may not perform well.

Another limitation is that, even though the optimal U-curve algorithms are non-exhaustive, they are still NP-hard, and may not be computed within a reasonable time with the computational resources available today. Nevertheless, suboptimal algorithms may be quite efficient and perform as good as optimal algorithms, as has been evidenced in the applications. A recipe to enhance the computational efficiency of the method would be to combine prior information in the design of the Learning Space with suboptimal algorithms.

5.6 Topics for future researches

The major perspectives for future researches concern empirical studies of the proposed approach. Although theoretically sound, the overall *quality* of learning via Learning Spaces depends on the (low) complexity of $\mathcal{M}^\star$, on the (great) size of $\epsilon^\star$ and on the (low) computational complexity of $\hat{\mathcal{M}}$, which are all concepts dependent on the data distribution P. Hence, to attest the quality of this framework, it is necessary to perform empirical studies to assess how it behaves on specific problems of interest. Even though outside the scope of this thesis, that aimed at presenting the theoretical foundations of learning via Learning Spaces, empirical studies are important to better outline what applications can benefit the most from this framework. Indeed, we expected with the applications only to illustrate the potential of Learning Space based techniques, rather than develop definitive algorithms to solve practical problems, what we leave for future researches.

From an algorithmic perspective, it is necessary to refine the U-curve algorithms proposed here and implement them to specific practical problems. In special, the implementation of refined U-curve algorithms to learn on the Learning Space for interval Boolean functions, and for applications apart from classification problems, are promising topics for research. However, any efficient implementation of a U-curve algorithm pass

through the development of efficient representations for models in non-Boolean lattices and of search strategies in Learning Spaces, which consists in determining the best path to go through the chains in it, which are open problems. These are imperative topics of research for anyone seeking to implement U-curve algorithms for non-Boolean lattices. A more specific topic for research would be to enhance the efficiency of the learning of multilayer W-operators, investigating, for example, how it could benefit from GPU.

From the consistency perspective, there are also many topics for future research. For instance, one could investigate distribution dependent bounds for the estimation errors, specially for type IV, which could be compared with such bounds for type II estimation error in $\mathcal{H}$. In a distribution dependent scenario, it would be interesting to better understand the role of the Conditional Entropy on the quality of the approach. Moreover, there is a need to better investigate the consistency when the data is generated by an ergodic process, as is the case when forecasting sequences generated by variable order Markov chains.

A possible line of research would be not only to study other frameworks for learning on $\hat{\mathcal{M}}$ besides learning with independent sample and learning by reusing, but also to further study the consistency and potential selection bias when learning by reusing.

Besides the topics described here and in the previous chapters, there is a lot of ground to break in the direction of developing families of Learning Spaces which solve a class of problems, showing the existence of the U-curve property for other Learning Spaces, and developing optimal and suboptimal U-curve algorithms. In no way, we exhausted this subject, but only scratched its surface, since Learning Spaces based methods might be tools to understand and enhance the always increasing niche of high performance and computing demanding learning applications.

Final Remark

The approach proposed in this thesis led to an important practical property of Machine Learning: the lack of data may be mitigated by high computational power, since one can employ computer power to look for the target model, and then learn with fewer samples on it. In a context of continuous increasing and popularization of high computational power, this property may be the key to understanding why Machine Learning has become so important in all branches of science, even in the ones where data is expensive and hard to get.

Appendix A

Vapnik-Chervonenkis theory

In this appendix, we present the main ideas and results of classical Vapnik-Chervonenkis (VC) theory, the stone upon which the convergence results in Chapter 2 are built. The notation used here is that defined in Chapter 1, specially in Section 1.2. The presentation of the theory is a simplified merge of [149], [150], [42] and [36], where the simplicity of the arguments is preferred over the refinement of the bounds. Hence, we present results which support those in Chapter 2 and outline the main ideas of VC theory, even though are not the tightest available bounds. We omit the proof of more technical results, and note that refined versions of the results presented here may be found at one or more of the references.

This appendix is a review of VC theory, except for novel results presented in Section A.2.3 for the case of unbounded loss functions, where we obtain new bounds for relative type I estimation error by extending the results in [36].

A.1 Generalized Glivenko-Cantelli Problems

The main results of VC theory are based on a generalization of the Glivenko-Cantelli Theorem, which can be stated as follows. Recall that $\mathcal{D}_N = \{Z_1, \dots, Z_N\}$ is a sequence of independent random vectors with a same distribution $P(z) := \mathbb{P}(Z \leq z)$, for $z \in \mathcal{Z} \subset \mathbb{R}^d$, defined in a probability space $(\Omega, \mathcal{S}, \mathbb{P})$.

In order to easy notation, we assume, without loss of generality, that $\Omega = \mathbb{R}^d$, $\mathcal{S}$ is the Borel σ-algebra of $\mathbb{R}^d$, the random vector Z is the identity $Z(\omega) = \omega$, for $\omega \in \Omega$, and $\mathbb{P}$ is the unique probability measure such that $\mathbb{P}(\omega : \omega \leq z) = P(z)$, for all $z \in \mathbb{R}^d$. Define

$$P_{\mathcal{D}_N}(z) := \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}\{Z_i \leq z\}, \qquad z \in \mathcal{Z}$$

as the empirical distribution of Z under sample $\mathcal{D}_N$.

The assertion of the theorem below is that of [42, Theorem 12.4]. Its bottom line is that the empirical distribution of random variables converges uniformly to P with probability one. We postpone its proof to Section A.2.

Theorem A.1 (Glivenko-Cantelli Theorem). *Assume $d = 1$ and $\mathcal{Z} = \mathbb{R}$. Then, for a fixed $\epsilon > 0$ and N great enough,*

$$\mathbb{P}\left(\sup_{z\in\mathbb{R}}\left|P(z) - P_{D_N}(z)\right| > \epsilon\right) \leq 8(N+1)\exp\left\{-N\frac{\epsilon^2}{32}\right\}. \tag{A.1}$$

Applying Borel-Cantelli Lemma [25, Theorem 4.3] (cf. Lemma B.14) to (A.1) yields

$$\lim_{N\to\infty}\sup_{z\in\mathbb{R}}\left|P(z) - P_{D_N}(z)\right| = 0 \text{ with probability one.}$$

In other words, P_{D_N} converges uniformly almost surely to P.

Theorem A.1 has the flavor of VC theory results: a rate of uniform convergence of the empirical probability of a class of events to their real probability, which implies the almost sure convergence. Indeed, letting $\mathcal{S}^\star \subset \mathcal{S}$ be a class of events, that is not necessarily a σ-algebra, and denoting

$$\mathbb{P}_{D_N}(A) = \frac{1}{N}\sum_{i=1}^{N}\mathbb{1}\{Z_i \in A\},$$

as the empirical probability of event $A \in \mathcal{S}$ under sample $\boldsymbol{D}_N$, the probability in (A.1) can be rewritten as

$$\mathbb{P}\left(\sup_{A\in\mathcal{S}^\star}\left|\mathbb{P}(A) - \mathbb{P}_{D_N}(A)\right| > \epsilon\right), \tag{A.2}$$

in which $\mathcal{S}^\star = \{A_z : z \in \mathbb{R}\}$ with $A_z = \{\omega \in \Omega : \omega \leq z\}$. If probability (A.2) converges to zero when N tends to infinity for a class $\mathcal{S}^\star \subsetneq \mathcal{S}$, we say there exists a *partial uniform convergence* of the empirical measure to $\mathbb{P}$.

Observe that in (A.2) not only the class $\mathcal{S}^\star$ is fixed, but also the probability measure $\mathbb{P}$, hence partial uniform convergence is dependent on the class and the probability. Nevertheless, in a distribution-free framework, such as that of learning (cf. Section 1.2), the convergences of interest should hold for any data generating distribution, which is the case, for example, of Glivenko-Cantelli Theorem, that presents a rate of convergence (A.1) which does not depend on P, holding for any probability measure and random variable Z. Therefore, once a class $\mathcal{S}^\star$ of interest is fixed, partial uniform convergence should hold for any data generating distribution, a problem which can be stated as follows.

Let $\mathcal{P}$ be the class of all possible probability distributions of a random variable with support in $\mathcal{Z}$, and let $\mathcal{S}^\star$ be a class of events. The *generalized Glivenko-Cantelli problem* (GGCP) is to find a positive constant a and a function $b : \mathbb{Z}_+ \mapsto \mathbb{R}_+$, such that $\lim_{N\to\infty} b(N)/\exp cN = 0, \forall c > 0$, satisfying, for N great enough,[1]

$$\sup_{P\in\mathcal{P}}\mathbb{P}\left(\sup_{A\in\mathcal{S}^\star}\left|\mathbb{P}(A) - \mathbb{P}_{D_N}(A)\right| > \epsilon\right) \leq b(N)\exp\{-a\epsilon^2 N\}, \tag{A.3}$$

[1] In the presentation of [149, Chapter 2] it is assumed that b is a positive constant, not depending on sample size N. Nevertheless, having b as a function of N of an order lesser than exponential does not change the qualitative behavior of this convergence, that is, also guarantees the almost sure converge due to Borel-Cantelli Lemma.

in which $\mathbb{P}$ is to be understood as dependent on P, since it is the unique probability measure on the Borel σ-algebra that equals P on the events $\{\omega \in \Omega : \omega \preceq z\}, z \in \mathbb{R}^d$. If the events are of the form $A = \{w \in \Omega : Z(w) \preceq z\}, z \in \mathbb{R}$, then (A.3) is equivalent to (A.1), although in the latter it is implicit that it holds for any distribution P.

The investigation of GGCP revolves around deducing necessary and sufficient conditions on the class $\mathcal{S}^\star$ for (A.3) to hold. So now, let study these conditions in order to establish the almost sure convergence to zero of type I estimation error (cf. (1.3)) when the loss function is binary, what may be stated as a GGCP.

A.2 Convergence to zero of type I estimation error

A.2.1 Binary loss functions

Fix a hypotheses space $\mathcal{H}$, a binary loss function ℓ, and consider the class $\mathcal{S}^\star = \{A_h : h \in \mathcal{H}\}$, such that $\mathbb{1}\{z \in A_h\} = \ell(z, h) \in \{0, 1\}, z \in \mathcal{Z}, h \in \mathcal{H}$, that is, if $z \in A_h$ the loss is one, and otherwise it is zero. For example, if $Z = (X, Y)$, $\mathcal{H}$ is a functional hypotheses space and ℓ is the simple loss function, then A_h may be explicitly written as

$$A_h = \{\omega : h(X(\omega)) \neq Y(\omega)\}.$$

In this instance, the probability in the left-hand side of (A.3) may be written as

$$\mathbb{P}\left(\sup_{h\in\mathcal{H}} \left|\mathbb{E}(\ell(Z, h)) - \mathbb{E}_{\mathcal{D}_N}(\ell(Z, h))\right| > \epsilon\right), \tag{A.4}$$

in which $\mathbb{E}$ is expectation with respect to $\mathbb{P}$ and $\mathbb{E}_{\mathcal{D}_N}$ is the empirical mean under $\mathcal{D}_N$. With the notation of Section 1.2, this last probability equals

$$\mathbb{P}\left(\sup_{h\in\mathcal{H}} \left|L(h) - L_{\mathcal{D}_N}(h)\right| > \epsilon\right),$$

the tail probability of type I estimation error on $\mathcal{H}$.

For each fixed $h \in \mathcal{H}$, we are comparing in (A.4) the mean of a binary function with its empirical mean, so we may apply Hoeffding's inequality [69] (cf. Theorem B.13) to obtain

$$\mathbb{P}\left(\left|\mathbb{E}(\ell(Z, h)) - \mathbb{E}_{\mathcal{D}_N}(\ell(Z, N))\right| > \epsilon\right) \leq 2\exp\{-2\epsilon^2 N\},$$

from which follows a solution of type I estimation error GGCP when the cardinality of $\mathcal{H}$ is finite, by applying an elementary union bound:

$$\begin{aligned}\mathbb{P}\left(\sup_{h\in\mathcal{H}} |\mathbb{E}(\ell(Z, h)) - \mathbb{E}_{\mathcal{D}_N}(\ell(Z, N))| > \epsilon\right) &\leq \sum_{h\in\mathcal{H}} \mathbb{P}\left(\left|\mathbb{E}(\ell(Z, h)) - \mathbb{E}_{\mathcal{D}_N}(\ell(Z, N))\right| > \epsilon\right) \\ &\leq 2|\mathcal{H}|\exp\{-2\epsilon^2 N\},\end{aligned}$$

what establishes the almost sure convergence to zero of type I estimation error when $\mathcal{H}$ is finite and ℓ is binary.

In order to treat the case when $\mathcal{H}$ has infinitely many hypotheses, we rely on a modification of the proof of Glivenko-Cantelli Theorem. We present this proof, outlining which arguments are valid for arbitrary $\mathcal{S}^\star \subset \mathcal{S}$, and which demand that its events are of form $A_z = \{w \in \Omega : \omega \leq z\}$. This is essentially the proof presented in [42, Theorem 12.4]. But before discussing the proof, we define the shatter coefficient of a class $\mathcal{S}^\star \subset \mathcal{S}$ of events in the Borel σ-algebra of $\mathbb{R}^d$.

Definition A.2. *Fix $\mathcal{S}^\star \subset \mathcal{S}$ and let*

$$\mathcal{G}_{\mathcal{S}^\star} = \{h_A(z) = \mathbb{1}\{z \in A\} : A \in \mathcal{S}^\star\}$$

be the characteristic functions of the sets in $\mathcal{S}^\star$. We define the shatter coefficient of $\mathcal{S}^\star$ as

$$S(\mathcal{S}^\star, N) := S(\mathcal{G}_{\mathcal{S}^\star}, N),$$

in which $S(\mathcal{G}_{\mathcal{S}^\star}, N)$ is the shatter coefficient of $\mathcal{G}_{\mathcal{S}^\star}$ (cf. Definition 1.4). From this definition follows that

$$d_{VC}(\mathcal{S}^\star) = d_{VC}(\mathcal{G}_{\mathcal{S}^\star}).$$

The shatter coefficient and VC dimension of a class $\mathcal{S}^\star$ is related to the dichotomies this class can build with N points by considering whether a point is in each set or not. We are now in position to present a proof of the Glivenko-Cantelli Theorem.

Proof of Glivenko-Cantelli Theorem. In order to easy notation, denote

$$\nu(A) = \mathbb{P}(A) \qquad \text{and} \qquad \nu_N(A) = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\{Z_i \in A\}$$

for $A \in \mathcal{S}^\star \subset \mathcal{S}$, where $\mathcal{S}^\star$ is any subset of $\mathcal{S}$. Define a sample $\mathcal{D}'_N = \{Z'_1, \dots, Z'_N\}$ of independent random vectors with distribution P, which is independent of $\mathcal{D}_N$, and let

$$\nu'_N(A) = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\{Z'_i \in A\}$$

for $A \in \mathcal{S}^\star$.

Let $B \in \mathcal{S}^\star$ be such that $|\nu_N(B) - \nu(B)| > \epsilon$, if a set satisfying this condition exists, and let it be an arbitrary set if this is not the case. Observe that B depends on sample $\mathcal{D}_N$ since its definition depends on ν_N. Then,

$$\begin{aligned}
\mathbb{P}\left(\sup_{A\in\mathcal{S}^\star} \left|\nu_N(A) - \nu'_N(A)\right| > \epsilon/2\right) &\geq \mathbb{P}\left(\left|\nu_N(B) - \nu'_N(B)\right| > \epsilon/2\right) \\
&\geq \mathbb{P}\left(|\nu_N(B) - \nu(B)| > \epsilon, \left|\nu'_N(B) - \nu(B)\right| < \epsilon/2\right) \\
&= \mathbb{E}\left(\mathbb{1}\{|\nu_N(B) - \nu(B)| > \epsilon\}\mathbb{P}\left(\left|\nu'_N(B) - \nu(B)\right| < \epsilon/2 | \mathcal{D}_N\right)\right),
\end{aligned}$$

in which the expectation is over the possible samples $\mathcal{D}_N$. By Chebyshev inequality, since

$\nu'_N(B)$ is independent of $\mathcal{D}_N$ and, conditioned on $\mathcal{D}_N$, B is a fixed set, the probability inside the expectation above is greater or equal to

$$1 - \frac{4\nu(B)(1-\nu(B))}{N\epsilon^2} \geq 1 - \frac{1}{N\epsilon^2} \geq 1/2$$

whenever N is great enough, i.e., $N \geq 2/\epsilon^2$. From the deductions above follows

$$\begin{aligned}\mathbb{P}\left(\sup_{A\in\mathcal{S}^\star}\left|\nu_N(A) - \nu'_N(A)\right| > \epsilon/2\right) &\geq \frac{1}{2}\mathbb{P}\left(|\nu_N(B) - \nu(B)| > \epsilon\right) \\ &\geq \frac{1}{2}\mathbb{P}\left(\sup_{A\in\mathcal{S}^\star}|\nu_N(A) - \nu(A)| > \epsilon\right), \end{aligned} \tag{A.5}$$

in which the last inequality follows from the definition of B, since if $\sup_{A\in\mathcal{S}^\star}|\nu_N(A) - \nu(A)| > \epsilon$, then there exists a B with $|\nu_N(B) - \nu(B)| > \epsilon$, so the event in the last probability implies the event in the second to last.

Now, let $\sigma_1, \dots, \sigma_N$ be independent random variables, also independent of $\mathcal{D}_N$ and $\mathcal{D}'_N$, with $\mathbb{P}(\sigma_i = 1) = \mathbb{P}(\sigma_i = -1) = 1/2$. Since all vectors in $\mathcal{D}_N$ and $\mathcal{D}'_N$ are independent and identically distributed, the following sums have the same distribution:

$$\sup_{A\in\mathcal{S}^\star}\left|\sum_{i=1}^N \mathbb{1}\{Z_i \in A\} - \mathbb{1}\{Z'_i \in A\}\right| \sim \sup_{A\in\mathcal{S}^\star}\left|\sum_{i=1}^N \sigma_i\left(\mathbb{1}\{Z_i \in A\} - \mathbb{1}\{Z'_i \in A\}\right)\right|.$$

Therefore, by (A.5) it follows that

$$\begin{aligned}\mathbb{P}\left(\sup_{A\in\mathcal{S}^\star}|\nu_N(A) - \nu(A)| > \epsilon\right) &\leq 2\,\mathbb{P}\left(\sup_{A\in\mathcal{S}^\star}\frac{1}{N}\left|\sum_{i=1}^N \mathbb{1}\{Z_i \in A\} - \mathbb{1}\{Z'_i \in A\}\right| > \epsilon/2\right) \\ &= 2\,\mathbb{P}\left(\sup_{A\in\mathcal{S}^\star}\frac{1}{N}\left|\sum_{i=1}^N \sigma_i\left(\mathbb{1}\{Z_i \in A\} - \mathbb{1}\{Z'_i \in A\}\right)\right| > \epsilon/2\right). \end{aligned} \tag{A.6}$$

By applying a union bound to the last probability, we have that it is lesser or equal to

$$\mathbb{P}\left(\sup_{A\in\mathcal{S}^\star}\frac{1}{N}\left|\sum_{i=1}^N \sigma_i\mathbb{1}\{Z_i \in A\}\right| > \epsilon/4\right) + \mathbb{P}\left(\sup_{A\in\mathcal{S}^\star}\frac{1}{N}\left|\sum_{i=1}^N \sigma_i\mathbb{1}\{Z'_i \in A\}\right| > \epsilon/4\right),$$

which equals

$$2\,\mathbb{P}\left(\sup_{A\in\mathcal{S}^\star}\frac{1}{N}\left|\sum_{i=1}^N \sigma_i\mathbb{1}\{Z_i \in A\}\right| > \epsilon/4\right). \tag{A.7}$$

All deductions up to now do not depend on the nature of the events in $\mathcal{S}^\star$. From now on, we assume that $\mathcal{Z} \subset \mathbb{R}$ and the events in $\mathcal{S}^\star$ are of the form $A_z = \{\omega : \omega \leq z\}$, for $z \in \mathbb{R}$. We now bound (A.7) by conditioning on the sample $\mathcal{D}_N$.

Fix a sequence $z_1, \dots, z_N \in \mathbb{R}$, and note that, as z ranges over $\mathbb{R}$, the number of different

vectors

$$(\mathbb{1}\{z_1 \leq z\}, \dots, \mathbb{1}\{z_N \leq z\}) = (\mathbb{1}\{z_1 \in A_z\}, \dots, \mathbb{1}\{z_N \in A_z\})$$

obtained is at most $N + 1$. In other words, this means the N-th shatter coefficient (cf. Definition A.2) of the class $S^\star$ is $N + 1$: $S(S^\star, N) = N + 1$.

Writing the probability in (A.7) as

$$\mathbb{P}\left(\sup_{z \in \mathbb{R}} \frac{1}{N} \left| \sum_{i=1}^{N} \sigma_i \mathbb{1}\{Z_i \leq z\} \right| > \epsilon/4 \right)$$

we observe that, conditioned on $\mathcal{D}_N$, the supremum inside the probability is actually a maximum over at most $N + 1$ possible values for the sum. Hence, applying a union bound, we obtain

$$\mathbb{P}\left(\sup_{A \in S^\star} \frac{1}{N} \left| \sum_{i=1}^{N} \sigma_i \mathbb{1}\{Z_i \in A\} \right| > \epsilon/4 \middle| \mathcal{D}_N \right) \leq (N + 1) \sup_{A \in S^\star} \mathbb{P}\left(\frac{1}{N} \left| \sum_{i=1}^{N} \sigma_i \mathbb{1}\{Z_i \in A\} \right| > \epsilon/4 \middle| \mathcal{D}_N \right).$$

Since, with $\mathcal{D}_N$ fixed, $\sum_{i=1}^{N} \sigma_i \mathbb{1}\{Z_i \in A\}$ is the sum of N independent zero mean random variables taking values in $\{-1, 1\}$, we may apply Hoeffding's Inequality (cf. Theorem B.13) to each probability within the supremum in the expression above yielding

$$\mathbb{P}\left(\sup_{A \in S^\star} \frac{1}{N} \left| \sum_{i=1}^{N} \sigma_i \mathbb{1}\{Z_i \in A\} \right| > \epsilon/4 \middle| \mathcal{D}_N \right) \leq 2(N + 1) \exp\left\{ -\frac{N^2}{32} \right\}.$$

Finally, by taking the expectation over all possible samples $\mathcal{D}_N$ on both sides of the above expression follows

$$\mathbb{P}\left(\sup_{A \in S^\star} \frac{1}{N} \left| \sum_{i=1}^{N} \sigma_i \mathbb{1}\{Z_i \in A\} \right| > \epsilon/4 \right) \leq 2(N + 1) \exp\left\{ -\frac{N^2}{32} \right\}.$$

In summary, recalling that $S(S^\star, N) = N + 1$ and the results in (A.6) and (A.7), we obtain

$$\mathbb{P}\left(\sup_{A \in S^\star} |\nu_N(A) - \nu(A)| > \epsilon \right) \leq 8S(S^\star, N) \exp\left\{ -N\frac{\epsilon^2}{32} \right\}, \tag{A.8}$$

and the proof is complete. □

From the proof above, it is clear that (A.8) is true for any class $S^\star \subset S$, with a respective shatter coefficient $S(S^\star, N)$. Indeed, the only argument in the proof that needs correction is that, when A ranges over $S^\star$, the number of different vectors

$$(\mathbb{1}\{z_1 \in A\}, \dots, \mathbb{1}\{z_N \in A\})$$

obtained is at most $S(S^\star, N)$, by definition of shatter coefficient. Then, substituting $N+1$ by $S(S^\star, N)$ in the proof, it remains valid and asserts the following result due to [152].

Theorem A.3. *For any probability measure $\mathbb{P}$ and class of sets $S^\star \subset S$, for fixed $N \in \mathbb{Z}$*

and $\epsilon > 0$, it is true that

$$\mathbb{P}\left(\sup_{A\in\mathcal{S}^\star}\left|\mathbb{P}(A)-\mathbb{P}_{D_N}(A)\right|>\epsilon\right)\leq 8S(\mathcal{S}^\star,N)\exp\left\{-N\frac{\epsilon^2}{32}\right\}.$$

From this theorem follows a bound for tail probabilities of type I estimation error when ℓ is binary.

Corollary A.4. *Fix a hypotheses space $\mathcal{H}$ and a loss function $\ell : \mathcal{Z}\times\mathcal{H}\mapsto\{0,1\}$. Let $\mathcal{S}^\star=\{A_h : h\in\mathcal{H}\}$, with*

$$\mathbb{1}\{z\in A_h\}=\ell(z,h), z\in\mathcal{Z}, h\in\mathcal{H}.$$

Then,

$$\mathbb{P}\left(\sup_{h\in\mathcal{H}}\left|L(h)-L_{D_N}(h)\right|>\epsilon\right)\leq 8\ S(\mathcal{H},N)\exp\left\{-N\frac{\epsilon^2}{32}\right\}, \tag{A.9}$$

with

$$S(\mathcal{H},N) := S(\mathcal{S}^\star,N).$$

Remark A.5. *We remark that $S(\mathcal{S}^\star,N)=S(\mathcal{G}_{\mathcal{H},\ell},N)$, as defined in Definition 1.5, when the loss ℓ is binary. Observe that $\mathcal{S}^\star$ depends on ℓ, although we omit the dependence to easy notation.*

The calculation of the quantities on the right-hand side of (A.9) is not straightforward, since the shatter coefficient is not easily determined for arbitrary N. Nevertheless, the shatter coefficient may be bounded by a quantity depending on the VC dimension of $\mathcal{H}$. This is the content of [149, Theorem 4.3], which we state without proof.

Theorem A.6. *If $d_{VC}(\mathcal{H})<\infty$, then*

$$\ln S(\mathcal{H},N)\begin{cases}=N\ln 2, & \text{if } N\leq d_{VC}(\mathcal{H})\\ \leq d_{VC}(\mathcal{H})\left(1+\ln\frac{N}{d_{VC}(\mathcal{H})}\right), & \text{if } N>d_{VC}(\mathcal{H})\end{cases}.$$

Remark A.7. *Theorem A.6 is true for any loss function ℓ, not only binary. It also holds for $S(\mathcal{S}^\star,N)$.*

Combining this theorem with Corollary A.4, we obtain the following result.

Corollary A.8. *Under the hypotheses of Corollary A.4 it holds*

$$\mathbb{P}\left(\sup_{h\in\mathcal{H}}\left|L(h)-L_{D_N}(h)\right|>\epsilon\right)\leq 8\ \exp\left\{d_{VC}(\mathcal{H})\left(1+\ln\frac{N}{d_{VC}(\mathcal{H})}\right)-N\frac{\epsilon^2}{32}\right\}. \tag{A.10}$$

In particular, if $d_{VC}(\mathcal{H})<\infty$, not only (A.10) converges to zero, but also

$$\sup_{h\in\mathcal{H}}\left|L(h)-L_{D_N}(h)\right|\xrightarrow[N\to\infty]{}0,$$

with probability one by Borel-Cantelli Lemma.

From Corollary A.8 follows the convergence to zero of type I estimation error when the loss function is binary and $d_{VC}(\mathcal{H})$ is finite. We now extend this result to real-valued bounded loss functions.

A.2.2 Bounded loss functions

Assume the loss function is bounded, that is, for all $z \in \mathcal{Z}$ and $h \in \mathcal{H}$,

$$0 \leq \ell(z,h) \leq C < \infty, \tag{A.11}$$

for a positive constant $C \in \mathbb{R}_+$. Throughout this section, a constant C satisfying (A.11) is fixed.

For any $h \in \mathcal{H}$, by definition of Lebesgue-Stieltjes integral, we have that

$$L(h) = \int_{\mathcal{Z}} \ell(z,h)\, dP(z) = \lim_{n\to\infty} \sum_{k=1}^{n-1} \frac{C}{n} \mathbb{P}\left(\ell(Z,h) > \frac{kC}{n}\right),$$

recalling that Z is a random variable with distribution P. In a same manner, we may also write the empirical error under $\mathcal{D}_N$ as

$$L_{D_N}(h) = \frac{1}{N}\sum_{i=1}^{N} \ell(Z_i,h) = \lim_{n\to\infty} \sum_{k=1}^{n-1} \frac{C}{n} \mathbb{P}_{D_N}\left(\ell(Z,h) > \frac{kC}{n}\right),$$

recalling that $\mathbb{P}_{D_N}$ is the empirical measure according to $\mathcal{D}_N$.

From the representation of L and L_{D_N} described above, we have that, for each $h \in \mathcal{H}$ fixed,

$$\begin{aligned}
\left|L(h) - L_{D_N}(h)\right| &= \left|\lim_{n\to\infty} \sum_{k=1}^{n-1} \frac{C}{n}\left(\mathbb{P}\left(\ell(Z,h) > \frac{kC}{n}\right) - \mathbb{P}_{D_N}\left(\ell(Z,h) > \frac{kC}{n}\right)\right)\right| \\
&\leq \left|\lim_{n\to\infty} \sum_{k=1}^{n-1} \frac{C}{n} \sup_{0\leq\beta\leq C} \left(\mathbb{P}\left(\ell(Z,h) > \beta\right) - \mathbb{P}_{D_N}\left(\ell(Z,h) > \beta\right)\right)\right| \\
&\leq \lim_{n\to\infty} \sum_{k=1}^{n-1} \frac{C}{n} \sup_{0\leq\beta\leq C} \left|\mathbb{P}\left(\ell(Z,h) > \beta\right) - \mathbb{P}_{D_N}\left(\ell(Z,h) > \beta\right)\right| \\
&= C \sup_{0\leq\beta\leq C} \left|\int_{\mathcal{Z}} \mathbb{1}\{\ell(z,h) > \beta\}\, dP(z) - \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\{\ell(Z_i,h) > \beta\}\right|.
\end{aligned}$$

We conclude that

$$\mathbb{P}\left(\sup_{h\in\mathcal{H}} \left|L(h) - L_{D_N(h)}\right| > \epsilon\right) \leq \mathbb{P}\left(\sup_{\substack{h\in\mathcal{H} \\ 0\leq\beta\leq C}} \left|\int_{\mathcal{Z}} \mathbb{1}\{\ell(z,h) > \beta\}\, dP(z) - \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\{\ell(Z_i,h)\}\right| > \frac{\epsilon}{C}\right).$$

Since the right-hand side of the expression above is a GGCP with

$$\mathcal{S}^\star = \{\{z \in \mathcal{Z} : \ell(z,h) > \beta\} : h \in \mathcal{H}, 0 \leq \beta \leq C\}$$

and, recalling the definition of shatter coefficient of $\mathcal{H}$ under a real-valued loss function ℓ (cf. Definition 1.4), we note that

$$S(\mathcal{G}_{\mathcal{H},\ell}, N) = S(\mathcal{S}^\star, N) \qquad \text{hence} \qquad d_{VC}(\mathcal{H}) = d_{VC}(\mathcal{S}^\star)$$

so a bound for the tail probabilities of type I estimation error when the loss function is bounded follows immediately from Theorems A.3 and A.6.

Corollary A.9. *Fix a hypotheses space $\mathcal{H}$ and a loss function $\ell : \mathcal{Z} \times \mathcal{H} \mapsto \mathbb{R}_+$, with $0 \leq \ell(z,h) \leq C$ for all $z \in \mathcal{Z}, h \in \mathcal{H}$. Then,*

$$\mathbb{P}\left(\sup_{h\in\mathcal{H}} \left|L(h) - L_{D_N}(h)\right| > \epsilon\right) \leq 8 \exp\left\{ d_{VC}(\mathcal{H}) \left(1 + \ln \frac{N}{d_{VC}(\mathcal{H})}\right) - N\frac{\epsilon^2}{32C^2} \right\}. \tag{A.12}$$

In particular, if $d_{VC}(\mathcal{H}) < \infty$, not only (A.12) converges to zero, but also

$$\sup_{h\in\mathcal{H}} \left|L(h) - L_{D_N}(h)\right| \xrightarrow[N\to\infty]{} 0,$$

with probability one by Borel-Cantelli Lemma.

It remains to treat the case of unbounded loss functions, which requires a different approach.

A.2.3 Unbounded loss functions

In the case of unbounded loss functions, distribution-free bounds as those obtained for bounded losses are not quite possible since, if the data distribution has *very heavy tails*, the convergence to zero of type I estimation error may not be exponential, even though it may hold, or may not hold at all. In this section, we establish conditions on P and $\mathcal{H}$ for the convergence in probability to zero of estimation errors when ℓ is unbounded. We repeat some definitions and concepts presented in Section 2.4 in order to facilitate the understanding.

We start by defining what it means for P to have heavy tails in this scenario, following ideas similar to [149, Section 5.7]. In what follows, we assume that $\ell(z,h) \geq 1$, for all $z \in \mathcal{Z}$ and $h \in \mathcal{H}$ (see Remark A.17 for an explanation). This can be accomplished by summing one to ℓ, without loss of generality, since the minimizers of errors L and L_{D_N} will still be the same.

For $1 < p < \infty$ and a fixed hypotheses space $\mathcal{H}$, define

$$\tau_p := \sup_{h\in\mathcal{H}} \frac{\left(\int_{\mathcal{Z}} \ell^p(z,h)\, dP(z)\right)^{\frac{1}{p}}}{\int_{\mathcal{Z}} \ell(z,h)\, dP(z)} = \sup_{h\in\mathcal{H}} \frac{L^p(h)}{L(h)},$$

in which $L^p(h) := \left(\int_{\mathcal{Z}} \ell^p(z,h)\, dP(z)\right)^{\frac{1}{p}}$. Although τ_p depends on P and $\mathcal{H}$, we omit them to simplify notation, since they will be clear from context. The weight of the tails of distribution P may be defined based on τ_p, as follows.

Definition A.10. *We say that distribution P on $\mathcal{H}$ under ℓ has:*

- *Light tails, if there exists a $p > 2$ such that $\tau_p < \infty$.*
- *Heavy tails, if there exists a $1 < p \leq 2$ such that $\tau_p < \infty$, but $\tau_p = \infty$ for all $p > 2$.*
- *Very heavy tails, if $\tau_p = \infty$ for all $p > 1$.*

In order to obtain bounds for estimation errors, we assume that P has at most heavy tails, which means there exists a $p > 1$, and possibly lesser than 2, with

$$\tau_p < \tau^\star < \infty \tag{A.13}$$

that is, P is in a class of distributions for which (A.13) holds. From now on, fix a $p > 1$ and a $\tau^\star$ such that (A.13) holds.

Besides this constraint in the distribution tails, we also assume that the loss function ℓ has a finite moment of order p, under P and under the empirical measure. That is, denoting by

$$\ell(\mathcal{D}_N, h) := (\ell(Z_1, h), \dots, \ell(Z_N, h)) \in \mathbb{R}^N \setminus \{0\},$$

the vector sample point errors, and defining, for $1 \leq q \leq p$,

$$L^q_{\mathcal{D}_N}(h) := \frac{\|\ell(\mathcal{D}_N, h)\|_q}{N^{\frac{1}{q}}}, \tag{A.14}$$

we assume that

$$\sup_{h \in \mathcal{H}} L^p_{\mathcal{D}_N}(h) < \infty \qquad \text{and} \qquad \sup_{h \in \mathcal{H}} L^p(h) < \infty, \tag{A.15}$$

in which the first inequality should hold with probability one, for all possible samples $\mathcal{D}_N$, and $\|\cdot\|_q$ is the q-norm in $\mathbb{R}^N$.

Since the moments L^p are increasing in p, (A.15) actually implies (A.13), so the former is the non-trivial constraint in distribution P. Although this is a deviation from the distribution-free framework, it is a mild constraint in distribution P which ought to be satisfied by the distributions of data used in many applications of interest (see [149, Section 5.7] and Remark A.16 for more details).

The first condition in (A.15) is more a feature of the loss function, than of distribution P. Actually, one can bound $L^p_{\mathcal{D}_N}(h)$ by a quantity depending on N and $L^q_{\mathcal{D}_N}(h)$ with $1 \leq q < p$, for any sample $\mathcal{D}_N$ of any distribution P. This is the content of the next lemma, which will be useful later on, and that implies the following: if $\sup_{h \in \mathcal{H}} L^1_{\mathcal{D}_N}(h) = \sup_{h \in \mathcal{H}} L_{\mathcal{D}_N}(h) < \infty$, then $\sup_{h \in \mathcal{H}} L^p_{\mathcal{D}_N}(h) < \infty$ for any $1 < p < \infty$, for N and $\mathcal{H}$ fixed.

Lemma A.11. *For fixed $\mathcal{H}$, $N \geq 1$ and $1 \leq q < p$, it follows that*

$$1 \leq \frac{L^p_{\mathcal{D}_N}(h)}{L^q_{\mathcal{D}_N}(h)} \leq N^{\frac{1}{q} - \frac{1}{p}}$$

for all $h \in \mathcal{H}$.

Proof. Recalling definition (A.14), we have that

$$\frac{L^p_{D_N}(h)}{L^q_{D_N}(h)} = N^{\frac{1}{q}-\frac{1}{p}} \frac{\|\ell(\mathcal{D}_N, h)\|_p}{\|\ell(\mathcal{D}_N, h)\|_q},$$

so it is enough to show that

$$N^{\frac{1}{p}-\frac{1}{q}} \leq \frac{\|\ell(\mathcal{D}_N, h)\|_p}{\|\ell(\mathcal{D}_N, h)\|_q} \leq 1.$$

Now, the right inequality above is clear, since if $w \in \mathbb{R}^N$ is such that $\|w\|_q = 1$, then

$$\|w\|_p^p = \sum_{i=1}^N |w_i|^p \leq \sum_{i=1}^N |w_i|^q = 1,$$

so the result follows when $\|w\|_q = 1$ by elevating both sided to the $1/p$ power. To conclude the proof it is enough to see that, for any $w \in \mathbb{R}^N \setminus \{0\}$,

$$\|w\|_p = \|w\|_q \left\| \frac{w}{\|w\|_q} \right\|_p \leq \|w\|_q \left\| \frac{w}{\|w\|_q} \right\|_q = \|w\|_q.$$

The left inequality is a consequence of Hölder's inequality, since, for $w \in \mathbb{R}^N$,

$$\sum_{i=1}^N |w_i|^q \cdot 1 \leq \left(\sum_{i=1}^N |w_i|^p \right)^{\frac{q}{p}} N^{1-\frac{q}{p}},$$

and the result follows by taking the $1/q$ power on both sides. □

For unbounded losses, rather than considering the convergence of type I estimation error to zero, we will consider the convergence of the relative type I estimation error, defined as

$$\sup_{h \in \mathcal{H}} \left| \frac{L(h) - L_{D_N}(h)}{L(h)} \right|. \tag{A.16}$$

On the one hand, since $L(h)$ may be unbounded, having L arbitrarily close to L_{D_N} uniformly in $\mathcal{H}$ is not reasonable, since this difference is expected to be proportional to L, that is, the *arbitrarily close* concept should be relative to the value of L. On the other hand, it seems reasonable that (A.16) converges almost surely to zero, since its denominator is controlling for the possibility of $L(h)$ to be arbitrarily large.

In order to establish bounds for the tail probabilities of (A.16) when (A.13) and (A.15) hold, we rely on the following novel technical theorem.

Theorem A.12. *Let $q = \sqrt{p}$. For any hypotheses space $\mathcal{H}$, loss function ℓ satisfying $\ell(h, z) \geq 1$, and $0 < \epsilon < 1$, it holds*

$$\mathbb{P}\left(\sup_{h \in \mathcal{H}} \left| \frac{L(h) - L_{D_N}(h)}{L(h)} \right| > \tau^\star \epsilon \right) \leq \mathbb{P}\left(\sup_{h \in \mathcal{H}} \frac{L(h) - L_{D_N}(h)}{L^p(h)} > \epsilon \right)$$

$$+\mathbb{P}\left(\sup_{h\in\mathcal{H}}\frac{L'_{D_N}(h)-L'(h)}{L'^{q}_{D_N}(h)}>\frac{\epsilon}{N^{\frac{1}{q}-\frac{1}{p}}}\right)+\mathbb{P}\left(\sup_{h\in\mathcal{H}}\frac{L_{D_N}(h)-L(h)}{L^{p}_{D_N}(h)}>\frac{\epsilon(1-\epsilon)}{N^{\frac{1}{q}-\frac{1}{p}}}\right)$$

in which L', L'_{D_N} *and* $L'^{k}_{D_N}$ *are the respective errors and* k *moments of loss function* $\ell'(z,h) := (\ell(z,h))^q$.

Proof. We first note that

$$\sup_{h\in\mathcal{H}}\frac{L_{D_N}(h)-L(h)}{L(h)}>\tau^\star\epsilon \implies \sup_{h\in\mathcal{H}}\left(\frac{L^q(h)}{L(h)}\frac{1}{\tau^\star}\right)\frac{L_{D_N}(h)-L(h)}{L^q(h)}>\epsilon$$
$$\implies \sup_{h\in\mathcal{H}}\frac{L_{D_N}(h)-L(h)}{L^q(h)}>\epsilon$$

since

$$\sup_{h\in\mathcal{H}}\left(\frac{L^q(h)}{L(h)}\frac{1}{\tau^\star}\right)\leq\sup_{h\in\mathcal{H}}\left(\frac{L^p(h)}{L(h)}\frac{1}{\tau^\star}\right)\leq 1$$

by (A.13). With an analogous deduction, it follows that

$$\sup_{h\in\mathcal{H}}\frac{L(h)-L_{D_N}(h)}{L(h)}>\tau^\star\epsilon \implies \sup_{h\in\mathcal{H}}\frac{L(h)-L_{D_N}(h)}{L^p(h)}>\epsilon.$$

Hence, the probability on the left hand-side of the statement is lesser or equal to

$$\mathbb{P}\left(\sup_{h\in\mathcal{H}}\frac{L(h)-L_{D_N}(h)}{L^p(h)}>\epsilon\right)+\mathbb{P}\left(\sup_{h\in\mathcal{H}}\frac{L_{D_N}(h)-L(h)}{L^q(h)}>\epsilon\right), \tag{A.17}$$

so it is enough to properly bound the second probability in (A.17).

In order to do so, we will intersect the event inside the probability with the following event, and its complement:

$$\sup_{h\in\mathcal{H}}\frac{L^q_{D_N}(h)-\delta L^p_{D_N}(h)}{L^q(h)}\leq 1 \iff \sup_{h\in\mathcal{H}}\frac{L^q_{D_N}(h)-L^q(h)}{L^p_{D_N}(h)}\leq\delta,$$

in which

$$\delta := \frac{\epsilon}{N^{\frac{1}{q}-\frac{1}{p}}}.$$

Proceeding in this way, we conclude that

$$\mathbb{P}\left(\sup_{h\in\mathcal{H}}\frac{L_{D_N}(h)-L(h)}{L^q(h)}>\epsilon\right)\leq\mathbb{P}\left(\sup_{h\in\mathcal{H}}\frac{L^q_{D_N}(h)-L^q(h)}{L^p_{D_N}(h)}>\delta\right)$$
$$+\mathbb{P}\left(\sup_{h\in\mathcal{H}}\left(\frac{L^q_{D_N}(h)-\delta L^p_{D_N}(h)}{L^q(h)}\right)\frac{L_{D_N}(h)-L(h)}{L^q_{D_N}(h)-\delta L^p_{D_N}(h)}>\epsilon,\sup_{h\in\mathcal{H}}\frac{L^q_{D_N}(h)-\delta L^p_{D_N}(h)}{L^q(h)}\leq 1\right)$$
$$\leq\mathbb{P}\left(\sup_{h\in\mathcal{H}}\frac{L^q_{D_N}(h)-L^q(h)}{L^p_{D_N}(h)}>\delta\right)+\mathbb{P}\left(\sup_{h\in\mathcal{H}}\frac{L_{D_N}(h)-L(h)}{L^q_{D_N}(h)-\delta L^p_{D_N}(h)}>\epsilon\right). \tag{A.18}$$

To bound the first probability above, we recall the definition of $L^p_{\mathcal{D}_N}(h)$ and note that $a^{\frac{1}{q}} - b^{\frac{1}{q}} \leq a - b$ if $q > 1$ and $1 \leq b \leq a$, so that

$$\mathbb{P}\left(\sup_{h\in\mathcal{H}} \frac{L^q_{\mathcal{D}_N}(h) - L^q(h)}{L^p_{\mathcal{D}_N}(h)} > \delta\right) \leq \mathbb{P}\left(\sup_{h\in\mathcal{H}} \frac{\left(L^q_{\mathcal{D}_N}(h)\right)^q - (L^q(h))^q}{N^{-\frac{1}{p}}\|\ell(\mathcal{D}_N, h)\|_p} > \delta\right). \tag{A.19}$$

Define loss function $\ell'(z,h) := (\ell(z,h))^q$, and let $L', L'^k, L'_{\mathcal{D}_N}$ and $L'^k_{\mathcal{D}_N}$ be the errors and k moments according to this new loss function. Then, the probability in (A.19) can be written as

$$\mathbb{P}\left(\sup_{h\in\mathcal{H}} \frac{L'_{\mathcal{D}_N}(h) - L'(h)}{\left(N^{-1}\|\ell'(\mathcal{D}_N, h)\|^{\frac{p}{q}}_{\frac{p}{q}}\right)^{\frac{1}{p}}} > \delta\right) \leq \mathbb{P}\left(\sup_{h\in\mathcal{H}} \frac{L'_{\mathcal{D}_N}(h) - L'(h)}{L'^q_{\mathcal{D}_N}(h)} > \delta\right) \tag{A.20}$$

since $\frac{p}{q} = q$ and $N^{\frac{1}{p}} \leq N^{\frac{1}{q}}$.

We turn to the second probability in (A.18). By applying Lemma A.11, and recalling the definition of δ, we have that $L^q_{\mathcal{D}_N}(h) - \delta L^p_{\mathcal{D}_N}(h)$ is equal to

$$L^p_{\mathcal{D}_N}(h)\left(\frac{L^q_{\mathcal{D}_N}(h)}{L^p_{\mathcal{D}_N}(h)} - \delta\right) \geq L^p_{\mathcal{D}_N}(h)\left(\frac{1}{N^{\frac{1}{q}-\frac{1}{p}}} - \frac{\epsilon}{N^{\frac{1}{q}-\frac{1}{p}}}\right) = \frac{L^p_{\mathcal{D}_N}(h)}{N^{\frac{1}{q}-\frac{1}{p}}}(1-\epsilon),$$

from which follows

$$\mathbb{P}\left(\sup_{h\in\mathcal{H}} \frac{L_{\mathcal{D}_N}(h) - L(h)}{L^q_{\mathcal{D}_N}(h) - \delta L^p_{\mathcal{D}_N}(h)} > \epsilon\right) \leq \mathbb{P}\left(\sup_{h\in\mathcal{H}} \frac{L_{\mathcal{D}_N}(h) - L(h)}{L^p_{\mathcal{D}_N}(h)} > \frac{\epsilon(1-\epsilon)}{N^{\frac{1}{q}-\frac{1}{p}}}\right). \tag{A.21}$$

The result follows by combining (A.17), (A.18), (A.20) and (A.21). □

An exponential bound for relative type I estimation error (A.16) depending on p, $\tau^\star$ and $d_{VC}(\mathcal{H})$ is a consequence of Theorems A.6 and A.12, and results in [36], which we state without a proof. Define, for a $0 < \varsigma < 1$ fixed,

$$\Gamma(p,\epsilon) = \frac{p-1}{p}(1+\varsigma)^{\frac{1}{p}} + \frac{1}{p}\left(\frac{p}{p-1}\right)^{p-1}\left(1 + \left(\frac{p-1}{p}\right)^p \varsigma^{\frac{1}{p}}\right)^{\frac{1}{p}}\left[1 + \frac{log(1/\epsilon)}{\left(\frac{p}{p-1}\right)^{p-1}}\right]^{\frac{p-1}{p}},$$

for $0 < \epsilon < 1, 1 < p \leq 2$, and

$$\Lambda(p) = \left(\frac{1}{2}\right)^{\frac{2}{p}}\left(\frac{p}{p-2}\right)^{\frac{p-1}{p}} + \frac{p}{p-1}\varsigma^{\frac{p-2}{2p}}$$

for $p > 2$.

Theorem A.13. *Fix a hypotheses space $\mathcal{H}$ and an unbounded loss function ℓ, and assume that (A.15) is in force. Then, the following holds:*

- *If P has light tails, so that (A.13) holds for a $p > 2$ fixed, then*

$$\mathbb{P}\left(\sup_{h\in\mathcal{H}} \frac{L(h) - L_{D_N}(h)}{\sqrt[p]{(L^p(h))^p + \varsigma}} > \Lambda(p)\epsilon \right) < 4\exp\left\{ d_{VC}(\mathcal{H})\left(1 + \ln\frac{2N}{d_{VC}(\mathcal{H})}\right) - \frac{\epsilon^2 N}{4} \right\}$$

and

$$\mathbb{P}\left(\sup_{h\in\mathcal{H}} \frac{L_{D_N}(h) - L(h)}{\sqrt[p]{\left(L^p_{D_N}(h)\right)^p + \varsigma}} > \Lambda(p)\epsilon \right) < 4\exp\left\{ d_{VC}(\mathcal{H})\left(1 + \ln\frac{2N}{d_{VC}(\mathcal{H})}\right) - \frac{\epsilon^2 N}{4} \right\},$$

for $0 < \epsilon < 1$ and $0 < \varsigma < \epsilon^2$.

- *If P has heavy tails, so that (A.13) holds only for a $1 < p \leq 2$ fixed, then*

$$\mathbb{P}\left(\sup_{h\in\mathcal{H}} \frac{L(h) - L_{D_N}(h)}{\sqrt[p]{(L^p(h))^p + \varsigma}} > \Gamma(p,\epsilon)\epsilon \right) < 4\exp\left\{ d_{VC}(\mathcal{H})\left(1 + \ln\frac{2N}{d_{VC}(\mathcal{H})}\right) - \frac{\epsilon^2 N^{\frac{2(p-1)}{p}}}{2^{\frac{p+2}{2}}} \right\}$$

and

$$\mathbb{P}\left(\sup_{h\in\mathcal{H}} \frac{L_{D_N}(h) - L(h)}{\sqrt[p]{\left(L^p_{D_N}(h)\right)^p + \varsigma}} > \Gamma(p,\epsilon)\epsilon \right) < 4\exp\left\{ d_{VC}(\mathcal{H})\left(1 + \ln\frac{2N}{d_{VC}(\mathcal{H})}\right) - \frac{\epsilon^2 N^{\frac{2(p-1)}{p}}}{2^{\frac{p+2}{2}}} \right\},$$

for $0 < \epsilon < 1$ and $0 < \varsigma^{\frac{p-1}{p}} < \epsilon^{\frac{p}{p-1}}$.

Theorem A.13, together with Theorem A.12, imply the following corollary, which is an exponential bound for relative type I estimation error when P has heavy or light tails. The value of ς in the definitions of $\Lambda(p)$ and $\Gamma(p,\epsilon)$ below can be arbitrarily small.

Corollary A.14. *Fix a hypotheses space $\mathcal{H}$, an unbounded loss function ℓ and $\epsilon > 0$. The following holds:*

- *If (A.13) holds for a $p \geq 4$ fixed, then*

$$\mathbb{P}\left(\sup_{h\in\mathcal{H}} \left| \frac{L(h) - L_{D_N}(h)}{L(h)} \right| > \tau^\star \Lambda(\sqrt{p})\epsilon \right) < 12\exp\left\{ d_{VC}(\mathcal{H})\left(1 + \ln\frac{2N}{d_{VC}(\mathcal{H})}\right) - \frac{\epsilon^2(1-\epsilon)^2 N^{1-\frac{2}{\sqrt{p}}+\frac{2}{p}}}{4} \right\}$$

- *If (A.13) holds for a $1 < p < 4$ fixed, then*

$$\mathbb{P}\left(\sup_{h\in\mathcal{H}} \left| \frac{L(h) - L_{D_N}(h)}{L(h)} \right| > \tau^\star \Gamma\left(\sqrt{p}, \frac{\epsilon}{N^{\frac{1}{\sqrt{p}}-\frac{1}{p}}}\right)\epsilon \right) < 12\exp\left\{ d_{VC}(\mathcal{H})\left(1 + \ln\frac{2N}{d_{VC}(\mathcal{H})}\right) - \frac{\epsilon^2 N^{\frac{2(\sqrt{p}-1)}{\sqrt{p}}-\frac{2}{\sqrt{p}}+\frac{2}{p}}}{2^{\frac{\sqrt{p}+2}{2}}} \right\}.$$

In both cases, if $d_{VC}(\mathcal{H}) < \infty$, then, by Borel-Cantelli Lemma,

$$\sup_{h\in\mathcal{H}}\left|\frac{L(h)-L_{D_N}(h)}{L(h)}\right| \xrightarrow[N\to\infty]{} 0,$$

with probability one.

Corollary A.14 establishes the convergence to zero of relative type I estimation error, and concludes our study of type I estimation error convergence in classical VC theory.

Remark A.15. *We simplified the bounds in Corollary A.14 since, by combining Theorems A.12 with A.13, we obtain a bound with three terms of different orders in N, where the exponential in each of them is multiplied by four. The term of the greatest order is that we show in Corollary A.14, with the exponential multiplied by twelve, since we can bound the two terms of lesser order by the one of the greatest order. This worsens the bound for fixed N, but ease notation and has the same qualitative effect of presenting an exponential bound for relative type I estimation error, which implies its almost sure convergence to zero.*

Remark A.16. *Condition (A.15) is not actually satisfied by many $\mathcal{H}$, for instance it does not hold for linear regression under the quadratic loss function. However, one can actually consider a $\mathcal{M} \subset \mathcal{H}$ such that (A.15) is true, with $d_{VC}(\mathcal{M}) = d_{VC}(\mathcal{H})$ and $L(h^\star) = L(h^\star_{\mathcal{M}})$, without loss of generality. For example, in linear regression one could consider only hypotheses with parameters bounded by a very large constant γ, excluding hypotheses that are unlikely to be the target one. Observe that, in this example, it is better to consider the bounds for relative type I estimation error of unbounded loss functions, rather than consider that the loss functions is bounded by a very large constant $C = \mathcal{O}(\gamma^2)$, which would generate very bad bounds when applying Corollary A.9. The results for unbounded loss functions holds for bounded ones, with p arbitrarily large.*

Remark A.17. *The main reason we assume that $\ell(z,h) \geq 1$, for all $z \in \mathcal{Z}$ and $h \in \mathcal{H}$, is to simplify the argument before (A.19), which could fail if the losses were lesser than one. We believe this assumption could be dropped at the cost of more technical results, which deviate from the main topic of this thesis. Nevertheless, the results in Corollary A.14 present an exponential bound for*

$$\mathbb{P}\left(\sup_{h\in\mathcal{H}}\left|\frac{L(h)-L_{D_N}(h)}{L(h)+1}\right| > \epsilon\right)$$

for any unbounded loss function ℓ. We note that, if we had not imposed this constraint in the loss function, we would have to deal with the denominators in the estimation errors, which could then be zero. This could have been easily accomplished by summing a constant to the denominators and then making it go to zero after the bounds are established, that is, find bounds for

$$\mathbb{P}\left(\sup_{h\in\mathcal{H}}\left|\frac{L(h)-L_{D_N}(h)}{L(h)+\varsigma}\right| > \epsilon\right),$$

and then make $\varsigma \longrightarrow 0$. This is done in [36]. Hence, by considering loss functions greater or equal to one, we have avoided the need to have heavier notations and more technical details when establishing the convergence of relative estimation errors.

A.3 Convergence to zero of type II estimation error

A bound for type II estimation error follows immediately from a bound obtained for type I estimation error. This is a consequence of the following elementary inequality, which can be found in part in [42, Lemma 8.2].

Lemma A.18. *For any hypotheses space $\mathcal{H}$ and possible sample $\mathcal{D}_N$,*

$$L(\hat{h}^{\mathcal{D}_N}) - L(h^\star) \leq 2 \sup_{h\in\mathcal{H}} |L(h) - L_{\mathcal{D}_N}(h)|, \tag{A.22}$$

and, if $\ell(z,h) \geq 1$, for all $z \in \mathcal{Z}$ and $h \in \mathcal{H}$, then

$$\frac{L(\hat{h}^{\mathcal{D}_N}) - L(h^\star)}{L(\hat{h}^{\mathcal{D}_N})} \leq 2 \sup_{h\in\mathcal{H}} \left| \frac{L(h) - L_{\mathcal{D}_N}(h)}{L(h)} \right|. \tag{A.23}$$

These inequalities yield

$$\mathbb{P}\left(L(\hat{h}^{\mathcal{D}_N}) - L(h^\star) > \epsilon \right) \leq \mathbb{P}\left(\sup_{h\in\mathcal{H}} |L(h) - L_{\mathcal{D}_N}(h)| > \epsilon/2 \right) \tag{A.24}$$

and

$$\mathbb{P}\left(\frac{L(\hat{h}^{\mathcal{D}_N}) - L(h^\star)}{L(\hat{h}^{\mathcal{D}_N})} > \epsilon \right) \leq \mathbb{P}\left(\sup_{h\in\mathcal{H}} \left| \frac{L(h) - L_{\mathcal{D}_N}(h)}{L(h)} \right| > \epsilon/2 \right). \tag{A.25}$$

Proof. The first inequality follows from

$$\begin{aligned} L(\hat{h}^{\mathcal{D}_N}) - L(h^\star) &= L(\hat{h}^{\mathcal{D}_N}) - L_{\mathcal{D}_N}(\hat{h}^{\mathcal{D}_N}) + L_{\mathcal{D}_N}(\hat{h}^{\mathcal{D}_N}) - L(h^\star) \\ &\leq L(\hat{h}^{\mathcal{D}_N}) - L_{\mathcal{D}_N}(\hat{h}^{\mathcal{D}_N}) + L_{\mathcal{D}_N}(h^\star) - L(h^\star) \\ &\leq 2 \sup_{h\in\mathcal{H}} |L(h) - L_{\mathcal{D}_N}(h)|. \end{aligned}$$

For the second one, analogous to the deduction above, we have that

$$\begin{aligned} \frac{L(\hat{h}^{\mathcal{D}_N}) - L(h^\star)}{L(\hat{h}^{\mathcal{D}_N})} &\leq \frac{L(\hat{h}^{\mathcal{D}_N}) - L_{\mathcal{D}_N}(\hat{h}^{\mathcal{D}_N})}{L(\hat{h}^{\mathcal{D}_N})} + \frac{L_{\mathcal{D}_N}(h^\star) - L(h^\star)}{L(h^\star)} \\ &\leq 2 \sup_{h\in\mathcal{H}} \left| \frac{L(h) - L_{\mathcal{D}_N}(h)}{L(h)} \right|, \end{aligned}$$

since $L(\hat{h}^{\mathcal{D}_N}) \geq L(h^\star)$. The inequalities (A.24) and (A.25) are direct from (A.22) and (A.23). □

Combining Lemma A.18 with Corollaries A.8 and A.9 we obtain the consistency of type II estimation error, when $d_{VC}(\mathcal{H}) < \infty$ and the loss function is bounded, what also concerns binary loss functions.

Corollary A.19. *Fix a hypotheses space $\mathcal{H}$ and a loss function $\ell : \mathcal{Z} \times \mathcal{H} \mapsto \mathbb{R}_+$, with $0 \leq \ell(z,h) \leq C$ for all $z \in \mathcal{Z}, h \in \mathcal{H}$. Then,*

$$\mathbb{P}\left(L(\hat{h}^{D_N}) - L(h^\star) > \epsilon\right) \leq 8 \exp\left\{d_{VC}(\mathcal{H})\left(1 + \ln\frac{N}{d_{VC}(\mathcal{H})}\right) - N\frac{\epsilon^2}{128C^2}\right\}. \tag{A.26}$$

In particular, if $d_{VC}(\mathcal{H}) < \infty$, not only (A.26) converges to zero, but also

$$L(\hat{h}^{D_N}) - L(h^\star) \xrightarrow[N\to\infty]{} 0,$$

with probability one by Borel-Cantelli Lemma.

Finally, combining Lemma A.18 with Corollary A.14, we obtain the consistency of relative type II estimation error when $d_{VC}(\mathcal{H}) < \infty$, the loss function is unbounded, and P satisfies (A.13).

Corollary A.20. *Fix a hypotheses space $\mathcal{H}$, an unbounded loss function ℓ and $\epsilon > 0$. The following holds:*

- *If (A.13) holds for a $p \geq 4$ fixed, then*

$$\mathbb{P}\left(\frac{L(\hat{h}^{D_N}) - L(h^\star)}{L(\hat{h}^{D_N})} > \tau^\star \Lambda(\sqrt{p})\epsilon\right) < 12\exp\left\{d_{VC}(\mathcal{H})\left(1 + \ln\frac{2N}{d_{VC}(\mathcal{H})}\right) - \frac{\epsilon^2(1-\epsilon/2)^2 N^{1-\frac{2}{\sqrt{p}}+\frac{2}{p}}}{16}\right\}$$

- *If (A.13) holds for a $1 < p < 4$ fixed, then*

$$\mathbb{P}\left(\frac{L(\hat{h}^{D_N}) - L(h^\star)}{L(\hat{h}^{D_N})} > \tau^\star \Gamma\left(\sqrt{p}, \frac{\epsilon}{N^{\frac{1}{\sqrt{p}}-\frac{1}{p}}}\right)\epsilon\right) < 12\exp\left\{d_{VC}(\mathcal{H})\left(1 + \ln\frac{2N}{d_{VC}(\mathcal{H})}\right) - \frac{\epsilon^2 N^{\frac{2(\sqrt{p}-1)}{\sqrt{p}}-\frac{2}{\sqrt{p}}+\frac{2}{p}}}{2^{\frac{\sqrt{p}+6}{2}}}\right\}.$$

In any case, if $d_{VC}(\mathcal{H}) < \infty$, then, by Borel-Cantelli Lemma,

$$\sup_{h\in\mathcal{H}} \frac{L(\hat{h}^{D_N}) - L(h^\star)}{L(\hat{h}^{D_N})} \xrightarrow[N\to\infty]{} 0,$$

with probability one.

This ends the study of type II estimation error convergence.

A.4 Finite VC dimension is sufficient and necessary for consistency

The results in the previous sections outline that finite VC dimension is a sufficient[2] condition for almost sure convergence to zero of type I estimation error. In this section, we outline that it may also be a necessary condition for this convergence. We consider a binary loss function and follow the reasoning of [149, Theorem 4.5].

Assuming that $d_{VC}(\mathcal{H}) = \infty$ and fixing a $\epsilon > 0$ and a $N \in \mathbb{Z}_+$, we will build a distribution P such that

$$\sup_{h \in \mathcal{H}} \left| L(h) - L_{\mathcal{D}_N}(h) \right| > 1 - \epsilon$$

with probability one. Fix a $n > N/\epsilon$ and, since $S(\mathcal{H}, n) = 2^n$, there exists a sequence

$$\mathcal{D} = \{z_1, \dots, z_n\}$$

which is shattered by the hypotheses in $\mathcal{H}$. We will consider the uniform probability measure concentrated in these points:

$$\mathbb{P}(Z = z_i) = \frac{1}{n}, i = 1, \dots, n.$$

Given a sample $\mathcal{D}_N$, consider the hypothesis h such that

$$\ell(z, h) = \begin{cases} 1, & \text{if } z \in \mathcal{D} \setminus \mathcal{D}_N \\ 0, & \text{otherwise} \end{cases},$$

which exists since $d_{VC}(\mathcal{H}) = \infty$. Observe that $|\mathcal{D} \setminus \mathcal{D}_N| \geq n - N$. In this case,

$$L_{\mathcal{D}_N}(h) = 0,$$

but

$$L(h) \geq \frac{n - N}{n} > 1 - \epsilon$$

since there are at least $n - N$ points with positive probability mass and loss one. Therefore, we conclude that, with probability one under this distribution,

$$\sup_{h \in \mathcal{H}} \left| L(h) - L_{\mathcal{D}_N}(h) \right| > 1 - \epsilon.$$

This establishes that finite VC dimension is a necessary condition for type I estimation error almost sure convergence to zero for binary loss functions.

[2] In the case of unbounded loss functions there is also a mild constraint in P.

Appendix B

Useful Mathematical concepts

In this appendix, we present some useful mathematical concepts that are referenced throughout the thesis.

B.1 Lattice theory

In this section, we present some definitions of lattice theory. We refer to [120] for an introduction to lattice theory.

Definition B.1 (Partially Ordered Set)**.** *A collection $\mathcal{A}$ is said a partially ordered set (poset) if there exists a partial relation $\leq$ satisfying, for $a, b, c \in \mathcal{A}$:*

- $a \leq a$
- $a \leq b$ *and* $b \leq a$*, then* $a = b$
- $a \leq b$ *and* $b \leq c$*, then* $a \leq c$

We denote a poset by $(\mathcal{A}, \leq)$.

Definition B.2 (Lattice)**.** *A poset $(\mathcal{A}, \leq)$ is a lattice if there exist two operations called join ($\vee$) and meet ($\wedge$) satisfying, for $a, b \in \mathcal{A}$:*

- $a \vee (a \wedge b) = a$
- $a \wedge (a \vee b) = a$

The join $a \vee b$ is the least upper bound of a and b, and the meet $a \wedge b$ is the greatest lower bound of a and b. We denote a lattice by $(\mathcal{A}, \leq, \vee, \wedge)$.

Definition B.3 (Sublattice)**.** *A sublattice $(\mathcal{A}', \leq, \vee, \wedge)$ of a lattice $(\mathcal{A}, \leq, \vee, \wedge)$ is a lattice, such that $\mathcal{A}' \subset \mathcal{A}$, which has the same partial order, join and meet as $\mathcal{A}$.*

Definition B.4 (Bounded Lattice)**.** *A lattice $(\mathcal{A}, \leq, \vee, \wedge)$ is said a bounded lattice if there exists $O, I \in \mathcal{A}$ such that*

- $O \leq a$ *for all* $a \in \mathcal{A}$
- $a \leq I$ *for all* $a \in \mathcal{A}$

The elements O and I are called, respectively, the least and greatest element of $\mathcal{A}$. We denote a bounded lattice by $(\mathcal{A}, \leq, \vee, \wedge, O, I)$.

Definition B.5 (Distributive Lattice)**.** *A lattice $(\mathcal{A}, \leq, \vee, \wedge)$ is said to be distributive if, for any $a, b, c \in \mathcal{A}$, the distributive laws*

$$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$$
$$a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$$

hold.

Definition B.6 (Complemented Lattice)**.** *A bounded lattice $(\mathcal{A}, \leq, \vee, \wedge, O, I)$ is said to be complemented if, for any $a \in \mathcal{A}$, there exists a unique $\bar{a} \in \mathcal{A}$ such that*

$$a \wedge \bar{a} = O \qquad \text{and} \qquad a \vee \bar{a} = I.$$

The element $\bar{a}$ is called the complement of a.

Definition B.7 (Complete Lattice)**.** *A lattice $(\mathcal{A}, \leq, \vee, \wedge)$ is said to be complete if every subset $\mathcal{A}' \subset \mathcal{A}$ has a join and a meet in $\mathcal{A}$, that is,*

$$\bigvee \mathcal{A}' \in \mathcal{A} \qquad \text{and} \qquad \bigwedge \mathcal{A}' \in \mathcal{A}.$$

In special, a complete lattice is a bounded lattice, and we also denote it by $(\mathcal{A}, \leq, \vee, \wedge, O, I)$.

Definition B.8 (Boolean Lattice)**.** *A lattice $(\mathcal{A}, \leq, \vee, \wedge)$ is said to be a Boolean lattice if it is a bounded, distributive and complemented lattice.*

Definition B.9 (Atoms of Boolean Lattice)**.** *The atoms of a Boolean lattice are the immediate successors of its lower bound O, that are*

$$A = \{a \in \mathcal{A} \setminus \{O\} \ : \ \text{if } b \leq a \text{ then } b = O\}.$$

Lemma B.10 (Lemma 5.3.2 of [120])**.** *A Boolean lattice $(\mathcal{A}, \leq, \vee, \wedge, O, I)$ is isomorphic to the complete lattice $(\mathcal{P}(A), \subset, \cup, \cap, \emptyset, A)$ of the powerset of its atoms. This means there is a bijection between $\mathcal{A}$ and $\mathcal{P}(A)$ that respects the partial order $\leq$ in $\mathcal{A}$ as the partial order $\subset$ in $\mathcal{P}(A)$.*

B.2 Directed acyclic graph

Definition B.11 (Directed Acyclic Graph of a poset)**.** *The directed acyclic graph of a poset $(\mathcal{A}, \leq)$ is the directed graph whose vertices are $\mathcal{A}$ and the edges connect every pair of subsequent elements, which are such that $a \leq b$ and if $a \leq c \leq b$ then either $a = c$ or $b = c$, with orientation from a to b. This graph has no cycles due to the definition of partial order.*

Definition B.12 (Distance in Directed Acyclic Graph)**.** *The distance between elements a, b in the directed acyclic graph $(\mathcal{A}, \leq)$ is the length of the shortest path from a to b.*

B.3 Hoeffding's Inequality

Theorem B.13 (Hoeffding's Inequality [69]). *Let $X_1, \dots, X_N$ be a sequence of random variables such that $-\infty < a_1 \leq X_i \leq b_i < \infty$ with probability one. Denote $S_N = \sum_{i=1}^{N} X_i$. Then, for any $t > 0$,*

$$\mathbb{P}\left(|S_N - \mathbb{E}(S_N)| > t\right) \leq 2\exp\left\{-\frac{2t^2}{\sum_{i=1}^{N}(b_i - a_i)^2}\right\}.$$

B.4 Borel-Cantelli Lemma

Lemma B.14 (Borel-Cantelli Lemma). *Let $(\Omega, \mathcal{S}, \mathbb{P})$ be a probability space, and $\{A_n\}$ be a sequence of events. If*

$$\sum_{n=1}^{\infty} \mathbb{P}(A_n) < \infty,$$

then

$$\mathbb{P}\left(\lim_{n\to\infty} \bigcap_{n=1}^{\infty} \bigcup_{k=n}^{\infty} A_k\right) = 0.$$

Corollary B.15. *Let $\{X_n\}$ be a sequence of non-negative random variables. If*

$$\sum_{n=1}^{\infty} \mathbb{P}(X_n > \epsilon) < \infty,$$

for all $\epsilon > 0$ fixed, then

$$\mathbb{P}\left(\lim_{n\to\infty} X_n = 0\right) = 1.$$

References

[1] Yaser S Abu-Mostafa, Malik Magdon-Ismail, and Hsuan-Tien Lin. *Learning from data*. Vol. 4. AMLBook New York, NY, USA: 2012 (cit. on pp. 3, 5, 9).

[2] Terrence M Adams and Andrew B Nobel. "Uniform convergence of Vapnik–Chervonenkis classes under ergodic sampling". In: *The Annals of Probability* 38.4 (2010), pp. 1345–1367 (cit. on p. 111).

[3] Charu C Aggarwal et al. "Neural networks and deep learning". In: *Springer* 10 (2018), pp. 978–3 (cit. on pp. 18, 125).

[4] Mouad MH Ali et al. "Overview of fingerprint recognition system". In: *2016 international conference on electrical, electronics, and optimization techniques (ICEEOT)*. IEEE. 2016, pp. 1334–1338 (cit. on p. 127).

[5] Davide Anguita et al. "In-sample and out-of-sample model selection and error estimation for support vector machines". In: *IEEE Transactions on Neural Networks and Learning Systems* 23.9 (2012), pp. 1390–1406 (cit. on p. 9).

[6] Sylvain Arlot and Peter L Bartlett. "Margin-adaptive model selection in statistical learning". In: *Bernoulli* 17.2 (2011), pp. 687–713 (cit. on pp. 10, 139).

[7] Sylvain Arlot, Alain Celisse, et al. "A survey of cross-validation procedures for model selection". In: *Statistics surveys* 4 (2010), pp. 40–79 (cit. on p. 29).

[8] Esmaeil Atashpaz-Gargari et al. "A fast Branch-and-Bound algorithm for U-curve feature selection". In: *Pattern Recognition* 73 (2018), pp. 172–188 (cit. on pp. 3, 14, 44, 82, 84, 94).

[9] Alan Baker. "Simplicity". In: *Stanford Encyclopedia of Philosophy* (2004) (cit. on p. 46).

[10] Halgurt Bapierre, Georg Groh, and Stefan Theiner. "A variable order markov model approach for mobility prediction". In: *Pervasive Computing* (2011), pp. 8–16 (cit. on p. 110).

[11] Junior Barrera, Edward R Dougherty, and Nina Sumiko Tomita. "Automatic programming of binary morphological machines by design of statistically optimal operators in the context of computational learning theory". In: *Journal of Electronic Imaging* 6.1 (1997), pp. 54–67 (cit. on pp. 17, 119, 121).

[12] Junior Barrera and Guillermo Pablo Salas. "Set operations on closed intervals and their applications to the automatic programming of morphological machines". In: *Journal of Electronic Imaging* 5.3 (1996), pp. 335–352 (cit. on pp. 119, 121).

[13] Junior Barrera et al. "Automatic programming of morphological machines by PAC learning". In: *Fundamenta Informaticae* 41.1, 2 (2000), pp. 229–258 (cit. on pp. 119, 121).

[14] Junior Barrera et al. “Constructing probabilistic genetic networks of Plasmodium falciparum from dynamical expression signals of the intraerythrocytic development cycle”. In: *Methods of Microarray Data Analysis V*. Springer, 2007, pp. 11–26 (cit. on p. 17).

[15] Junior Barrera et al. “From Mathematical Morphology to machine learning of image operators”. In: *São Paulo Journal of Mathematical Sciences* (2022), pp. 1–42 (cit. on pp. 4, 120).

[16] Peter L Bartlett. “Fast rates for estimation error and oracle inequalities for model selection”. In: *Econometric Theory* 24.2 (2008), pp. 545–552 (cit. on pp. 10, 139).

[17] Peter L Bartlett et al. “Nearly-tight VC-dimension and Pseudodimension Bounds for Piecewise Linear Neural Networks.” In: *Journal of Machine Learning Research* 20.63 (2019), pp. 1–17 (cit. on p. 39).

[18] HW Becker and John Riordan. “The arithmetic of Bell and Stirling numbers”. In: *American journal of Mathematics* 70.2 (1948), pp. 385–394 (cit. on p. 86).

[19] Ron Begleiter, Ran El-Yaniv, and Golan Yona. “On prediction using variable order Markov models”. In: *Journal of Artificial Intelligence Research* 22 (2004), pp. 385–421 (cit. on p. 108).

[20] Mikhail Belkin et al. “Reconciling modern machine-learning practice and the classical bias–variance trade-off”. In: *Proceedings of the National Academy of Sciences* 116.32 (2019), pp. 15849–15854 (cit. on p. 13).

[21] Eric Temple Bell. “Exponential polynomials”. In: *Annals of Mathematics* (1934), pp. 258–277 (cit. on p. 86).

[22] Eric Temple Bell. “The iterated exponential integers”. In: *Annals of Mathematics* (1938), pp. 539–557 (cit. on p. 86).

[23] Shai Bendavid et al. “Characterizations of Learnability for Classes of (0,..., n)-Valued Functions”. In: *Journal of Computer and System Sciences* 50.1 (1995), pp. 74–86 (cit. on p. 22).

[24] Peter J Bickel et al. “Regularization in statistics”. In: *Test* 15.2 (2006), pp. 271–344 (cit. on p. 11).

[25] Patrick Billingsley. *Probability and measure*. John Wiley & Sons, 2008 (cit. on p. 144).

[26] Rainer Böhme et al. “Bitcoin: Economics, technology, and governance”. In: *Journal of economic Perspectives* 29.2 (2015), pp. 213–38 (cit. on p. 114).

[27] George EP Box et al. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015 (cit. on p. 108).

[28] Andrej Bratko et al. “Spam filtering using statistical data compression models”. In: *The Journal of Machine Learning Research* 7 (2006), pp. 2673–2698 (cit. on p. 110).

[29] Marcel Brun et al. “Design of optimal binary filters under joint multiresolution–envelope constraint”. In: *Pattern recognition letters* 24.7 (2003), pp. 937–945 (cit. on pp. 119, 121).

[30] Marcel Brun et al. “Nonlinear filter design using envelopes”. In: *Journal of Mathematical Imaging and Vision* 21.1 (2004), pp. 81–97 (cit. on pp. 119, 121).

[31] Randal E Bryant. “Graph-based algorithms for boolean function manipulation”. In: *Computers, IEEE Transactions on* 100.8 (1986), pp. 677–691 (cit. on p. 98).

[32] Joel Edu Sánchez Castro. “Model Selection for Learning Boolean Hypothesis”. PhD thesis. Universidade de São Paulo, 2018 (cit. on pp. 3, 9, 35, 94, 98, 106).

[33] Gavin C Cawley and Nicola LC Talbot. "On over-fitting in model selection and subsequent selection bias in performance evaluation". In: *The Journal of Machine Learning Research* 11 (2010), pp. 2079–2107 (cit. on p. 69).

[34] Ondřej Čepek, David Kronus, and Petr Kučera. "Recognition of interval Boolean functions". In: *Annals of Mathematics and Artificial Intelligence* 52.1 (2008), pp. 1–24 (cit. on pp. 127, 133).

[35] Dan Ciregan, Ueli Meier, and Jürgen Schmidhuber. "Multi-column deep neural networks for image classification". In: *2012 IEEE conference on computer vision and pattern recognition*. IEEE. 2012, pp. 3642–3649 (cit. on p. 119).

[36] Corinna Cortes, Spencer Greenberg, and Mehryar Mohri. "Relative deviation learning bounds and generalization with unbounded loss functions". In: *Annals of Mathematics and Artificial Intelligence* 85.1 (2019), pp. 45–70 (cit. on pp. 143, 155, 157).

[37] Corinna Cortes and Vladimir Vapnik. "Support-vector networks". In: *Machine learning* 20.3 (1995), pp. 273–297 (cit. on pp. 8, 22).

[38] Thomas W Cusick and Pantelimon Stanica. *Cryptographic Boolean functions and applications*. Academic Press, 2017 (cit. on p. 17).

[39] George Cybenko. "Approximation by superpositions of a sigmoidal function". In: *Mathematics of control, signals and systems* 2.4 (1989), pp. 303–314 (cit. on p. 141).

[40] Li Deng and Xiao Li. "Machine learning paradigms for speech recognition: An overview". In: *IEEE Transactions on Audio, Speech, and Language Processing* 21.5 (2013), pp. 1060–1089 (cit. on p. 108).

[41] Misha Denil et al. "Predicting parameters in deep learning". In: *Proceedings of the 26th International Conference on Neural Information Processing Systems-Volume 2*. 2013, pp. 2148–2156 (cit. on pp. 39, 140).

[42] Luc Devroye, László Györfi, and Gábor Lugosi. *A probabilistic theory of pattern recognition*. Vol. 31. Springer, 1996 (cit. on pp. 12, 24, 139, 143, 146, 158).

[43] Thomas G Dietterich. "Machine learning for sequential data: A review". In: *Joint IAPR international workshops on statistical techniques in pattern recognition (SPR) and structural and syntactic pattern recognition (SSPR)*. Springer. 2002, pp. 15–30 (cit. on p. 108).

[44] Jie Ding, Vahid Tarokh, and Yuhong Yang. "Model selection techniques: An overview". In: *IEEE Signal Processing Magazine* 35.6 (2018), pp. 16–34 (cit. on p. 9).

[45] Pedro Domingos. "The role of Occam's razor in knowledge discovery". In: *Data mining and knowledge discovery* 3.4 (1999), pp. 409–425 (cit. on pp. 13, 81).

[46] Marta M Dornelles and Nina ST Hirata. "Selection of windows for w-operator combination from entropy based ranking". In: *2015 28th SIBGRAPI Conference on Graphics, Patterns and Images*. IEEE. 2015, pp. 64–71 (cit. on pp. 119, 121).

[47] Edward Dougherty. *Mathematical morphology in image processing*. Vol. 1. CRC press, 2018 (cit. on p. 120).

[48] Edward R Dougherty and Roberto A Lotufo. *Hands-on morphological image processing*. Vol. 59. SPIE press, 2003 (cit. on pp. 119, 121).

[49] Edward R Dougherty et al. "Multiresolution analysis for optimal binary filters". In: *Journal of Mathematical Imaging and Vision* 14.1 (2001), pp. 53–72 (cit. on pp. 17, 119, 121).

[50] Edward R Dougherty et al. "Performance of error estimators for classification". In: *Current Bioinformatics* 5.1 (2010), pp. 53–67 (cit. on p. 46).

[51] B Efron. "Bootstrap Methods: Another Look at the Jackknife". In: *The Annals of Statistics* (1979), pp. 1–26 (cit. on p. 27).

[52] Bradley Efron. "Estimating the error rate of a prediction rule: improvement on cross-validation". In: *Journal of the American statistical association* 78.382 (1983), pp. 316–331 (cit. on p. 27).

[53] Bradley Efron and Robert Tibshirani. "Improvements on cross-validation: the 632+ bootstrap method". In: *Journal of the American Statistical Association* 92.438 (1997), pp. 548–560 (cit. on p. 27).

[54] Thomas Elsken, Jan Hendrik Metzen, Frank Hutter, et al. "Neural architecture search: A survey." In: *Journal of Machine Learning Research* 20.55 (2019), pp. 1–21 (cit. on p. 39).

[55] Gustavo Estrela et al. "An efficient, parallelized algorithm for optimal conditional entropy-based feature selection". In: *Entropy* 22.4 (2020), p. 492 (cit. on pp. 14, 44, 82, 84, 94, 98).

[56] Ian Fischer. "The conditional entropy bottleneck". In: *Entropy* 22.9 (2020), p. 999 (cit. on p. 105).

[57] Sergey Foss, Dmitry Korshunov, Stan Zachary, et al. *An introduction to heavy-tailed and subexponential distributions*. Vol. 6. Springer, 2011 (cit. on p. 70).

[58] Jerome H Friedman. "On bias, variance, 0/1—loss, and the curse-of-dimensionality". In: *Data mining and knowledge discovery* 1.1 (1997), pp. 55–77 (cit. on p. 13).

[59] Giorgio Gnecco and Marcello Sanguineti. "Approximation Error Bounds via Rademacher's Complexity". In: *Applied Mathematical Sciences* 2.4 (2008), pp. 153–176 (cit. on pp. 31, 67).

[60] Isabelle Guyon and André Elisseeff. "An introduction to variable and feature selection". In: *Journal of machine learning research* 3.Mar (2003), pp. 1157–1182 (cit. on p. 10).

[61] Isabelle Guyon et al. "Model selection: beyond the bayesian/frequentist divide." In: *Journal of Machine Learning Research* 11.1 (2010) (cit. on p. 9).

[62] James Douglas Hamilton. *Time series analysis*. Princeton university press, 2020 (cit. on p. 108).

[63] Boris Hanin. "Universal function approximation by deep neural nets with bounded width and relu activations". In: *Mathematics* 7.10 (2019), p. 992 (cit. on p. 141).

[64] Boris Hanin and Mark Sellke. "Approximating continuous functions by relu nets of minimal width". In: *arXiv preprint arXiv:1710.11278* (2017) (cit. on p. 141).

[65] Nick Harvey, Christopher Liaw, and Abbas Mehrabian. "Nearly-tight VC-dimension bounds for piecewise linear neural networks". In: *Conference on Learning Theory*. PMLR. 2017, pp. 1064–1068 (cit. on p. 39).

[66] Andreas Heinecke, Jinn Ho, and Wen-Liang Hwang. "Refinement and universal approximation via sparsely connected ReLU convolution nets". In: *IEEE Signal Processing Letters* 27 (2020), pp. 1175–1179 (cit. on p. 141).

[67] Nina ST Hirata. "Multilevel training of binary morphological operators". In: *IEEE Transactions on pattern analysis and machine intelligence* 31.4 (2008), pp. 707–720 (cit. on pp. 119, 121).

[68] Roberto Hirata Junior et al. "Multiresolution design of aperture operators". In: *Journal of Mathematical Imaging and Vision* 16.3 (2002), pp. 199–222 (cit. on pp. 119, 121).

[69] Wassily Hoeffding. "Probability inequalities for sums of bounded random variables". In: *Journal of the American statistical association* 58.301 (1963), pp. 13–30 (cit. on pp. 145, 163).

[70] KiHoon Hong. "Bitcoin as an alternative investment vehicle". In: *Information Technology and Management* 18.4 (2017), pp. 265–275 (cit. on p. 114).

[71] Kurt Hornik. "Approximation capabilities of multilayer feedforward networks". In: *Neural networks* 4.2 (1991), pp. 251–257 (cit. on p. 141).

[72] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. "Multilayer feedforward networks are universal approximators". In: *Neural networks* 2.5 (1989), pp. 359–366 (cit. on p. 141).

[73] Gordon Hughes. "On the mean accuracy of statistical pattern recognizers". In: *IEEE transactions on information theory* 14.1 (1968), pp. 55–63 (cit. on p. 82).

[74] Anil K Jain, Robert P. W. Duin, and Jianchang Mao. "Statistical pattern recognition: A review". In: *IEEE Transactions on pattern analysis and machine intelligence* 22.1 (2000), pp. 4–37 (cit. on p. 82).

[75] Katarzyna Janocha and Wojciech Czarnecki. "On Loss Functions for Deep Neural Networks in Classification". In: *Schedae Informaticae* 25 (Feb. 2017) (cit. on p. 19).

[76] Kevin Jarrett et al. "What is the best multi-stage architecture for object recognition?" In: *2009 IEEE 12th international conference on computer vision*. IEEE. 2009, pp. 2146–2153 (cit. on p. 119).

[77] George H John, Ron Kohavi, and Karl Pfleger. "Irrelevant features and the subset selection problem". In: *Machine learning: proceedings of the eleventh international conference*. 1994, pp. 121–129 (cit. on p. 10).

[78] Jesse Johnson. "Deep, skinny neural networks are not universal approximators". In: *arXiv preprint arXiv:1810.00393* (2018) (cit. on p. 141).

[79] Marek Karpinski and Angus Macintyre. "Polynomial bounds for VC dimension of sigmoidal and general Pfaffian neural networks". In: *Journal of Computer and System Sciences* 54.1 (1997), pp. 169–176 (cit. on p. 39).

[80] Patrick Kidger and Terry Lyons. "Universal approximation with deep narrow networks". In: *Conference on learning theory*. PMLR. 2020, pp. 2306–2327 (cit. on p. 141).

[81] Ron Kohavi et al. "A study of cross-validation and bootstrap for accuracy estimation and model selection". In: *Ijcai*. Vol. 14. 2. Montreal, Canada. 1995, pp. 1137–1145 (cit. on p. 27).

[82] Vladimir Koltchinskii. *Oracle Inequalities in Empirical Risk Minimization and Sparse Recovery Problems: Ecole d'Eté de Probabilités de Saint-Flour XXXVIII-2008*. Vol. 2033. Springer Science & Business Media, 2011 (cit. on pp. 10, 139).

[83] Vladimir Koltchinskii. "Rademacher penalties and structural risk minimization". In: *IEEE Transactions on Information Theory* 47.5 (2001), pp. 1902–1914 (cit. on pp. 10, 139).

[84] Yassin Kortli et al. "Face recognition systems: A survey". In: *Sensors* 20.2 (2020), p. 342 (cit. on p. 127).

[85] Bjoern Krollner, Bruce J Vanstone, Gavin R Finnie, et al. "Financial time series forecasting with machine learning techniques: a survey." In: *ESANN*. 2010 (cit. on p. 108).

[86] David Kronus. "Interval Representations of Boolean Functions". PhD thesis. Charles University in Prague, 2007 (cit. on pp. 127, 133).

[87] Solomon Kullback and Richard A Leibler. "On information and sufficiency". In: *The annals of mathematical statistics* 22.1 (1951), pp. 79–86 (cit. on p. 20).

[88] Fabien Lauer, Ching Y Suen, and Gérard Bloch. "A trainable feature extractor for handwritten digit recognition". In: *Pattern Recognition* 40.6 (2007), pp. 1816–1824 (cit. on p. 119).

[89] Yann LeCun, Corinna Cortes, and CJ Burges. "MNIST handwritten digit database". In: *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist* 2 (2010) (cit. on pp. 4, 99, 101, 117, 118).

[90] Yann LeCun et al. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324 (cit. on p. 119).

[91] Moshe Leshno et al. "Multilayer feedforward networks with a nonpolynomial activation function can approximate any function". In: *Neural networks* 6.6 (1993), pp. 861–867 (cit. on p. 141).

[92] Hongzhou Lin and Stefanie Jegelka. "Resnet with one-neuron hidden layers is a universal approximator". In: *Advances in neural information processing systems* 31 (2018) (cit. on p. 141).

[93] Zhou Lu et al. "The expressive power of neural networks: A view from the width". In: *Advances in neural information processing systems* 30 (2017) (cit. on p. 141).

[94] David JC MacKay, David JC Mac Kay, et al. *Information theory, inference and learning algorithms*. Cambridge university press, 2003 (cit. on p. 102).

[95] Anthony J Macula. "Covers of a finite set". In: *Mathematics Magazine* 67.2 (1994), pp. 141–144 (cit. on p. 5).

[96] Joshua Magarick. "Sequential learning and variable length Markov chains". PhD thesis. University of Pennsylvania, 2016 (cit. on p. 108).

[97] Thomas Marill and D Green. "On the effectiveness of receptors in recognition systems". In: *IEEE transactions on Information Theory* 9.1 (1963), pp. 11–17 (cit. on p. 82).

[98] David C Martins, Roberto M Cesar, and Junior Barrera. "W-operator window design by minimization of mean conditional entropy". In: *Pattern analysis and applications* 9.2 (2006), pp. 139–153 (cit. on pp. 119, 121).

[99] Pascal Massart. *Concentration inequalities and model selection: Ecole d'Eté de Probabilités de Saint-Flour XXXIII-2003*. Springer, 2007 (cit. on pp. 3, 9–11, 138, 139).

[100] Georges Matheron. *Random sets and integral geometry*. John Wiley & Sons, 1974 (cit. on p. 120).

[101] Geoffrey J McLachlan. *Discriminant analysis and statistical pattern recognition*. John Wiley & Sons, 2005 (cit. on p. 82).

[102] Charles A Micchelli, Massimiliano Pontil, and Peter Bartlett. "Learning the Kernel Function via Regularization." In: *Journal of machine learning research* 6.7 (2005) (cit. on p. 11).

[103] Annette M Molinaro, Richard Simon, and Ruth M Pfeiffer. "Prediction error estimation: a comparison of resampling methods". In: *Bioinformatics* 21.15 (2005), pp. 3301–3307 (cit. on p. 28).

[104] Ahmed Afif Monrat, Olov Schelén, and Karl Andersson. "A survey of blockchain from the perspectives of applications, challenges, and opportunities". In: *IEEE Access* 7 (2019), pp. 117134–117151 (cit. on p. 114).

[105] Laurent Najman and Hugues Talbot. *Mathematical morphology: from theory to applications*. John Wiley & Sons, 2013 (cit. on p. 120).

[106] Patrenahalli M. Narendra and Keinosuke Fukunaga. "A branch and bound algorithm for feature subset selection". In: *IEEE Transactions on computers* 26.09 (1977), pp. 917–922 (cit. on p. 82).

[107] Balas K Natarajan. "On learning sets and functions". In: *Machine Learning* 4.1 (1989), pp. 67–97 (cit. on p. 22).

[108] John Ashworth Nelder and Robert WM Wedderburn. "Generalized linear models". In: *Journal of the Royal Statistical Society: Series A (General)* 135.3 (1972), pp. 370–384 (cit. on p. 16).

[109] John Neter et al. *Applied linear statistical models*. 3rd ed. Irwin Chicago, 1990 (cit. on p. 21).

[110] Ulisses M. Braga Neto and Edward R. Dougherty. *Error Estimation for Pattern Recognition*. Wiley, 2015 (cit. on pp. 27, 29).

[111] Behnam Neyshabur et al. "The role of over-parametrization in generalization of neural networks". In: *International Conference on Learning Representations*. 2018 (cit. on pp. 39, 140).

[112] Andrew Y Ng. "Feature selection, L 1 vs. L 2 regularization, and rotational invariance". In: *Proceedings of the twenty-first international conference on Machine learning*. 2004, p. 78 (cit. on p. 11).

[113] Luca Oneto, Sandro Ridella, and Davide Anguita. "Tikhonov, Ivanov and Morozov regularization for support vector machine learning". In: *Machine Learning* 103.1 (2016), pp. 103–136 (cit. on p. 11).

[114] Overfitting. *Oxford Dictionaries Online*. 2020. URL: https://www.lexico.com/definition/overfitting (cit. on p. 1).

[115] Jayashree Padmanabhan and Melvin Jose Johnson Premkumar. "Machine learning in automatic speech recognition: A survey". In: *IETE Technical Review* 32.4 (2015), pp. 240–251 (cit. on p. 108).

[116] Jooyoung Park and Irwin W Sandberg. "Universal approximation using radial-basis-function networks". In: *Neural computation* 3.2 (1991), pp. 246–257 (cit. on p. 141).

[117] Sejun Park et al. "Minimum width for universal approximation". In: *arXiv preprint arXiv:2006.08859* (2020) (cit. on p. 141).

[118] Allan Pinkus. "Approximation theory of the MLP model in neural networks". In: *Acta numerica* 8 (1999), pp. 143–195 (cit. on p. 141).

[119] Karl Popper. *The logic of scientific discovery*. Routledge, 2005 (cit. on p. 1).

[120] Franco P Preparata and Raymond Tzuu-Yau Yeh. *Introduction to discrete structures for computer science and engineering*. Addison-Wesley Longman Publishing Co., Inc., 1973 (cit. on pp. 161, 162).

[121] Pavel Pudil, Jana Novovičová, and Josef Kittler. "Floating search methods in feature selection". In: *Pattern recognition letters* 15.11 (1994), pp. 1119–1125 (cit. on p. 82).

[122] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria, 2016 (cit. on p. 101).

[123] Marc'Aurelio Ranzato et al. "Efficient learning of sparse representations with an energy-based model". In: *Advances in neural information processing systems* 19 (2006) (cit. on p. 119).

[124] Marc'Aurelio Ranzato et al. "Unsupervised learning of invariant feature hierarchies with applications to object recognition". In: *2007 IEEE conference on computer vision and pattern recognition*. IEEE. 2007, pp. 1–8 (cit. on p. 119).

[125] Sebastian Raschka. "Model evaluation, model selection, and algorithm selection in machine learning". In: *arXiv preprint arXiv:1811.12808* (2018) (cit. on p. 9).

[126] Sarunas Raudys and Vitalijus Pikelis. "On dimensionality, sample size, classification error, and complexity of classification algorithm in pattern recognition". In: *IEEE Transactions on pattern analysis and machine intelligence* 3 (1980), pp. 242–252 (cit. on pp. 13, 82).

[127] Sarunas J Raudys, Anil K Jain, et al. "Small sample size effects in statistical pattern recognition: Recommendations for practitioners". In: *IEEE Transactions on pattern analysis and machine intelligence* 13.3 (1991), pp. 252–264 (cit. on p. 82).

[128] Marcelo S Reis. "Minimization of decomposable in U-shaped curves functions defined on poset chains–algorithms and applications". PhD thesis. Institute of Mathematics and Statistics, University of Sao Paulo, Brazil (in Portuguese), 2012 (cit. on pp. 3, 82, 84, 94).

[129] Marcelo S Reis and Junior Barrera. "Solving problems in mathematical morphology through reductions to the U-curve problem". In: *International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing*. Springer. 2013, pp. 49–60 (cit. on pp. 119, 121).

[130] Marcelo S Reis et al. "featsel: A framework for benchmarking of feature selection algorithms and cost functions". In: *SoftwareX* 6 (2017), pp. 193–197 (cit. on pp. 3, 14, 82, 84, 94).

[131] Marcelo S Reis et al. "Optimal Boolean lattice-based algorithms for the U-curve optimization problem". In: *Information Sciences* (2018) (cit. on pp. 3, 14, 44, 82, 84, 94).

[132] Robert A Rigby and D Mikis Stasinopoulos. "Generalized additive models for location, scale and shape". In: *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 54.3 (2005), pp. 507–554 (cit. on p. 16).

[133] Marcelo Ris, Junior Barrera, and David C Martins. "U-curve: A branch-and-bound optimization algorithm for U-shaped cost functions on Boolean lattices applied to the feature selection problem". In: *Pattern Recognition* 43.3 (2010), pp. 557–568 (cit. on pp. 3, 14, 44, 82, 84, 94).

[134] Jorma Rissanen. "A universal data compression system". In: *IEEE Transactions on information theory* 29.5 (1983), pp. 656–664 (cit. on p. 110).

[135] Frank Rosenblatt. *The perceptron, a perceiving and recognizing automaton*. Cornell Aeronautical Laboratory, 1957 (cit. on pp. 7, 18).

[136] Richard J Rossi. *Mathematical statistics: an introduction to likelihood based inference*. John Wiley & Sons, 2018 (cit. on pp. 16, 21, 32).

[137] Sebastian Ruder. "An overview of gradient descent optimization algorithms". In: *arXiv preprint arXiv:1609.04747* (2016) (cit. on pp. 121, 125).

[138] Syed Jawad Hussain Shahzad et al. "Is Bitcoin a better safe-haven investment than gold and commodities?" In: *International Review of Financial Analysis* 63 (2019), pp. 322–330 (cit. on p. 114).

[139] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014 (cit. on pp. 7, 22, 51).

[140] Wojciech Siedlecki and Jack Sklansky. "A note on genetic algorithms for large-scale feature selection". In: *Handbook of pattern recognition and computer vision*. World Scientific, 1993, pp. 88–107 (cit. on p. 82).

[141] Wojciech Siedlecki and Jack Sklansky. "On automatic feature selection". In: *Handbook of pattern recognition and computer vision*. World Scientific, 1993, pp. 63–87 (cit. on p. 82).

[142] Pierre Soille. *Morphological image analysis: principles and applications*. Springer Science & Business Media, 2013 (cit. on p. 120).

[143] Petr Somol et al. "Adaptive floating search methods in feature selection". In: *Pattern recognition letters* 20.11-13 (1999), pp. 1157–1163 (cit. on p. 82).

[144] Eduardo D Sontag. "VC dimension of neural networks". In: *NATO ASI Series F Computer and Systems Sciences* 168 (1998), pp. 69–96 (cit. on pp. 39, 140).

[145] Mervyn Stone. "Cross-validatory choice and assessment of statistical predictions". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 36.2 (1974), pp. 111–133 (cit. on p. 29).

[146] William M Thorburn. "The myth of Occam's razor". In: *Mind* 27.107 (1918), pp. 345–353 (cit. on p. 81).

[147] Yingjie Tian and Yuqi Zhang. "A comprehensive survey on regularization strategies in machine learning". In: *Information Fusion* 80 (2022), pp. 146–166 (cit. on p. 11).

[148] Leslie G Valiant. "A theory of the learnable". In: *Communications of the ACM* 27.11 (1984), pp. 1134–1142 (cit. on pp. 3, 7, 50).

[149] Vladimir Vapnik. *Statistical learning theory. 1998*. Vol. 3. Wiley, New York, 1998 (cit. on pp. 1, 7, 12, 70, 71, 143, 144, 149, 151, 152, 160).

[150] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2000 (cit. on pp. 1, 7, 9, 20, 22, 143).

[151] Vladimir Vapnik and Alexey Chervonenkis. *Theory of pattern recognition*. 1974 (cit. on p. 7).

[152] Vladimir N Vapnik and Alexey Ya Chervonenkis. "On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities". In: *Theory of Probability and its Applications* 16 (1971), pp. 264–280 (cit. on pp. 7, 148).

[153] Vladimir N Vapnik and Alexey Ya Chervonenkis. "Oordered Risk Minimization II". In: *Automation and Remote Control* 35.9 (1974), pp. 1403–1412 (cit. on p. 7).

[154] Vladimir N Vapnik and Alexey Ya Chervonenkis. "Ordered Risk Minimization I". In: *Automation and Remote Control* 35.8 (1974), pp. 1226–1235 (cit. on p. 7).

[155] A Wayne Whitney. "A direct method of nonparametric measurement selection". In: *IEEE transactions on computers* 100.9 (1971), pp. 1100–1103 (cit. on p. 82).

[156] Chuan-Kun Wu, Dengguo Feng, et al. *Boolean functions and their applications in cryptography*. Springer, 2016 (cit. on p. 17).

[157] Keyulu Xu et al. "How powerful are graph neural networks?" In: *arXiv preprint arXiv:1810.00826* (2018) (cit. on p. 141).

[158] Jie Yang et al. “Predicting next location using a variable order Markov model”. In: *Proceedings of the 5th ACM SIGSPATIAL International Workshop on GeoStreaming*. 2014, pp. 37–42 (cit. on p. 110).

[159] Dmitry Yarotsky. “Universal approximations of invariant maps by neural networks”. In: *Constructive Approximation* 55.1 (2022), pp. 407–474 (cit. on p. 141).

[160] Hyeon-Joong Yoo. “Deep convolution neural networks in computer vision: a review”. In: *IEIE Transactions on Smart Processing and Computing* 4.1 (2015), pp. 35–43 (cit. on p. 121).

[161] Ding-Xuan Zhou. “Universality of deep convolutional neural networks”. In: *Applied and computational harmonic analysis* 48.2 (2020), pp. 787–794 (cit. on p. 141).

[162] Amin Zollanvari, Alex Pappachen James, and Reza Sameni. “A theoretical analysis of the peaking phenomenon in classification”. In: *Journal of Classification* 37.2 (2020), pp. 421–434 (cit. on pp. 13, 82).

Index

activation function, 18

Bell number, 86
Boolean functions, 17, 35
Boolean partition lattice, 35

Conditional Entropy, 102
consistency of Model Selection, 49, 72
continuous chain, 40
cross-validation, 27

Empirical Risk Minimization, 7, 20, 21, 27, 28, 43, 44, 50, 54, 59, 65, 103, 113, 124
entropy, 21
equivalence of hypotheses spaces, 15
estimation errors, 12, 42, 47, 50, 59, 145, 158

functional hypotheses spaces, 15, 17, 19, 21, 23, 26, 32, 119, 145

Generalized Additive Models for Location Scale and Shape, 16
Generalized Linear Models, 16
global minimum, 41
global minimum of a continuous chain, 41

hypotheses space, 5, 14

in-sample error, 7, 15
inf-strong local minimum, 40
inf-weak U-curve property, 83
interval Boolean functions, 131

Kullback–Leibler divergence, 20

lattice gradient descent algorithm, 125
Lattice Learning Spaces, 30
learning by reusing, 46
Learning Space generator, 32
learning with independent sample, 47
least squares method, 21
linear classifiers, 18, 38
link function, 16
loss function, 14

Machine Learning, 6
Maximum Likelihood, 16, 20, 21, 32
model, 5, 14
Model Selection, 9
multilayer W-operator, 121

Neural Architecture Search, 39
neural network, 18, 38

Occam's razor, 13, 81
out-of-sample error, 7, 14
overfitting, 1

PAC-learning, 50
parametric poset, 32
Partition Lattice Learning Space, 34, 38, 58, 65, 81, 86, 92, 98, 102, 108, 110, 127
partition-hypothesis duality, 33
peaking phenomenon, 13, 82

quadratic loss function, 7, 17

regression, 16, 21
regularization, 11
relative estimation errors, 71, 153

shatter coefficient, 22, 146
simple loss function, 7, 17

strong local minimum, 40
strong U-curve property, 83
Structured Risk Minimization, 9
sup-strong local minimum, 40
sup-weak U-curve property, 83, 86, 89, 133
Support Vector Machine, 8, 22

target hypotheses, 7, 19
target model, 13, 41

U-curve algorithm, 94, 97, 112, 124, 126
U-curve property, 14, 83
unbounded loss function, 70, 151

validation sample, 27
variable order Markov chain, 108
variable selection, 33
VC dimension, 7, 22, 23, 30, 41, 146, 160
VC theory, 8, 43, 45, 50, 143, 157

W-operator, 119
weak local minimum, 40
weak U-curve property, 83
weight of distribution tail, 70, 151