Covariate Shift Adaptation and Dataset Shift Decomposition in Machine Learning

Felipe Maia Polo

Dissertation Presented
to the
Institute of Mathematics and Statistics
of the
University of São Paulo
to
Obtain the Degree
of
Master of Science

Program: Statistics

Supervisor: Prof. Dr. Renato Vicente

During the development of this work, the author received financial support from the Brazilian National Council for Scientific and Technological Development (CNPq) (process 132857/2019-7) during the first two years and from Fundunesp and the Advanced Institute for Artificial Intelligence (AI2) (processo 3061/2019-CCP) for the final months.

São Paulo, December of 2021

Adaptação para *Covariate Shift* e decomposição do *Dataset Shift* no Aprendizado de Máquina

Felipe Maia Polo

Dissertação apresentada
ao
Instituto de Matemática e Estatística
da
Universidade de São Paulo
para
Obtenção do Título
de
Mestre em Ciências

Programa: Estatística

Orientador: Prof. Dr. Renato Vicente

Durante o desenvolvimento deste trabalho, o autor recebeu auxílios financeiros do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) (processo 132857/2019-7) durante os dois primeiros anos e da Fundunesp e do *Advanced Institute for Artificial Intelligence (AI2)* (process 3061/2019-CCP) durante os meses finais.

São Paulo, Dezembro de 2021

Adaptação para *Covariate Shift* e decomposição do *Dataset Shift* no Aprendizado de Máquina

Esta versão da dissertação contém as correções e alterações sugeridas pela Comissão Julgadora durante a defesa da versão original do trabalho, realizada em 15/10/2021. Uma cópia da versão original está disponível no Instituto de Matemática e Estatística da Universidade de São Paulo.

Comissão Julgadora:

- Prof. Dr. Renato Vicente - IME-USP
- Prof. Dr. Fabio Gagliardi Cozman - EP-USP
- Prof. Dr. Marcos Oliveira Prates - UFMG

Acknowledgement

First of all, I would like to thank my entire family, especially my parents Telma and Eduardo and my brother Henrique, but also my grandparents, uncles and cousins. All of them were very important during my trajectory and always gave me a lot of strength. Also, I would like to thank my girlfriend Jaislan, who shares with me all the good and bad moments.

Secondly, I would like to thank all the mentors I gained during my master's degree. Some special thanks go to: (i) my advisor Renato Vicente, who believed in my potential and gave me great freedom to develop several projects; (ii) professors Luís Gustavo Esteves, Anatoly Yambartsev and Victor Fossaluza for their teachings; (iii) professors Fabio Cozman and Rafael Izbicki, who were always open to help me or talk about research.

Thirdly, I would like to thank the Advanced Institute for Artificial Intelligence (AI2) for the financial support at the end of the master's degree and Serasa Experian DataLab for partnering and providing datasets and computational power during the development of this work. In addition, I would like to thank the researchers and data scientists of AI2 and DataLab for the exchange of experiences we had over the last year.

Fourthly, I would like to thank my old friends and those I gained during my master's degree. Some special thanks go to: (i) the friends of "Bar do Zeca", for their great friendship and hangouts in São Paulo; (ii) the dear friends of IME/USP, who made the master's degree a lot lighter; (iii) the "Coopequianos" and friends for the eternal partnership; (iv) scientific initiation students, whom I co-advised, because of their trust and friendship.

Fifthly, I would like to thank Tikal Tech for partnering during the master's years and for the freedom I had to innovate there.

Lastly, I would like to thank Conselho Nacional de Desenvolvimento Científico e Tecnológico" (CNPq) for the financial support for two years of the master's degree.

Agradecimentos

Em primeiro lugar, gostaria de fazer um agradecimento à toda minha família, em especial meus pais Telma e Eduardo e meu irmão Henrique, mas sem esquecer avós, tios e primos. Todos foram muitos importantes durante a minha trajetória e sempre me deram muita força. Além disso, gostaria de agradecer à minha namorada Jaislan, que vibra comigo em todas as minhas conquistas.

Em segundo lugar, gostaria de agradecer a todos os mentores que ganhei durante o mestrado. Alguns agradecimentos especiais são para: (i) meu orientador Renato Vicente, que acreditou no meu potencial e me deu grande liberdade para desenvolvimento de diversos projetos; (ii) os professores Luís Gustavo Esteves, Anatoly Yambartsev e Victor Fossaluza por seus ensinamentos; (iii) os professores Fabio Cozman e Rafael Izbicki sempre estiveram abertos para me ajudar ou conversar sobre pesquisa.

Em terceiro lugar, gostaria de agradecer ao Advanced Institute for Artificial Intelligence (AI2) pelo auxílio financeiro ao fim do mestrado e ao Serasa Experian DataLab pela parceria e fornecimento de conjuntos de dados e poder computacional durante o desenvolvimento deste trabalho. Além disso, gostaria de agradecer aos pesquisadores e cientistas de dados do AI2 e DataLab pela troca de experiências que tivemos no último ano.

Em quarto lugar, gostaria de agradecer aos amigos de sempre e aqueles que ganhei durante o mestrado. Alguns agradecimentos especiais são para: (i) o pessoal do Bar do Zeca, pela grande amizade e rolês em São Paulo; (ii) os amigos queridos do IME/USP, que tornarem o mestrado mais leve; (iii) os amigos Coopequianos e agregados pela eterna parceria; (iv) os alunos de iniciação científica, que co-oriento, pela confiança.

Em quinto lugar, gostaria de agradecer à Tikal Tech pela parceira durante os anos de mestrado e pela liberdade que tive para inovar.

Em último lugar, gostaria de agradecer ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo auxílio financeiro durante dois anos do mestrado.

Abstract

POLO, F. M. **Covariate Shift Adaptation and Dataset Shift Decomposition in Machine Learning**. Master in Statistics - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2021.

In supervised learning, we often have access to a limited sample, in size or quality (e.g., lack of labels), of the population/distribution of interest, for which we want to create predictive models. However, it is possible that we have less limited access to data sampled from another population, more or less similar to the one of interest. Training models using only data from the population of interest may be impossible or result in sub-optimal models, so it would be interesting to use data from the other population in order to get better results or make training possible. In these situations, as the distributions of interest and the one that we can sample with few restrictions are different, we say that there is dataset shift. In dataset shift situations, employing domain adaptation techniques when training supervised models is essential for theoretical guarantees of good results in the population of interest. The two kinds of dataset shift we will discuss about in this work are covariate shift and concept drift/shift.

The main objectives of this work are: (i) to review the main concepts and methods related to covariate shift and covariate shift adaptation; (ii) propose contributions to the covariate shift adaptation literature, connecting concepts present in modern literature; (iii) propose the decomposition of the dataset shift into covariate shift and expected concept drift/shift as a new approach to better understand situations in which we deal with dataset shift.

Keywords: Dataset Shift, Covariate Shift, Domain Adaptation, Effective Sample Size, Dimensionality, Dataset Shift Decomposition, Concept Drift, Machine Learning, Statistics.

Resumo

POLO, F. M. **Adaptação para Covariate Shift e decomposição do Dataset Shift no Aprendizado de Máquina**. Programa de Mestrado em Estatística - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2021.

No aprendizado supervisionado, muitas vezes temos acesso a uma amostra limitada, em tamanho ou qualidade (e.g., falta de rótulos), de dados da população/distribuição de interesse, para a qual queremos criar modelos preditivos. No entanto, é possível que tenhamos acesso pouco limitado a dados amostrados de outra população, mais ou menos parecida com a de interesse. Treinar modelos utilizando somente dados da população de interesse pode ser impossível ou resultar em modelos sub-ótimos, então seria interessante utilizar os dados provenientes da outra população a fim de obter melhores resultados ou tornar o treinamento possível. Nessas situações, como as distribuições de interesse e aquela que podemos amostrar com poucas restrições são diferentes, dizemos que há dataset shift. Em situações de dataset shift, empregar técnicas de adaptação de domínio ao treinar modelos supervisionados é essencial para garantias teóricas de bons resultados na população de interesse. Os dois tipos de dataset shift que discutiremos neste trabalho são covariate shift e concept drift/shift.

Os objetivos principais deste trabalho são: (i) revisar principais conceitos e métodos relacionados ao covariate shift e covariate shift adaptation; (ii) propor contribuições para a literatura de covariate shift adaptation, conectando conceitos presentes em discussões atuais; (iii) propor a decomposição do dataset Shift em covariate shift e concept drift/shift esperado como uma nova abordagem para melhor entendimento de situações em que lidamos com dataset shift.

Palavras-chave: Dataset Shift, Covariate Shift, Adaptação de Domínio, Effective Sample Size, Dimensionalidade, Decomposição do Dataset Shift, Concept Drift, Machine Learning, Estatística.

Contents

List of Figures

List of Tables

Mathematical Notation

In this chapter, we present a significant part of the notation used during this work. Some specific notation are presented on the text itself.

The notation was inspired by the textbook *"Deep Learning"* (Goodfellow *et al.*, 2016).

Numbers and Arrays

$a,\ x,\ y$	Examples of scalars
a, x, y	Examples of random variables
$\boldsymbol{a},\ \boldsymbol{x},\ \boldsymbol{y}$	Examples of vectors
$\mathbf{a},\ \mathbf{x},\ \mathbf{y}$	Examples of random vectors
$\boldsymbol{A},\ \boldsymbol{X},\ \boldsymbol{Y}$	Examples of matrices
$\mathbf{A},\ \mathbf{X},\ \mathbf{Y}$	Examples of random matrices
$\boldsymbol{I}_n$	Identity matrix with n rows and n columns
$\boldsymbol{I}$	Identity matrix with dimensionality implied by context

Sets

$\mathcal{A}$	Example of an arbitrary set
$\mathcal{A} \backslash \mathcal{B}$	Set subtraction, i.e., the set containing the elements of $\mathcal{A}$ that are not in $\mathcal{B}$
$[n] = \{1, \ldots, n\}$	The set of all integers between 0 and n
$\mathbb{R}$	The set of real numbers
$\mathbb{N}$	The set of natural numbers, i.e., $\{n\}_{n=1}^{\infty}$
$\mathcal{X}$	Input/Features Space
$\mathcal{Y}$	Output Space
$\mathcal{H}$	Hypothesis Class
$\mathbb{H}$	A Reproducing Kernel Hilbert Space (RKHS) or simply a Hilbert Space

Functions

$f : \mathcal{A} \to \mathcal{B}$	The function f with domain $\mathcal{A}$ and codomain $\mathcal{B}$
$f \circ h$	Composition of the functions f and h
$h(\boldsymbol{x}; \boldsymbol{\theta})$	A function of $\boldsymbol{x}$ parametrized by $\boldsymbol{\theta}$ (Sometimes we write $h_\theta(\boldsymbol{x})$ or $h(\boldsymbol{x})$)
$\log(x)$	Natural logarithm of x
$\|\|\boldsymbol{x}\|\|_p$	L^p norm of $\boldsymbol{x}$
$\|\|\boldsymbol{x}\|\|$	A norm, implied by the context, of $\boldsymbol{x}$
$\mathbb{I}_{\text{condition}}$	Indicator function. It is 1 if the condition is true, 0 otherwise

Calculus

$\frac{\mathrm{d}f}{\mathrm{d}x}$	Derivative of a function f with respect to x
$\frac{\partial f}{\partial x}$	Partial derivative of a function f with respect to x
$\nabla_{\boldsymbol{x}} f$	Gradient vector of a function f with respect to $\boldsymbol{x}$
$\int f(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$	Indefinite integral or definite integral over the entire domain of $\boldsymbol{x}$
$\int_{\mathcal{X}} f(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$	Definite integral with respect to $\boldsymbol{x}$ over the set $\mathcal{X}$

Probability Theory

$\mathbb{P},\ P,\ Q$	Examples of probability measures/distributions
$p,\ q$	Examples of probability density functions
$\mathbf{x} \sim P$	Random vector $\mathbf{x}$ has distribution P
$\mathbb{E}_{\mathbf{x}\sim P}[f(\mathbf{x})]$	Expectation of $f(\mathbf{x})$ with respect to P
$\mathrm{Var}_{\mathbf{x}\sim P}[f(\mathbf{x})]$	Variance of $f(\mathbf{x})$ under P
$P_{\mathbf{x}},\ P_{\mathrm{y}},\ P_{\mathbf{x},\mathrm{y}}$	Target population probability distributions of features and labels
$p_{\mathbf{x}},\ p_{\mathrm{y}},\ p_{\mathbf{x},\mathrm{y}}$	Target population probability density/mass functions of features and labels
$Q_{\mathbf{x}},\ Q_{\mathrm{y}},\ Q_{\mathbf{x},\mathrm{y}}$	Source population probability distributions of features and labels
$q_{\mathbf{x}},\ q_{\mathrm{y}},\ q_{\mathbf{x},\mathrm{y}}$	Source population probability density/mass functions of features and labels

Chapter 1

Introduction

In supervised learning, we often have access to a limited sample, in size or quality (e.g., lack of labels), of the population/distribution of interest, for which we want to create predictive models. However, it is possible that we have less limited access to data sampled from another population, more or less similar to the one of interest. Training models using only data from the population of interest may be impossible or result in sub-optimal models, so it would be interesting to use data from the other population in order to get better results or make training possible. In these situations, as the distributions of interest and the one that we can sample with few restrictions are different, we say that there is dataset shift. In dataset shift situations, employing domain adaptation techniques when training supervised models is essential for theoretical guarantees of good results in the population of interest. There are several types of dataset shift already documented in the literature (Moreno-Torres *et al.*, 2012; Quionero-Candela *et al.*, 2009), however the focus of this work will be almost complete in a particular case called covariate shift (Sugiyama and Kawanabe, 2012), and in the last chapter we also speak a little about concept drift/shift (Moreno-Torres *et al.*, 2012).

The main contributions of this work are presented in the following. The first contribution is presenting an extensive literature review regarding covariate shift and covariate shift adaptation. We start the review presenting a general view of the covariate shift problem, visiting fundamental works on the topics. We end our review presenting an extensive set of methods for density ratio estimation, which offers a solution for the covariate shift adaptation problem. We briefly discuss classic strategies for density ratio estimation and also

present more modern methodologies. The second contribution of this work is proposing a new understanding relationship between three central concepts in the modern covariate shift literature, which are effective sample size (ESS), features' dimensionality and generalization. Despite the three concepts being present in the recent literature, their connections are still unclear. We show that: (i) bigger ESSs lead to sharper generalization bounds, (ii) data dimensionality is directly linked to the ESS, and (iii) dimensionality reduction can make the ESS bigger. The third contribution is proposing a new way to characterize dataset shift in supervised learning tasks, permiting the researcher to quantify and decompose the total dataset shift, into a part that represents the covariate shift and another that represents the concept shift. With that, one can quantify each term separately and better understand the nature shifting data. We close that chapter showing an application using real credit data from Brazilians in the transition to the COVID-19 pandemic period.

This work is organized as follows: in Chapter 2, we make a brief review of fundamental concepts of statistical learning theory, which enables the formalization of concepts presented later on; in Chapter 3, we explain the covariate shift problem, exploring theoretical and applied aspects, besides reviewing the main way to solve the problem, which is using the importance weighting method; in Chapter 4, we provide an extensive review of methods for estimating density ratios used during importance weighting; in Chapter 5, we propose a new unifying theory that connects effective sample size, dimensionality and generalization, which are concepts present in the modern literature of covariate shift adaptation; finally, in Chapter 6, we propose a new approach to better understand data under dataset shift - in this approach, we decompose the dataset shift, materialized by a divergence between two distributions, into covariate shift and expected concept drift/shift.

Have a good reading!

Chapter 2

Elements of Statistical Learning Theory

This chapter aims at introducing the reader to fundamental elements of statistical learning theory. We present key concepts of the field that will be the basis of discussions in the rest of this dissertation such as models, loss functions, risk/errors, learning algorithms, generalization, and Bayes risk. For this chapter, we used as reference two textbooks used in more theoretical courses on statistical learning, which are Shalev-Shwartz and Ben-David (2014) and Mohri *et al.* (2012).

In the supervised learning framework, we have an independent and identically distributed data set $\{(\mathbf{x}_i, \mathrm{y}_i)\}_{i=1}^{n} \overset{iid}{\sim} F_{\mathbf{x},\mathrm{y}}$ and we want to learn a function h that helps us to predict the value of y_{n+1} if we know $\mathbf{x}_{n+1}$. The random vector $(\mathbf{x}_{n+1}, \mathrm{y}_{n+1})$ is an out-of-sample data point sampled from the same distribution as the original data. Assuming the probability distribution $F_{\mathbf{x},\mathrm{y}}$ measures events from a sigma-algebra of $\mathcal{X} \times \mathcal{Y}$, we call the set $\mathcal{X}$ the *input space*, features' space or covariates space and $\mathcal{Y}$ the *output space*, labels' space or targets' space.

Given that our goal is to learn a function h that helps us predict y when we know $\mathbf{x}$, it is important to give more details on the set in which h may be defined. We assume $\mathbf{x}$ to be a random vector of features and y to be a random variable.

Definition 2.1. (Hypothesis Set or model): A hypothesis set $\mathcal{H}$ is a subset of measurable functions that have $\mathcal{X}$ as domain and $\mathcal{Y}$ as cododomain.

In practice, we set $\mathcal{H}$ beforehand and try to find $h \in \mathcal{H}$ that best suits our context. For example, $\mathcal{H}$ could be the set of all neural networks with a certain architecture or the set of

all regression hyperplanes. To choose $h \in \mathcal{H}$ that best satisfies us, we first need to formaly understand what "satisfy" means. We first define the concept of loss function.

Definition 2.2. (Loss Function): A loss function L is a function that compares, two by two, elements of two sets $\mathcal{Y}'$ and $\mathcal{Y}$ and returns us a non-negative real scalar. That is, $L : \mathcal{Y}' \times \mathcal{Y} \to \mathbb{R}_+$.

In many cases, $\mathcal{Y}' = \mathcal{Y}$, but this is not strictly necessary.

Suppose we have a loss function L, two hypotheses $h_1, h_2 \in \mathcal{H}$ and a data point $(\boldsymbol{x}, y)$ sampled according to the distribution $F_{\mathbf{x},\mathrm{y}}$. If $L[h_1(\boldsymbol{x}), y] \leq L[h_2(\boldsymbol{x}), y]$, we can think that the hypothesis h_1 explains y given $\boldsymbol{x}$ at least as well as h_2 does. Despite giving us some intuition, the last example has nothing to do with learning, as h_1 and h_2 are given in advance and the $(\boldsymbol{x}, y)$ point is deterministic. To move forward, we have to understand the concept of statistical risk. From now on, in order to facilitate the notation, we will assume that $F_{\mathbf{x}}$ is the marginal distribution of $\mathbf{x}$ and F_{y} is the marginal distribution of y.

Definition 2.3. (Statistical Risk): The statistical risk associated with a loss function L is a function of the hypothesis $h \in \mathcal{H}$ and it is defined as the expected value of $L[h(\mathbf{x}), \mathrm{y}]$. That is, the statistical risk associated with L is $R_L : \mathcal{H} \to \mathbb{R}_+$, given that:

$$R_L(h) = \mathop{\mathbb{E}}_{(\mathbf{x},\mathrm{y})\sim F_{\mathbf{x},\mathrm{y}}} L[h(\mathbf{x}), \mathrm{y}] \tag{2.1}$$

The statistical risk is also known in the literature as the generalization error because it gives us the expected out-of-sample error/loss. The ultimate goal of supervised statistical learning is to find a $h^* \in \mathcal{H}$ hypothesis that minimizes the statistical risk, that is, to find a $h^* \in \operatorname{argmin}_{h\in\mathcal{H}} R_L(h)$. The practical problem with this objective is that we, almost surely, do not know the statistical risk; that is, we cannot assess it directly. What happens in the real world is that we estimate the risk or some key features related to it and, in this way, we try to minimize it. One way of estimating statistical risk is using the empirical risk.

Definition 2.4. (Empirical Risk): The empirical risk associated with a loss function L is a function of the hypotheses $h \in \mathcal{H}$ and is defined as the arithmetic mean of $L[h(.), .]$ assessed on the data $\mathcal{S} = \{(\mathbf{x}_i, \mathrm{y}_i)\}_{i=1}^n \in (\mathcal{X} \times \mathcal{Y})^n$ sampled independently and identically

from $F_{\mathbf{x},\mathrm{y}}$. That is, the empirical risk associated with L is $\hat{R}_L : \mathcal{H} \times (\mathcal{X} \times \mathcal{Y})^n \to \mathbb{R}_+$, where:

$$\hat{R}_L(h, \mathcal{S}) = \frac{1}{n} \sum_{i=1}^{n} L[h(\mathbf{x}_i), \mathrm{y}_i] \tag{2.2}$$

The empirical risk is also known as empirical error. The empirical error can be seen as an estimator for the generalization error. We can still notice that if h is fixed, this estimator is an unbiased and consistent estimator for the generalization error. In the following example, we can also infer about its convergence rate:

Example 2.5. (*Generalization Bound*): We address the classification case in this example. In an ordinary classification problem we have $\mathcal{X} = \mathbb{R}^d$ and $\mathcal{Y} = \{0, 1, ..., K\}$, where $\mathcal{Y}$ is a set of possible labels. We have a sample $\mathcal{S} = \{(\mathbf{x}_i, \mathrm{y}_i)\}_{i=1}^n$ with components sampled independently from $F_{\mathbf{x},\mathrm{y}}$. We also denote $\mathcal{S} \sim F_{\mathcal{S}}$. Fixing $\mathcal{H}$ and $\epsilon > 0$, we choose any $h \in \mathcal{H}$. It is natural that our loss function is defined as follows $L(y', y) = \mathbb{I}[y' \neq y]$. Then the statistical risk is given by:

$$R_L(h) = \mathop{\mathbb{E}}_{(\mathbf{x},\mathrm{y}) \sim F_{\mathbf{x},\mathrm{y}}} \mathbb{I}[h(\mathbf{x}) \neq \mathrm{y}] \tag{2.3}$$

$$= F_{\mathbf{x},\mathrm{y}}\left(\{h(\mathbf{x}) \neq \mathrm{y}\}\right) \tag{2.4}$$

And the empirical risk is given by:

$$\hat{R}_L(h, \mathcal{S}) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}[h(\mathbf{x}_i) \neq \mathrm{y}_i] \tag{2.5}$$

Note that L is a bounded function (between 0 and 1) and, because of that, we can apply Hoeffding's inequality for bounded random variables (Vershynin, 2019) to get the following result:

$$F_{\mathcal{S}}\left[\left|\hat{R}_L(h, \mathcal{S}) - R_L(h)\right| \geq \epsilon\right] \leq 2\exp(-2n\epsilon^2) \tag{2.6}$$

This result assures us that, for a fixed hypothesis $h \in \mathcal{H}$, if our sample size is large enough, the empirical error will be close to the generalization error with high probability.

So far, when working with a hypothesis in a class $\mathcal{H}$, we assumed a fixed h. However, this does not make much sense within our learning context because, given a sample, we want to learn/infer the best possible hypothesis and not adopt one in advance. To introduce ourselves to learning, we must first understand the concept of learning algorithm.

Definition 2.6. (Learning Algorithm): Given a class of hypothesis $\mathcal{H}$ and given the set of the samples of size n, a learning algorithm $\mathcal{A} : (\mathcal{X} \times \mathcal{Y})^n \to \mathcal{H}$ is a map with input $\mathcal{S} \in (\mathcal{X} \times \mathcal{Y})^n$ and output $h_{\mathcal{S}} = \mathcal{A}(\mathcal{S}) \in \mathcal{H}$ respecting some optimality criteria.

The optimality criteria could be, for example, minimizing the empirical error. Under this criterion, we are working with the learning paradigm of *empirical risk minimization - ERM*. More formally, under the empirical risk minimization paradigm, given the class of hypotheses $\mathcal{H}$, a loss function L and a sample $\mathcal{S}$, the algorithm $\mathcal{A}$ returns a hypothesis $h_{\mathcal{S}}^{\text{ERM}} \in \text{argmin}_{h \in \mathcal{H}} \hat{R}_L(h, \mathcal{S}) \subseteq \mathcal{H}$.

Next, we present the concept of Agnostic Probably Almost Correct (PAC) Learnability (Shalev-Shwartz and Ben-David, 2014):

Definition 2.7. (Agnostic PAC Learnability): A hypothesis class $\mathcal{H}$ is Agnostic Probably Almost Correct Learnable if there is a function $n_{\mathcal{H}} : (0,1)^2 \to \mathbb{N}$, often called by sample complexity, with the following property: for every $\epsilon, \delta \in (0,1)$ and distribution $F_{\mathbf{x},\mathrm{y}}$ over $\mathcal{X} \times \mathcal{Y}$, when we feed a learning algorithm $\mathcal{A}$ with a sample $\mathcal{S} = \{(\mathbf{x}_i, \mathrm{y}_i)\}_{i=1}^n$ sampled independently from $F_{\mathbf{x},\mathrm{y}}$, $n \geq n_{\mathcal{H}}(\epsilon, \delta)$, it returns a hypothesis $h_{\mathcal{S}}$ such that:

$$F_{\mathcal{S}}\left[R_L(h_{\mathcal{S}}) - \min_{h \in \mathcal{H}} R_L(h) \geq \epsilon\right] \leq \delta \tag{2.7}$$

In other words, we say a class of hypothesis is Agnostic PAC Learnable if, given a large enough sample, we find a solution close to the best solution in the hypothesis class with high probability. Agnostic PAC Learnability is a strong assumption given it is a "distribution-free" concept, although it is still useful.

We now present and prove an important result that states the every finite hypothesis class is Agnostic PAC Learnable when we work with bounded loss functions. Although we usually do not work with finite classes in practice, that result is still appealing since infinite classes

reduce to finite classes when we work with a computer due to models' parameterization. More details in that remark can be found in Chapter 4 of Shalev-Shwartz and Ben-David (2014).

Theorem 2.8. ***(Finite hypothesis classes are PAC learnable):*** *If $\mathcal{H}$ is a finite class, the loss function*[1] *$L \in [0,1]$, and we adopt the empirical risk minimization paradigm, we are also able to minimize, with high probability, the generalization error when n is big. That is, for every $\epsilon, \delta \in (0,1)$ and for every distribution $F_{\mathbf{x},\mathrm{y}}$ over $\mathcal{X} \times \mathcal{Y}$, when we feed a learning algorithm $\mathcal{A}$ with a sample $\mathcal{S} = \{(\mathbf{x}_i, \mathrm{y}_i)\}_{i=1}^{n}$ sampled independently from $F_{\mathbf{x},\mathrm{y}}$, $n \geq n_{\mathcal{H}}(\epsilon, \delta)$, it returns a hypothesis $h_{\mathcal{S}}^{\mathrm{ERM}}$ such that:*

$$F_{\mathcal{S}}\left[R_L(h_{\mathcal{S}}^{\mathrm{ERM}}) - \min_{h \in \mathcal{H}} R_L(h) \geq \epsilon\right] \leq \delta \tag{2.8}$$

Proof. The proof is similar to that presented in Shalev-Shwartz and Ben-David (2014) and can be found in Appendix A. □

In the paragraphs above and examples we only addressed the empirical risk minimization (ERM) paradigm for learning, however there are other approaches such as the regularized risk minimization (REG). In this approach, fixing a hypothesis class $\mathcal{H}$ and having a sample $\mathcal{S}$, the learning algorithm returns the hypothesis $h_{\mathcal{S}}^{\mathrm{REG}}$ that minimizes the sum of the empirical error and a simple term that quantify the complexity of the hypothesis in $\mathcal{H}$. Formally we have $h_{\mathcal{S}}^{\mathrm{REG}} \in \mathrm{argmin}_{h \in \mathcal{H}}\ \hat{R}_L(h, \mathcal{S}) + \lambda||h|| \subseteq \mathcal{H}$, where $||.||$ is a norm and $\lambda \geq 0$ is a hyperparameter that controls the trade-off between the empirical risk minimization and the complexity of $h \in \mathcal{H}$ (Mohri *et al.*, 2012). The parameter λ is usually chosen in a cross-validation procedure, and the regularized solution usually works better as it allows the control for *overfitting* (Hastie *et al.*, 2009).

So far, we have assumed that the hypothesis class $\mathcal{H}$ is taken a priori, which happens in practice, but we have not commented on the consequences of that choice. By choosing a class of hypotheses (linear regressions, neural networks, decision trees, etc.), we are embedding in our analysis what we call by *inductive bias* (Shalev-Shwartz and Ben-David, 2014). The

[1] Actually, L only need to be bound in $[a, b]$, for a, b reals. We assume $a = 0$ and $b = 1$, for a simplification purpose and because this is the case of the popular loss function $L(y', y) = \mathbb{I}[y' \neq y]$ used to assess models in classification tasks.

inductive bias is how we see the world through the hypothesis contained in the chosen class, that is, from the model adopted. The inductive bias is inevitable in practice because we do not know the process that generates data, and we work with approximations of reality. It is important to reaffirm that we are exposed to this type of bias to understand how we can have better models. An interesting way to better understand the dynamics of statistical learning is to decompose the generalization error into the *estimation*, *approximation*, and *Bayes* errors (Mohri *et al.*, 2012). In order to understand this decomposition, we must first understand the definition of Bayes Error/Risk (Mohri *et al.*, 2012):

Definition 2.9. (Bayes Risk): Given a joint probability distribution $F_{\mathbf{x},\mathrm{y}}$ over the set $\mathcal{X} \times \mathcal{Y}$ and a loss function L, we define $\mathcal{F}$ as the set of all measurable functions with domain $\mathcal{X}$ and codomain $\mathcal{Y}$. The Bayes Risk is defined as:

$$R_L^{\text{BAYES}} := \inf_{f \in \mathcal{F}} R_L(f) \tag{2.9}$$

The Bayes Risk is a lower bound for the generalization error. An important observation is that we never work with the function class/models $\mathcal{F}$, but we work with a subset $\mathcal{H}$ of it since the first is very broad. That is, almost inevitably $\inf_{h \in \mathcal{H}} R_L(h) > R_L^{\text{BAYES}}$, which can be understood as a result of the inductive bias. Now that we can better understand what the Bayes Risk is and its message, we are ready to understand how we can decompose the generalization error. Fixing a hypothesis class $\mathcal{H}$ and a loss function L, we define $h^* \in \operatorname{argmin}_{h \in \mathcal{H}} R_L(h)$ and we write the following decomposition for the generalization error of a particular h in $\mathcal{H}$:

$$R_L(h) = [R_L(h) - R_L(h^*)] + [R_L(h^*) - R_L^{\text{BAYES}}] + R_L^{\text{BAYES}} \tag{2.10}$$

It is possible to see that we have three terms on the RHS of the equation, the first is the estimation error, the second is the approximation error and the third is the Bayes risk. This equation tells us that h makes errors primarily because we were unable to reach the best hypothesis of the class $\mathcal{H}$ defined a priori; second, h makes errors because the class $\mathcal{H}$ is not as broad enough and does not include the best possible hypothesis; lastly, h makes errors

because there is a irreducible risk - due to noise - which is impossible to settle. There is usually a trade-off between the first two terms on the RHS: on the one hand we can choose a richer hypothesis class $\mathcal{H}$ that makes the approximation error decrease but on the other hand it is harder to approach the best solution within the class $\mathcal{H}$ increasing the estimation error.

Chapter 3

Introducing the Problem of Covariate Shift

The most fundamental assumption in statistical machine learning is that the training data are sampled from the same probability distribution which we have interest in. Restricting ourselves to the supervised learning scenario, we often assume that we have a random sample/dataset $\mathcal{S} = \{(\mathbf{x}_i, \mathrm{y}_i)\}_{i=1}^{n}$ sampled independently from a distribution $P_{\mathbf{x},\mathrm{y}}$ and that we would like to infer about a out-of-sample quantity y_{m+1} given the features $\mathbf{x}_{m+1}$, both of which were also sampled from $P_{\mathbf{x},\mathrm{y}}$. Unfortunately, this is not the case in many practical situations. When the distribution from which our data were sampled is not the same as the distribution of interest, we can say that there was some kind of *dataset shift* or that we are working with statistical learning in non-stationary environments. There are several types of *dataset shift* problems (Quionero-Candela *et al.*, 2009), that is, the distributions that we can sample from and the one of our interest can be different for many reasons and this also change the way we face each situation. In this work we are especially interested in one type of *dataset shift* which is known as *covariate shift* - in this type of non-stationarity, we assume that the joint distribution of features and labels shifts only by the marginal distribution of the features, while the conditional distribution of the labels given the features remains static. In Section 3.1 we focus our efforts on detailing the *covariate shift* problem.

We refer to the distribution we can completely sample from as the source/training distribution ($Q_{\mathbf{x},\mathrm{y}}$) and the distribution of interest as the target/test distribution ($P_{\mathbf{x},\mathrm{y}}$). It is

assumed we sample unlabeled samples from $P_{\mathbf{x}}$, in contrast to $Q_{\mathbf{x},\mathrm{y}}$, in which we can sample labeled data points. Also, for the sake of simplicity, we assume both marginal distributions of features $Q_{\mathbf{x}}$ and $P_{\mathbf{x}}$ are absolutely continuous w.r.t. the Lebesgue measure with probability density functions $q_{\mathbf{x}}$ and $p_{\mathbf{x}}$, such that $\text{support}(p_{\mathbf{x}}) \subseteq \text{support}(q_{\mathbf{x}})$.

3.1 Understanding *Covariate Shift*

Covariate shift is understood as a scenario in which we have a training/source joint distribution $Q_{\mathbf{x},\mathrm{y}}$ which differs from the test/target distribution $P_{\mathbf{x},\mathrm{y}}$. Features and labels are sampled according to the same conditional distribution $Q_{\mathrm{y}|\mathbf{x}} = P_{\mathrm{y}|\mathbf{x}}$ but different marginals $Q_{\mathbf{x}} \neq P_{\mathbf{x}}$. We thus suppose that labeled pairs $\{(\mathbf{x}_i, \mathrm{y}_i)\}_{i=1}^{n}$ are sampled independently from $Q_{\mathbf{x},\mathrm{y}}$, while unlabeled vectors $\{\mathbf{x}'_i\}_{i=1}^{n'}$ are independently sampled from $P_{\mathbf{x}}$. Our objective is training supervised models with data sampled from $Q_{\mathbf{x},\mathrm{y}}$ but with good performance to predict labels from unlabeled samples from $P_{\mathbf{x},\mathrm{y}}$.

It remains to be understood why this is a problem within the statistical supervised learning framework. The answer is not obvious, given that covariate shift is a problem linked only to the marginal distribution of features, and we are generally concerned with estimating quantities linked to the conditional distribution of labels given the features, especially when using discriminative (and non-generative) models/algorithms such as linear regression, logistic regression, MLP neural networks, random forest, XGBoost, etc.

By fixing a hypothesis class $\mathcal{H}$, a loss function L and remembering the fundamental concepts of the statistical learning theory presented in Chapter 2, we know that our main objective here is to minimize, with respect to the hypotheses in $\mathcal{H}$, the statistical risk assessed according to the distribution of interest (target/test). Our goal then is to find $h^* \in \mathcal{H}$, such that:

$$h^* \in \arg\min_{h \in \mathcal{H}} R_L(h) = \tag{3.1}$$

$$= \arg\min_{h \in \mathcal{H}} \mathop{\mathbb{E}}_{(\mathbf{x},\mathrm{y}) \sim P_{\mathbf{x},\mathrm{y}}} [L(h(\mathbf{x}), \mathrm{y})] \tag{3.2}$$

$$= \arg\min_{h \in \mathcal{H}} \mathbb{E}_{\mathbf{x} \sim P_{\mathbf{x}}} \mathbb{E}_{\mathrm{y}|\mathbf{x}} [L(h(\mathbf{x}), \mathrm{y})] \tag{3.3}$$

Looking at the equations above, it is possible to check that the distribution $P_{\mathbf{x}}$ can play a central role giving more or less importance to different regions of $\mathcal{X}$ when evaluating the hypothesis in $\mathcal{H}$.

If we could find $h \in \mathcal{H}$ that minimizes the second expected value in 3.3 for all values $\boldsymbol{x} \in \mathcal{X}$, then covariate shift would not cause big problems, since the minimization of $R_L(h)$ would not depend on the measure associated with $\mathbf{x}$. Moreover, this is the case explored by Shimodaira (2000), in which the authors claim that correctly specified models would be immune to covariate shift if we have a big enough sample. That is because the risk would be minimized asymptotically. By a correctly specified model, Shimodaira (2000) means a scenario in which we choose a model class $\mathcal{H}$ that contains the actual label generating function we are trying to estimate.

However, in practical situations, $\mathcal{H}$ is usually a class of models that we use to approximate reality and therefore we will not find $h \in \mathcal{H}$ that minimizes the second expected in 3.3 for all values $\boldsymbol{x} \in \mathcal{X}$. Furthermore, we always work with limited sample size. Therefore the marginal distribution of the features becomes an important factor in the computation of risk since it weighs regions of $\mathcal{X}$ according to their importance. In Example 3.1, we present a simple situation in which covariate shift is a relevant factor and should be considered in the modeling process.

Example 3.1. (*Covariate Shift*): In this example, extracted from Sugiyama and Kawanabe (2012), we show a situation in which covariate shift makes empirical error minimization, with or without regularization, a learning strategy that does not work well. We assume that the features are sampled from the following source distribution $\mathrm{x}_i \sim \mathcal{N}\big(1, \frac{1}{4}\big)$ and target distribution $\mathrm{x}'_i \sim \mathcal{N}\big(2, \frac{1}{16}\big)$, $i = 1, ..., 200$. Since we are dealing with covariate shift, the conditional distribution of labels is identical in both source and target domains. In this example, the conditional distribution of labels is given as follows:

$$\mathrm{y}_i|\mathrm{x}_i = x_i \sim \mathcal{N}\left(\frac{\mathrm{sen}(\pi x_i)}{\pi x_i},\ \frac{1}{16}\right) \tag{3.4}$$

In the Figure 3.1 it is possible to see the theoretical densities of $\mathbf{x}_i$ and $\mathbf{x}'_i$.

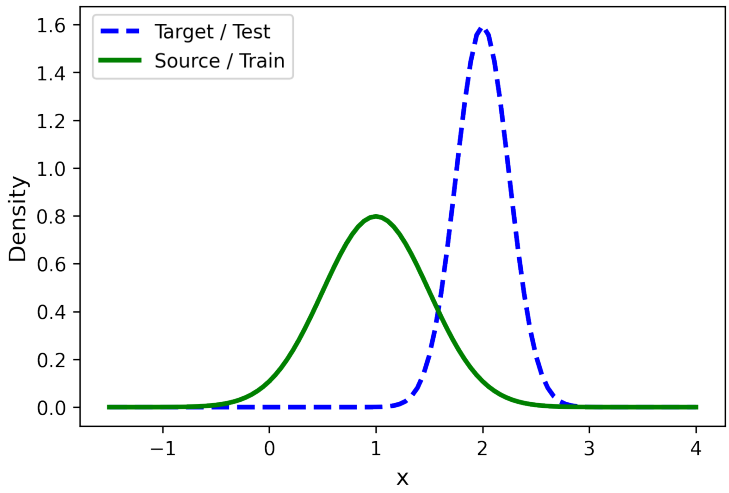

Figure 3.1: *Theoretical densities of* $\mathbf{x}'_i$ *sampled from target distribution (blue) e* $\mathbf{x}_i$ *sampled from source distribution (green). This plot shows how divergent feature distributions are in different populations.*

In Figure 3.2, one can check the joint distribution of the sampled points of the feature and label distributions of the two populations.

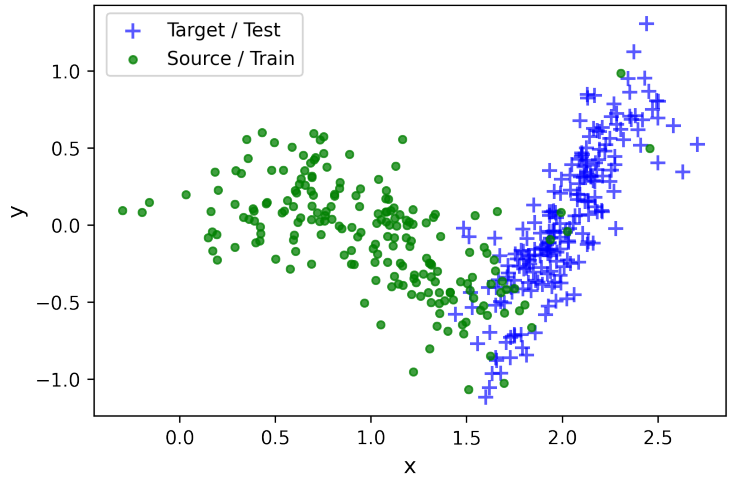

Figure 3.2: *Scatter plot of samples* $(\boldsymbol{x}, y)$ *from both source and target populations. Even though the conditional distribution of labels is the same in both populations, a supervised model that performs well source (green) domain does not necessarily perform well on target (blue) domain.*

It is important to remember that, in real situations, we do not have access to labels of samples from the target distribution, but only to their features. In this example we deal with the problem in a naive way: we consider the source/train and target/test population distributions are the same - not performing any correction, just to see how our approach fails. As with any other supervised task, we first define a hypothesis class and a learning algorithm. For the sake of example, we use the hypothesis class $\mathcal{H}$ of linear regressions, so

we model our response variable as follows:

$$\mathrm{y} = h(\mathrm{x}) + \varepsilon,\ \varepsilon \sim \mathcal{N}(0, \sigma^2) \tag{3.5}$$

$$= \beta_0 + \beta_1 \mathrm{x} + \varepsilon,\ \varepsilon \sim \mathcal{N}(0, \sigma^2) \tag{3.6}$$

In this example, our learning algorithm minimizes the empirical mean squared error given by:

$$\widehat{MSE}(\boldsymbol{\beta}) = \frac{1}{200}||\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}||^2 \tag{3.7}$$

With

$$\boldsymbol{X} = \begin{pmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_{200} \end{pmatrix}, \qquad \boldsymbol{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_{200} \end{pmatrix}, \qquad \boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix} \tag{3.8}$$

The solution for $\boldsymbol{\beta}$ is given by

$$\begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \end{bmatrix} = \left(\boldsymbol{X}^\top \boldsymbol{X}\right)^{-1} \boldsymbol{X}^\top \boldsymbol{y} \tag{3.9}$$

After obtaining the line that minimizes the empirical error, we can plot it with the dispersion of points already sampled.

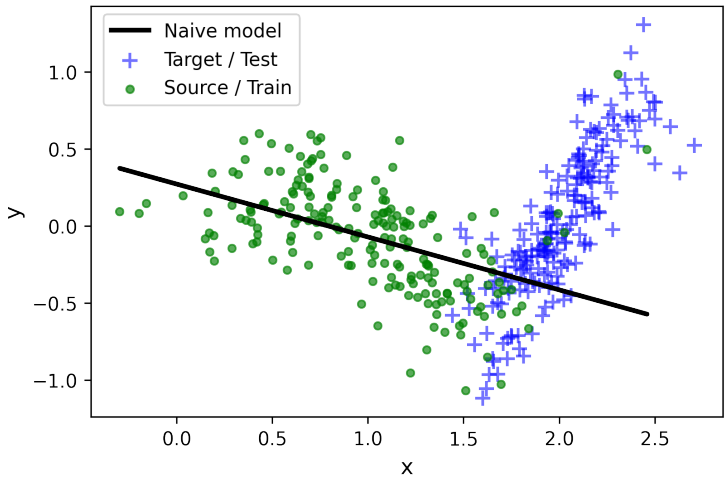

Figure 3.3: *The solution for a linear regression model that minimizes empirical mean squared error evaluated on source/training data. This model performs reasonably well on the source population but poorly on the target population.*

It is possible to see that the result worsens if we want to infer the labels for target/test samples. The naive approach, that is, training our model only considering data from the source distribution, works poorly due to covariate shift.

3.2 Covariate Shift Adaptation

A way to get around the covariate shift problem, obtaining results that generalize better with respect to the target population, is to weight the data points in our sample in order to mimic the distribution of interest (Huang *et al.*, 2007; Kanamori *et al.*, 2009a; Shimodaira, 2000; Sugiyama *et al.*, 2008). In this approach, each data point from source distribution $(\mathbf{x}_i, \mathrm{y}_i) \sim Q_{\mathbf{x},\mathrm{y}}$ in our training set receives a weight w_i. These weights are used to adapt the calculation of the empirical error, with or without a regularizer. A first step in understanding the rationale behind this solution is to realize that the statistical risk assessed in the target distribution $P_{\mathbf{x},\mathrm{y}}$ can be rewritten in terms of the source distribution $Q_{\mathbf{x},\mathrm{y}}$. For now on we assume: (i) $Q_{\mathrm{y}|\mathbf{x}} = P_{\mathrm{y}|\mathbf{x}}$ and $Q_{\mathbf{x}} \neq P_{\mathbf{x}}$; (ii) distributions $P_{\mathbf{x}}$ and $Q_{\mathbf{x}}$ have probability density

functions $p_{\mathbf{x}}$ and $q_{\mathbf{x}}$ such that $\text{support}(p_{\mathbf{x}}) \subseteq \text{support}(q_{\mathbf{x}})$. Then:

$$R_L(h) = \mathbb{E}_{(\mathbf{x},\mathrm{y})\sim P_{\mathbf{x},\mathrm{y}}}\left[L(h(\mathbf{x}),\mathrm{y})\right] \tag{3.10}$$

$$= \mathbb{E}_{\mathbf{x}\sim P_{\mathbf{x}}}\mathbb{E}_{\mathrm{y}|\mathbf{x}}\left[L(h(\mathbf{x}),\mathrm{y})\right] \tag{3.11}$$

$$= \int p_{\mathbf{x}}(\boldsymbol{x})\mathbb{E}_{\mathrm{y}|\mathbf{x}}\left[L(h(\boldsymbol{x}),\mathrm{y})\right]\mathrm{d}\boldsymbol{x} \tag{3.12}$$

$$= \int \frac{p_{\mathbf{x}}(\boldsymbol{x})}{q_{\mathbf{x}}(\boldsymbol{x})} q_{\mathbf{x}}(\boldsymbol{x})\mathbb{E}_{\mathrm{y}|\mathbf{x}}\left[L(h(\boldsymbol{x}),\mathrm{y})\right]\mathrm{d}\boldsymbol{x} \tag{3.13}$$

$$= \mathbb{E}_{\mathbf{x}\sim Q_{\mathbf{x}}}\mathbb{E}_{\mathrm{y}|\mathbf{x}}\left[w(\mathbf{x})L(h(\mathbf{x}),\mathrm{y})\right] \tag{3.14}$$

$$= \mathbb{E}_{(\mathbf{x},\mathrm{y})\sim Q_{\mathbf{x},\mathrm{y}}}\left[w(\mathbf{x})L(h(\mathbf{x}),\mathrm{y})\right] \tag{3.15}$$

The density ratio function $w = p_{\mathbf{x}}/q_{\mathbf{x}}$ is often called by "importance function" or "weighting function". This result leads to the following definition:

Definition 3.2. (Weighted Statistical Risk): The weighted statistical risk associated with a loss function L and a measurable non-negative weighting function w is a function of the hypothesis $h \in \mathcal{H}$. Assuming $(\mathbf{x},\mathrm{y}) \sim F_{\mathbf{x},\mathrm{y}}$, it is defined as the expected value of $w(\mathbf{x})L[h(\mathbf{x}),\mathrm{y}]$. That is, the statistical risk associated with L is $R_L : \mathcal{H} \to \mathbb{R}_+$, where:

$$R_{L,w}(h) = \mathop{\mathbb{E}}_{(\mathbf{x},\mathrm{y})\sim F_{\mathbf{x},\mathrm{y}}}\left[w\left(\mathbf{x}\right)L\left(h\left(\mathbf{x}\right),\mathrm{y}\right)\right] \tag{3.16}$$

Now that we have expressed the statistical risk of our interest in terms of the source/training distribution, it is easier to understand how we could minimize it. The two most direct alternatives would be to minimize the weighted empirical error or the regularized weighted empirical error weighted by the importance function w.

Definition 3.3. (Weighted Empirical Risk): The weighted empirical risk associated with a loss function L and a measurable weighting function w is a function of the hypothesis $h \in \mathcal{H}$ and is defined as the weighted average of $L(h(.),.)$ evaluated in the sample $\mathcal{S} = \{(\mathbf{x}_i,\mathrm{y}_i)\}_{i=1}^n \in (\mathcal{X}\times\mathcal{Y})^n$ independently sampled from $F_{\mathbf{x},\mathrm{y}}$, where the weighting is done by the measurable function w. Mathematically, the weighted empirical risk associated with L

and the weighting function w is $\hat{R}_{L,w} : \mathcal{H} \times (\mathcal{X} \times \mathcal{Y})^n \to \mathbb{R}_+$:

$$\hat{R}_{L,w}(h, \mathcal{S}) = \frac{1}{n} \sum_{i=1}^{n} w(\mathbf{x}_i) L(h(\mathbf{x}_i), \mathrm{y}_i) \tag{3.17}$$

Definition 3.4. (Weighted Regularized Empirical Risk): The weighted regularized empirical risk associated with a loss function L and a measurable weighting function w is a function of the hypothesis $h \in \mathcal{H}$ and is defined as the weighted average of $L(h(.), .)$ evaluated in the sample $\mathcal{S} = \{(\mathbf{x}_i, \mathrm{y}_i)\}_{i=1}^{n} \in (\mathcal{X} \times \mathcal{Y})^n$ independently sampled from $F_{\mathbf{x},\mathrm{y}}$ plus a regularizer term Ω that controls the complexity of hypothesis. Mathematically, the weighted regularized empirical risk associated with L and the weighting function w is $\hat{R}_{L,w,\Omega} : \mathcal{H} \times (\mathcal{X} \times \mathcal{Y})^n \to \mathbb{R}_+$:

$$\hat{R}_{L,w,\Omega}(h, \mathcal{S}) = \frac{1}{n} \sum_{i=1}^{n} w(\mathbf{x}_i) L(h(\mathbf{x}_i), \mathrm{y}_i) + \Omega(h) \tag{3.18}$$

When the regularizer function Ω appears explicitly in the objective function above, it is generally given by a norm. It can be implicitly constraining $\mathcal{H}$, as it can happen when we control tree depth in decision trees. To understand how minimizing the weighted empirical risk would work in practice if we fix w equals the density ratio, we return to the Example 3.1.

Example 3.5. (*Covariate Shift* Adaptation): We follow everything as defined in the Example 3.1, then we have that $\mathbf{x}_i \sim \mathcal{N}\left(1, \frac{1}{4}\right)$ and $\mathbf{x}'_i \sim \mathcal{N}\left(2, \frac{1}{16}\right)$, therefore the weighting function in this case will be given by the function w as defined below:

$$w(x) = \frac{p_{\mathrm{x}}(x)}{q_{\mathrm{x}}(x)} \tag{3.19}$$

$$= \frac{\frac{1}{\frac{1}{4}\sqrt{2\pi}} \exp\left[-\frac{(x-2)^2}{2\frac{1}{16}}\right]}{\frac{1}{\frac{1}{2}\sqrt{2\pi}} \exp\left[-\frac{(x-1)^2}{2\frac{1}{4}}\right]} \tag{3.20}$$

$$= 2 \exp\left[2(x-1)^2 - 8(x-2)^2\right] \tag{3.21}$$

Our learning algorithm that now minimizes the weighted mean squared empirical error

returns the following solution:

$$\begin{bmatrix} \hat{\beta}_0 \\ \\ \hat{\beta}_1 \end{bmatrix} = \left(\boldsymbol{X}^\top \boldsymbol{W} \boldsymbol{X}\right)^{-1} \boldsymbol{X}^\top \boldsymbol{W} \boldsymbol{y} \tag{3.22}$$

Given that $\boldsymbol{W}$ is a diagonal matrix with entries $W_{i,i} = w(x_i)$. An interesting thing to look at is how w takes values according to possible values of x. In Figure 3.4 it is possible to see the function w and the marginal distributions from which the features were sampled.

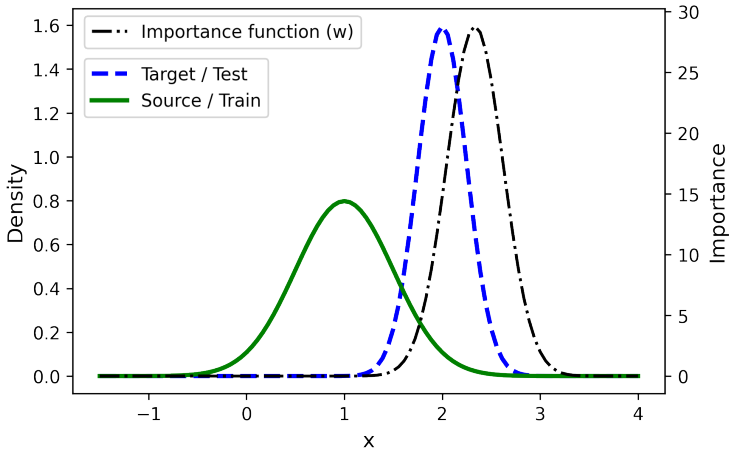

Figure 3.4: *Plotting the weighting function $w(x)$ and theoretical densities of $\mathbf{x}'_i$ sampled from target distribution (blue) e $\mathbf{x}_i$ sampled from source distribution (green). This plot shows how $w(x)$ is given by the density ratio.*

In the previous example, we can see from Figure 3.3 that the solution was not reasonable when we wanted to generalize results to the population of the non-selected. Now we see from Figure 3.5 that we achieved a much better result for our purpose.

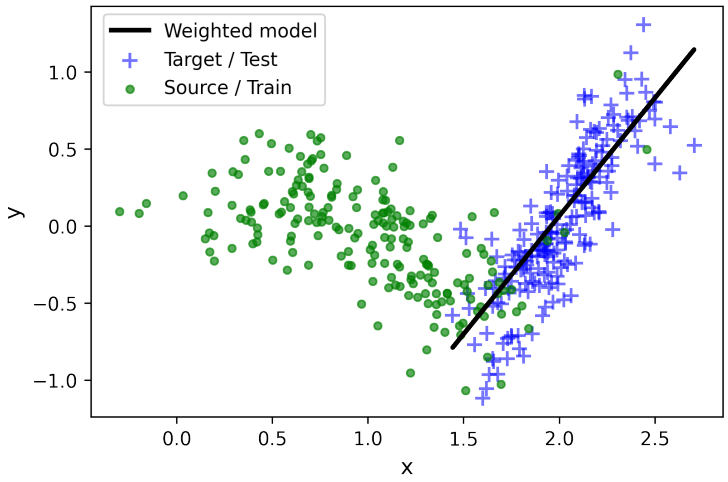

Figure 3.5: *The solution for a linear regression model that minimizes the weighted empirical mean squared error evaluated on source/training data. This model performs reasonably well on the target population.*

In real situations, we often do not know an analytical form for w, and we have to estimate it. In the next chapter, we present methods to estimate w.

3.3 Importance Weighted Cross Validation (IWCV)

One of the most popular methods to evaluate machine learning models is cross-validation, given its generality and easiness of understanding. It has been shown in the literature that cross-validation offers an almost unbiased procedure to estimate the statistical risk of a supervised model (Schölkopf *et al.*, 2002). By "almost unbiased," we mean it is unbiased considering the total of $m < n$ data points used to train the model. Indeed, the classical properties from cross-validation are not valid when we have covariate shift, but Sugiyama *et al.* (2007) offers an adaption, and then it is possible to recover them.

Consider the training dataset $\mathcal{T} = \{(y_i, \boldsymbol{x}_i)\}_{i=1}^n$, a partition of $\mathcal{T}$ with randomly chosen elements of roughly the same size $\{\mathcal{T}_j\}_{j=1}^k$, a loss function L and the importance function $w(\boldsymbol{x}) = p(\boldsymbol{x})/q(\boldsymbol{x})$. Also, let $\hat{h}_{j'} \in \mathcal{H}$ be the hypothesis learned using as the training set the data points $\cup_{j \neq j'} \mathcal{T}_j$. Then, the $k-$fold importance weighting cross-validation (kIWCV) estimate of the generalization error is given by

$$\widehat{R}_{L,kIWCV} = \frac{1}{k} \sum_{j=1}^{k} \frac{1}{|\mathcal{T}_j|} \sum_{(y,\boldsymbol{x}) \in \mathcal{T}_j} w(\boldsymbol{x}) L(\hat{h}_j(\boldsymbol{x}), y) \tag{3.23}$$

A special case is when $k = n$. In this case, we have the importance weighting leave-one-out cross-validation (IWLOOCV) method

$$\widehat{R}_{L,IWLOOCV} = \frac{1}{n}\sum_{i=1}^{n} w(\boldsymbol{x}_i)L(\hat{h}_i(\boldsymbol{x}_i), y_i) \tag{3.24}$$

Where $\hat{h}_i$ is learned using as training set $\mathcal{T} \setminus \{(\boldsymbol{x}_i, y_i)\}$.

3.4 Conclusion

In this chapter, we went through the fundamentals of covariate shift and covariate shift adaptation. Firstly, we defined covariate shift and showed why it could a problem in real applications of machine learning. Secondly, we discussed about covariate shift adaptation using importance weighting, which is the most popular way of solving that problem. Finally, we showed how to adapt cross validation when using importance weighting. In the next chapter, we introduce several ways of estimating the importance function, also known as density ratio.

Chapter 4

Importance Estimation Methods for Covariate Shift Adaptation

Although we presented a solution to the covariate shift problem, we still have to estimate w in order to perform covariate shift adaptation. There are some popular ways of making this estimate in the literature, and we present a literature review of importance/density ratio estimation in the next sections. Other importance estimation methods are also presented in Appendix B so this chapter is not too long.

We refer to the source/training distribution of features as $Q_{\mathbf{x}}$ and the target/test distribution of features as $P_{\mathbf{x}}$. Also, for the sake of simplicity, we assume both distributions are absolutely continuous with probability density functions $q_{\mathbf{x}}$ and $p_{\mathbf{x}}$, such that $\text{support}(p_{\mathbf{x}}) \subseteq \text{support}(q_{\mathbf{x}})$. Furthermore, we always consider $\boldsymbol{x}$ and $\{\boldsymbol{x}_i\}_{i=1}^{n}$ to be data points independently sampled from $Q_{\mathbf{x}}$, and $\boldsymbol{x}'$ and $\{\boldsymbol{x}'_i\}_{i=1}^{n'}$ to be data points independently sampled from $P_{\mathbf{x}}$, i.e., realizations of $\mathbf{x} \sim Q_{\mathbf{x}}$ and $\mathbf{x}' \sim P_{\mathbf{x}}$, respectively.

4.1 Kernel Density Estimation

The most direct solution for the weight function w estimation problem is to do nonparametric estimation of the densities $p_{\mathbf{x}}(\boldsymbol{x})$ and $q_{\mathbf{x}}(\boldsymbol{x})$ using kernels and then take the ratio between the functions (Shimodaira, 2000). When $\mathbf{x}'$ and $\mathbf{x}$ are absolutely continuous random variables/vectors, a popular choice is to use the Gaussian kernel.

Example 4.1. (*Gaussian kernel*): The Gaussian kernel is given by the map $(\boldsymbol{x}, \boldsymbol{c}, \sigma) \mapsto K_\sigma(\boldsymbol{x}, \boldsymbol{c})$, where $\boldsymbol{x}, \boldsymbol{c} \in \mathbb{R}^d$ and $\sigma > 0$:

$$K_\sigma(\boldsymbol{x}, \boldsymbol{c}) = \exp\left(-\frac{||\boldsymbol{x} - \boldsymbol{c}||^2}{2\sigma^2}\right) \tag{4.1}$$

This kernel gets the special name of Gaussian kernel because its functional form is the core of the density associated with a random variable with the normal distribution. In this case, it is common to put the subscript σ to denote the kernel bandwidth and can be chosen by cross validation (Wasserman, 2006). The Gaussian kernel gives us a measure of similarity between $\boldsymbol{x}$ and $\boldsymbol{c}$. We could also say that the kernel is *centered* at point $\boldsymbol{c}$.

If $\mathbf{x}$ is an absolutely continuous random vector with density f and we have a sequence of data points $\{\boldsymbol{x}_i\}_{i=1}^n$ which are independent realizations of that vector, then an estimate for the density f using the Gaussian kernel K_σ would be given by:

$$\hat{f}(\boldsymbol{x}) = \frac{1}{n(2\pi\sigma^2)^{d/2}} \sum_{i=1}^{n} K_\sigma(\boldsymbol{x}, \boldsymbol{x}_i) \tag{4.2}$$

Where d is the length of the vectors $\boldsymbol{x}_i$. In our case, in which we want to approximate a density ratio, it would be enough to estimate the two densities separately and then take the ratio of the estimated functions. Explicitly, having two sequences of data points, $\{\boldsymbol{x}_i\}_{i=1}^n$ and $\{\boldsymbol{x}'_i\}_{i=1}^{n'}$, sampled i.i.d. from densities $q_{\mathbf{x}}$ (source) and $p_{\mathbf{x}}$ (target), a training data point is weighted by

$$\hat{w}(\boldsymbol{x}) = \frac{\hat{p}_{\mathbf{x}}(\boldsymbol{x})}{\hat{q}_{\mathbf{x}}(\boldsymbol{x})} \tag{4.3}$$

Although this solution potentially solves the problem of estimating w, when d is large, this approach suffer from the curse of dimensionality - the points become widely spaced in high dimensions and the number of them in our dataset would have to grow exponentially as a function of d for a good estimate (Wasserman, 2006). An alternative is estimating w directly, which can be an easier task.

4.2 Probabilistic Classification Method

This is a relatively simple and effective way to estimate the weighting function. First, consider a binary random variable t that indicates whether a random data point $\mathbf{x}$ is a sample of the target distribution $P_{\mathbf{x}}$ and not of the source distribution $Q_{\mathbf{x}}$. Thus, we can write $p_{\mathbf{x}}(\boldsymbol{x}) = f_{\mathbf{x}|\mathrm{t}}(\boldsymbol{x}|\mathrm{t}=1)$, $q_{\mathbf{x}}(\boldsymbol{x}) = f_{\mathbf{x}|\mathrm{t}}(\boldsymbol{x}|\mathrm{t}=0)$, and $f_{\mathbf{x}}(\boldsymbol{x}) = \mathbb{P}(\mathrm{t}=0)q_{\mathbf{x}}(\boldsymbol{x}) + \mathbb{P}(\mathrm{t}=1)p_{\mathbf{x}}(\boldsymbol{x})$. Using Bayes' Theorem, we can write w as follows (Sugiyama *et al.*, 2012b):

$$w(\boldsymbol{x}) = \frac{p_{\mathbf{x}}(\boldsymbol{x})}{q_{\mathbf{x}}(\boldsymbol{x})} \tag{4.4}$$

$$= \frac{f_{\mathbf{x}|\mathrm{t}}(\boldsymbol{x}|\mathrm{t}=1)}{f_{\mathbf{x}|\mathrm{t}}(\boldsymbol{x}|\mathrm{t}=0)} \tag{4.5}$$

$$= \frac{\mathbb{P}(\mathrm{t}=0)}{\mathbb{P}(\mathrm{t}=1)} \frac{\mathbb{P}(\mathrm{t}=1|\mathbf{x}=\boldsymbol{x})}{\mathbb{P}(\mathrm{t}=0|\mathbf{x}=\boldsymbol{x})} \tag{4.6}$$

$$\propto \frac{\mathbb{P}(\mathrm{t}=1|\mathbf{x}=\boldsymbol{x})}{\mathbb{P}(\mathrm{t}=0|\mathbf{x}=\boldsymbol{x})} \tag{4.7}$$

Suppose we have data points $\{\boldsymbol{x}'_i\}_{i=1}^{n'}$ independently sampled from $P_{\mathbf{x}}$ and $\{\boldsymbol{x}_i\}_{i=1}^{n}$ independently sampled from $Q_{\mathbf{x}}$. Appending $\{(\boldsymbol{x}'_i, 1)\}_{i=1}^{n'}$ and $\{(\boldsymbol{x}_i, 0)\}_{i=1}^{n}$, and then training a probabilistic classifier, e.g., MLP neural networks (Hastie *et al.*, 2009) or XGBoost (Chen and Guestrin, 2016), to discriminate samples according to labels 1 and 0, it is possible to approximate $\mathbb{P}(\mathrm{t}=1|\mathbf{x}=\boldsymbol{x})$ and $\mathbb{P}(\mathrm{t}=0|\mathbf{x}=\boldsymbol{x})$. On the other hand, the quantities $\mathbb{P}(\mathrm{t}=0)$ and $\mathbb{P}(\mathrm{t}=1)$ can be estimated by $n/(n'+n)$ and $n'/(n'+n)$, respectively.

One particular situation where we can effectively use the probabilistic classifier method is when we face a problem of sample selection bias due to "missing at random" (MAR) data points (Moreno-Torres *et al.*, 2012). That is a (covariate shift) problem of missing data, where labels are missing for a dataset subset. Our interest is in the distribution that generated the entire dataset, diverging from the most common way to understand covariate

shift. In this specific case, the weighting function could be written as follows:

$$w(\boldsymbol{x}) = \frac{f_{\mathbf{x}}(\boldsymbol{x})}{q_{\mathbf{x}}(\boldsymbol{x})} \tag{4.8}$$

$$= \frac{f_{\mathbf{x}}(\boldsymbol{x})}{f_{\mathbf{x}|\mathrm{t}}(\boldsymbol{x}|\mathrm{t}=0)} \tag{4.9}$$

$$= \frac{\mathbb{P}(\mathrm{t}=0)}{\mathbb{P}(\mathrm{t}=0|\mathbf{x}=\boldsymbol{x})} \tag{4.10}$$

$$\propto \frac{1}{\mathbb{P}(\mathrm{t}=0|\mathbf{x}=\boldsymbol{x})} \tag{4.11}$$

The optimization of hyperparameters will depend on the model chosen to estimate the conditional probability distribution of t. It can be done as in any other supervised task using cross validation.

4.3 Spectral Series Estimator

The idea of the method introduced by Izbicki *et al.* (2014) is to decompose w into a series of orthonormal eigenfunctions of a kernel-based operator. Let $L^2(\mathcal{X}, Q_{\mathbf{x}})$ be a vector space of functions $h : \mathcal{X} \to \mathbb{R}$ such that $||h||^2_{Q_{\mathbf{x}}} = \langle h, h \rangle_{Q_{\mathbf{x}}} < \infty$, where the inner product is given by:

$$\langle h, g \rangle_{Q_{\mathbf{x}}} = \int_{\mathcal{X}} h(\boldsymbol{x}) g(\boldsymbol{x}) q_{\mathbf{x}}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \tag{4.12}$$

Let K_σ be a Gaussian kernel[1], and let $\{\psi_j\}_{j=1}^{\infty}$ be eigenfunctions of the integral operator (Rosasco *et al.*, 2010; Wainwright, 2019) $\kappa : L^2(\mathcal{X}, Q_{\mathbf{x}}) \to L^2(\mathcal{X}, Q_{\mathbf{x}})$ defined as

$$\kappa(h)(\boldsymbol{x}) = \langle K_\sigma(\boldsymbol{x}, .), h(.) \rangle_{Q_{\mathbf{x}}} \tag{4.13}$$

The, the set of functions $\{\psi_j\}_{j=1}^{\infty}$ forms a orthonormal basis of the space $L^2(\mathcal{X}, Q_{\mathbf{x}})$

[1] Other positive semidefinite, symmetric and bounded kernels are also possible.

(Izbicki *et al.*, 2014; Minh, 2010). If $w \in L^2(\mathcal{X}, Q_{\mathbf{x}})$, then

$$w = \sum_{j=1}^{\infty} \beta_j \psi_j \tag{4.14}$$

For a set of scalars $\{\beta_j\}_{j=1}^{\infty}$. Given that $\{\psi_j\}_{j=1}^{\infty}$ is an orthonormal basis, we have that

$$\langle \psi_i, \psi_j \rangle_{Q_{\mathbf{x}}} = \mathbb{I}(i = j) \tag{4.15}$$

And,

$$\beta_j = \langle w, \psi_j \rangle_{Q_{\mathbf{x}}} \tag{4.16}$$

$$= \int_{\mathcal{X}} w(\boldsymbol{x}) \psi_j(\boldsymbol{x}) q_{\mathbf{x}}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \tag{4.17}$$

$$= \mathop{\mathbb{E}}_{\mathbf{x}' \sim P_{\mathbf{x}}} \Big[\psi_j(\mathbf{x}') \Big] \tag{4.18}$$

That is, $\beta_j \psi_j$ is the orthogonal projection of w onto the subspace generated by ψ_j. Consider we have the samples $\{\boldsymbol{x}_i\}_{i=1}^{n}$ and $\{\boldsymbol{x}'_i\}_{i=1}^{n'}$ sampled from $Q_{\mathbf{x}}$ (source) and $P_{\mathbf{x}}$ (target). To estimate each element of the basis $\{\psi_j\}_{j=1}^{\infty}$, we first calculate Gram's matrix $[K_\sigma(\boldsymbol{x}_i, \boldsymbol{x}_j)]_{i,j=1}^{n}$ and then diagonalize it in order to obtain the first J eigenvalues $\lambda_1 \geq ... \geq \lambda_J \geq 0$ and their respective eigenvectors $\{\widetilde{\psi}_j\}_{j=1}^{J}$. The eigenvectors of Gram's matrix are given by:

$$\widetilde{\psi}_j := \Big(\widetilde{\psi}_j(\boldsymbol{x}_1), ..., \widetilde{\psi}_j(\boldsymbol{x}_n) \Big), \quad j \in [J] \tag{4.19}$$

To this point, we estimated the basis functions applied to the training sample from source population. However, we would like to extend them to any other point in order to estimate the scalars $\{\beta_j\}_{j=1}^{J}$ using samples from $P_{\mathbf{x}}$. For that, it is possible to use the Nystrom Extension (Drineas and Mahoney, 2005; Izbicki *et al.*, 2014) to obtain the desired function

for all $\boldsymbol{x} \in \mathcal{X}$:

$$\widehat{\psi}_j(\boldsymbol{x}) = \frac{\sqrt{n}}{\lambda_j} \sum_{i=1}^{n} \widetilde{\psi}_j(\boldsymbol{x}_i) K_\sigma(\boldsymbol{x}, \boldsymbol{x}_i) \tag{4.20}$$

Then, each β_j is estimated using samples from $P_{\mathbf{x}}$ as:

$$\widehat{\beta}_j = \frac{1}{n'} \sum_{i=1}^{n'} \widehat{\psi}_j(\boldsymbol{x}'_i) \tag{4.21}$$

Finally, the Spectral Series Estimator is given by

$$\widehat{w} = \max\left[0, \sum_{j=1}^{J} \widehat{\beta}_j \widehat{\psi}_j\right] \tag{4.22}$$

The model selection step is straightforward using the empirical mean squared error and cross validation to choose the best values for hyperparameters σ and J. In this case, J controls the bias-variance trade-off (Izbicki *et al.*, 2014).

4.4 Kullback-Leibler Importance Estimation Procedure (KLIEP)

The Kullback-Leibler Importance Estimation Procedure (KLIEP) was introduced by Sugiyama *et al.* (2008), and consists in modelling the importance function w as a linear combination of basis functions:

$$w_{\boldsymbol{\beta}}(\boldsymbol{x}) = \sum_{t=1}^{T} \beta_t \varphi_t(\boldsymbol{x}) \tag{4.23}$$

$$= \boldsymbol{\beta}^\top \boldsymbol{\varphi}(\boldsymbol{x}) \tag{4.24}$$

The basis functions $\boldsymbol{\varphi} = (\varphi_1, ..., \varphi_T)$ are hyperparameters and the vector $\boldsymbol{\beta} = (\beta_1, ..., \beta_T)$ is learned from data. Due to the fact we can write a model for $p_{\mathbf{x}}$ as $p_{\mathbf{x}}(\boldsymbol{x}; \boldsymbol{\beta}) = w_{\boldsymbol{\beta}}(\boldsymbol{x}) q_{\mathbf{x}}(\boldsymbol{x})$, we admit $\beta_t,\ \varphi_t \geq 0$ for $t = 1, ..., T$. The functions $\{\varphi_t\}_{t=1}^{T}$ can be defined in many ways; however, it is common to use Gaussian kernels centered at random points from the set of

target/test data points, namely $\varphi_t(\boldsymbol{x}) = K_\sigma\big(\boldsymbol{x}, \boldsymbol{x}'_i\big)$. For each t we randomly select a sample from the test set. Writing the Kullback-Leibler Divergence (Kullback, 1997) between our model $p_{\mathbf{x}}(\boldsymbol{x};\boldsymbol{\beta})$ and the data density $p_{\mathbf{x}}(\boldsymbol{x})$

$$J(\boldsymbol{\beta}) = D_{\mathrm{KL}}\left[p_{\mathbf{x}}||p_{\mathbf{x}}(.;\boldsymbol{\beta})\right] \tag{4.25}$$

$$= \mathop{\mathbb{E}}_{\mathbf{x}'\sim P_{\mathbf{x}}}\left\{\log\left[\frac{p_{\mathbf{x}}\left(\mathbf{x}'\right)}{p_{\mathbf{x}}(\mathbf{x}';\boldsymbol{\beta})}\right]\right\} \tag{4.26}$$

$$= \mathop{\mathbb{E}}_{\mathbf{x}'\sim P_{\mathbf{x}}}\left\{\log\left[\frac{p_{\mathbf{x}}\left(\mathbf{x}'\right)}{w_{\boldsymbol{\beta}}\left(\mathbf{x}'\right)q_{\mathbf{x}}\left(\mathbf{x}'\right)}\right]\right\} \tag{4.27}$$

$$= \mathop{\mathbb{E}}_{\mathbf{x}'\sim P_{\mathbf{x}}}\left\{\log\left[\frac{p_{\mathbf{x}}\left(\mathbf{x}'\right)}{q_{\mathbf{x}}\left(\mathbf{x}'\right)}\right]\right\} - \mathop{\mathbb{E}}_{\mathbf{x}'\sim P_{\mathbf{x}}}\left\{\log\left[w_{\boldsymbol{\beta}}\left(\mathbf{x}'\right)\right]\right\} \tag{4.28}$$

$$= C - J'(\boldsymbol{\beta}) \tag{4.29}$$

Where C is a constant term that does not depend on $w_{\boldsymbol{\beta}}$ and can be ignored. Minimizing $J(\boldsymbol{\beta})$ w.r.t. $\boldsymbol{\beta}$ is equivalent to maximizing $J'(\boldsymbol{\beta})$ w.r.t. $\boldsymbol{\beta}$. Given the samples $\{\boldsymbol{x}_i\}_{i=1}^{n}$ and $\{\boldsymbol{x}'_i\}_{i=1}^{n'}$ independently sampled from $Q_{\mathbf{x}}$ (source) and $P_{\mathbf{x}}$ (target), we write the empirical objective function as

$$\hat{J}'(\boldsymbol{\beta}) = \frac{1}{n'}\sum_{i=1}^{n'}\log\left[w_{\boldsymbol{\beta}}\left(\boldsymbol{x}'_i\right)\right] \tag{4.30}$$

$$= \frac{1}{n'}\sum_{i=1}^{n'}\log\left[\sum_{t=1}^{T}\beta_t\varphi_t(\boldsymbol{x}'_i)\right] \tag{4.31}$$

$$= \frac{1}{n'}\sum_{i=1}^{n'}\log\left[\boldsymbol{\beta}^\top\boldsymbol{\varphi}(\boldsymbol{x}'_i)\right] \tag{4.32}$$

Interestingly, we could write $\hat{J}'(\boldsymbol{\beta}) = \frac{1}{n'}\sum_{i=1}^{n'}\log\left[p_{\mathbf{x}}(\boldsymbol{x}'_i;\boldsymbol{\beta})\right] + C'$. Then maximizing $\hat{J}'(\boldsymbol{\beta})$ is equivalent to maximize the likelihood.

Note that it is not possible to identify $\boldsymbol{\beta}$ solely by maximizing the function above. Given that $p_{\mathbf{x}}(.;\boldsymbol{\beta})$ must approximate a density function, it makes sense if we consider the following

constraint:

$$1 = \int p_{\mathbf{x}}(\boldsymbol{x}; \boldsymbol{\beta}) \mathrm{d}\boldsymbol{x} = \int w_{\boldsymbol{\beta}}(\boldsymbol{x})\, q_{\mathbf{x}}(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x} \tag{4.33}$$

$$\approx \frac{1}{n} \sum_{i=1}^{n} w_{\boldsymbol{\beta}}(\boldsymbol{x}_i) = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{\beta}^{\top} \boldsymbol{\varphi}(\boldsymbol{x}_i) \tag{4.34}$$

Then, the empirical version of our optimization problem is:

$$\max_{\boldsymbol{\beta} \in \mathbb{R}^T} \sum_{i=1}^{n'} \log \left[\boldsymbol{\beta}^{\top} \boldsymbol{\varphi}(\boldsymbol{x}'_i) \right] \tag{4.35}$$

$$\text{s.t.} \sum_{i=1}^{n} \boldsymbol{\beta}^{\top} \boldsymbol{\varphi}(\boldsymbol{x}_i) = n \text{ and } \boldsymbol{\beta}, \boldsymbol{\varphi} \geq 0 \tag{4.36}$$

The optimization problem is then solved by a variation of the gradient ascent algorithm, ensuring feasibility conditions in an iterative fashion (Sugiyama *et al.*, 2008). According to the authors, we may also include a regularization term, imposing some constraints on $\boldsymbol{\beta}$. It is also possible to lose the condition 4.34 to allow some variation (Sugiyama *et al.*, 2008).

Gaussian kernels are used as basis functions. Sugiyama *et al.* (2008) recommends the kernels to be centered at a subset of the target/test data points. Two points require clarification: (i) why we choose points from the test set and (ii) why we work with a subset of the data and not with all available data points. First, we chose to center the kernels on the test set data points as they are located in regions of space where w assumes large values. Second, we only work with a subset of the data because, according to Sugiyama *et al.* (2008), the algorithm's performance tends not to change much from a reasonable number of selected points, e.g., $T \approx 100$. On the other hand, the optimization algorithm can be very costly if T is too large. Sampling without replacement T test data points from $\{\boldsymbol{x}'_j\}_{j=1}^{n'}$, we get the set of points $\{\boldsymbol{c}_t\}_{=1}^{T}$ and we can represent our model for w as:

$$w_{\boldsymbol{\beta}}(\boldsymbol{x}) = \sum_{t=1}^{T} \beta_t K_{\sigma}(\boldsymbol{x}, \boldsymbol{c}_t) \tag{4.37}$$

$$= \sum_{t=1}^{T} \beta_t \exp\left(-\frac{||\boldsymbol{x} - \boldsymbol{c}_t||^2}{2\sigma^2} \right) \tag{4.38}$$

The model selection is straightforward since this is a hyperparameter tuning phase. As our objective is to maximize $\hat{J}'$, Sugiyama *et al.* (2008) suggests researchers use an ordinary K-fold cross validation procedure to tune regularization parameters or σ.

4.5 Least-Squares Importance Fitting (LSIF)

The Least-Squares Importance Fitting (LSIF) approach was introduced by Kanamori *et al.* (2009b). Just like KLIEP, we model w as a linear combination of basis functions:

$$w_{\boldsymbol{\beta}}(\boldsymbol{x}) = \sum_{t=1}^{T} \beta_t \varphi_t(\boldsymbol{x}) \tag{4.39}$$

$$= \boldsymbol{\beta}^\top \boldsymbol{\varphi}(\boldsymbol{x}) \tag{4.40}$$

The basis functions $\boldsymbol{\varphi}$ and the vector $\boldsymbol{\beta}$ follow the same specifications given in Section 4.4. Writing and manipulating the objective function $J(\boldsymbol{\beta})$, which we want to minimize w.r.t. $\boldsymbol{\beta}$:

$$J(\boldsymbol{\beta}) = \frac{1}{2} \mathop{\mathbb{E}}_{\mathbf{x} \sim Q_{\mathbf{x}}} \left\{ \left[w_{\boldsymbol{\beta}}(\mathbf{x}) - w(\mathbf{x}) \right]^2 \right\} \tag{4.41}$$

$$= \frac{1}{2} \mathop{\mathbb{E}}_{\mathbf{x} \sim Q_{\mathbf{x}}} \left[w_{\boldsymbol{\beta}}^2(\mathbf{x}) \right] - \mathop{\mathbb{E}}_{\mathbf{x} \sim Q_{\mathbf{x}}} \left[w_{\boldsymbol{\beta}}(\mathbf{x}) \, w(\mathbf{x}) \right] + \frac{1}{2} \mathop{\mathbb{E}}_{\mathbf{x} \sim Q_{\mathbf{x}}} \left[w^2(\mathbf{x}) \right] \tag{4.42}$$

$$= \frac{1}{2} \mathop{\mathbb{E}}_{\mathbf{x} \sim Q_{\mathbf{x}}} \left[w_{\boldsymbol{\beta}}^2(\mathbf{x}) \right] - \mathop{\mathbb{E}}_{\mathbf{x}' \sim P_{\mathbf{x}}} \left[w_{\boldsymbol{\beta}}(\mathbf{x}') \right] + C \tag{4.43}$$

$$= \frac{1}{2} \mathop{\mathbb{E}}_{\mathbf{x} \sim Q_{\mathbf{x}}} \left\{ \left[\boldsymbol{\beta}^\top \boldsymbol{\varphi}(\mathbf{x}) \right]^2 \right\} - \mathop{\mathbb{E}}_{\mathbf{x}' \sim P_{\mathbf{x}}} \left[\boldsymbol{\beta}^\top \boldsymbol{\varphi}(\mathbf{x}') \right] + C \tag{4.44}$$

$$= J'(\boldsymbol{\beta}) + C \tag{4.45}$$

Where C is a constant term that does not depend on $w_{\boldsymbol{\beta}}$ and we can simply ignore it.

Our main optimization problem is then given by:

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^T} \quad \frac{1}{2}\mathop{\mathbb{E}}_{\mathbf{x}\sim Q_{\mathbf{x}}}\left\{\left[\boldsymbol{\beta}^\top\boldsymbol{\varphi}\left(\mathbf{x}\right)\right]^2\right\} - \mathop{\mathbb{E}}_{\mathbf{x}'\sim P_{\mathbf{x}}}\left[\boldsymbol{\beta}^\top\boldsymbol{\varphi}\left(\mathbf{x}'\right)\right] \tag{4.46}$$

$$\text{s.t. } \boldsymbol{\beta}, \boldsymbol{\varphi} \geq 0 \tag{4.47}$$

Having the instances $\{\boldsymbol{x}_i\}_{i=1}^n$ and $\{\boldsymbol{x}'_i\}_{i=1}^{n'}$, we write the empirical version of the objective function as follows:

$$\hat{J}'(\boldsymbol{\beta}) = \frac{1}{2n}\sum_{i=1}^{n}\left[\boldsymbol{\beta}^\top\boldsymbol{\varphi}\left(\boldsymbol{x}_i\right)\right]^2 - \frac{1}{n'}\sum_{i=1}^{n'}\boldsymbol{\beta}^\top\boldsymbol{\varphi}\left(\boldsymbol{x}'_j\right) \tag{4.48}$$

$$= \frac{1}{2n}\sum_{i=1}^{n}\left[\sum_{t=1}^{T}\beta_t\varphi_t\left(\boldsymbol{x}_i\right)\right]^2 - \frac{1}{n'}\sum_{i=1}^{n'}\sum_{t=1}^{T}\beta_t\varphi_t\left(\boldsymbol{x}'_j\right) \tag{4.49}$$

$$= \frac{1}{2n}\sum_{i=1}^{n}\sum_{t,t'=1}^{T}\beta_t\beta_{t'}\varphi_t\left(\boldsymbol{x}_i\right)\varphi_{t'}\left(\boldsymbol{x}_i\right) - \frac{1}{n'}\sum_{i=1}^{n'}\sum_{t=1}^{T}\beta_t\varphi_t\left(\boldsymbol{x}'_j\right) \tag{4.50}$$

$$= \frac{1}{2}\sum_{t,t'=1}^{T}\beta_t\beta_{t'}\left[\frac{1}{n}\sum_{i=1}^{n}\varphi_t\left(\boldsymbol{x}_i\right)\varphi_{t'}\left(\boldsymbol{x}_i\right)\right] - \sum_{t=1}^{T}\beta_t\left[\frac{1}{n'}\sum_{i=1}^{n'}\varphi_t\left(\boldsymbol{x}'_j\right)\right] \tag{4.51}$$

$$= \frac{1}{2}\sum_{t,t'=1}^{T}\beta_t\beta_{t'}\widehat{H}_{t,t'} - \sum_{t=1}^{T}\beta_t\hat{h}_t \tag{4.52}$$

$$= \frac{1}{2}\boldsymbol{\beta}^\top\widehat{\boldsymbol{H}}\boldsymbol{\beta} - \boldsymbol{\beta}^\top\hat{\boldsymbol{h}} \tag{4.53}$$

Where $\widehat{\boldsymbol{H}}$ is a matrix of dimensions $T \times T$ and $\hat{\boldsymbol{h}}$ is a vector of size T. The entry (t, t') of $\widehat{\boldsymbol{H}}$ and the entry t of $\hat{\boldsymbol{h}}$ are given by

$$\widehat{H}_{t,t'} = \frac{1}{n}\sum_{i=1}^{n}\varphi_t\left(\boldsymbol{x}_i\right)\varphi_{t'}\left(\boldsymbol{x}_i\right) \qquad \hat{h}_t = \frac{1}{n'}\sum_{i=1}^{n'}\varphi_t\left(\boldsymbol{x}'_j\right) \tag{4.54}$$

Reformulating the optimization problem to its empirical version:

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^T} \quad \left[\frac{1}{2}\boldsymbol{\beta}^\top \widehat{\boldsymbol{H}}\boldsymbol{\beta} - \boldsymbol{\beta}^\top \hat{\boldsymbol{h}} + \lambda||\boldsymbol{\beta}||_1\right] \tag{4.55}$$

$$\text{s.t. } \boldsymbol{\beta}, \boldsymbol{\varphi} \geq 0 \tag{4.56}$$

Where $||\beta||_1$ is a regularization term of the type l_1, which is used to induce sparsity in the solution, and $\lambda \geq 0$ is a hyperparameter used to control this penalty, consequently the bias-variance trade-off. In practice, we could also use other regularization functions such as $||\beta||_2^2$, which would give us a regularization of the type l_2. The basis functions $\{\varphi_t\}_{t=0}^T$ are usually given by Gaussian kernels, and the non-negativity condition for these functions would already be satisfied - the reasoning behind the basis functions is the same as presented in Section 4.4. The above problem is a convex quadratic program, so it has a unique solution, and its solution can be calculated efficiently by ordinary nonlinear programming techniques. Kanamori *et al.* (2009b) shows that if we work with regularization of the type l_1, it is possible to efficiently use an algorithm to obtain the regularization path solution.

Kanamori *et al.* (2009b) states the model selection procedure can be implemented using Information Criteria or standard cross validation methods. Also, using regularization of type l_1 could make the choice of λ more efficient, as discussed above.

4.6 Conclusion

In this chapter, we introduced several ways of estimating the importance function, also known as density ratio. Our presentation did not intend to cover all possible methods, but indeed we tried to cover a good portion of what we considered to be the most important ones, ensuring diversity in nature and motivations of the methods.

Chapter 5

Effective Sample Size, Dimensionality, and Generalization in Covariate Shift Adaptation

In this chapter, we will briefly review what has been presented so far and then introduce the reader to this dissertation's major contributions. The ideas contained in this chapter were reproduced from Polo and Vicente (2021).

5.1 Introduction

A fundamental assumption in supervised statistical learning is that the data used to train our models and the data we want to make predictions for are sampled from the same distribution. Usually, real-world machine leaning applications, explicitly or implicitly, rely on this assumption. However, that assumption is violated when there is covariate shift (Shimodaira, 2000; Sugiyama and Kawanabe, 2012). In this scenario, we have a training/source joint distribution $Q_{\mathbf{x},\mathrm{y}}$ which differs from the test/target distribution $P_{\mathbf{x},\mathrm{y}}$. Features are sampled from different marginals $Q_{\mathbf{x}} \neq P_{\mathbf{x}}$ while labels are sampled according to the same conditional distribution $Q_{\mathrm{y}|\mathbf{x}} = P_{\mathrm{y}|\mathbf{x}}$. In the training phase, labeled pairs $\{(\mathbf{x}_i, \mathrm{y}_i)\}_{i=1}^{n}$ are identically and independently sampled from $Q_{\mathbf{x},\mathrm{y}}$, while unlabeled vectors $\{\mathbf{x}'_i\}_{i=1}^{m}$ are identically and independently sampled from $P_{\mathbf{x}}$. If the marginal distributions of features

have density functions $p_{\mathbf{x}}$ and $q_{\mathbf{x}}$, such that support$(p_{\mathbf{x}}) \subseteq$ support$(q_{\mathbf{x}})$, the most common approach to adapt a model for the target distribution is to employ an empirical error weighted by $w(\boldsymbol{x}) = p_{\mathbf{x}}(\boldsymbol{x})/q_{\mathbf{x}}(\boldsymbol{x})$ (Huang *et al.*, 2007; Kanamori *et al.*, 2009a; Shimodaira, 2000; Sugiyama and Kawanabe, 2012; Sugiyama *et al.*, 2007).

The weighting scheme may fail when the effective sample sizes (ESS) are small. According to common wisdom, a small ESS hurts model's performance in the target distribution. As previous research argues, e.g., Wang and Rudin (2017), that kind of scenario is common when working with high-dimensional data. However, to the best of our knowledge, there is no unified and rigorous view on how the three key concepts (i) effective sample size (ESS), (ii) data dimensionality, and (iii) generalization of supervised models under covariate shift are connected to each other. In this chapter, we present a unified theory connecting the three concepts. Moreover, we also explore how dimensionality reduction or feature selection can increase the effective sample size.

This chapter is organized as follows. In Section 5.2, we discuss previous results and explain our contribution to the debate. In Section 5.3, we briefly review importance weighting and introduce a new connection between effective sample size and generalization in the context of covariate shift adaptation. In Section 5.4, we introduce dimensionality to the problem showing how it connects to the other two concepts and then illustrate these connections with a toy experiment. Finally, in Section 5.5, we show how dimensionality reduction and feature selection can lead to a larger effective sample size. We conclude our discussion with real-data experiments that supports feature selection before covariate shift adaptation as a good practice.

5.2 Related Work

There is a rich literature on the problem of covariate shift adaptation[1] or related subjects. The main interest has been to develop methods to estimate the density ratio w (Huang *et al.*, 2007; Izbicki *et al.*, 2014; Kanamori *et al.*, 2009a; Liu *et al.*, 2017; Sugiyama *et al.*, 2008). Some of the proposed methods aim to reliably estimate w in high-dimensional and unstable

[1] See Sugiyama and Kawanabe (2012) for a general view.

settings (Izbicki *et al.*, 2014; Liu *et al.*, 2017), when the more traditional approaches may fail. However, according to the common wisdom of the area, even if we could perfectly estimate w, we would still have to deal with poor performance due to small effective sample sizes (ESS), especially in high-dimensional settings. Understanding the role of small ESS and possible ways to attenuate it may, therefore, be productive. The covariate shift adaptation literature has already tried to articulate the relationships between ESS and generalization in high-dimensional settings, also proposing dimensionality reduction as a cure. In spite of that, we believe these previous attempts fail in connecting these concepts in a unified manner and as explicitly as we propose to do in this chapter.

In recent years, Reddi *et al.* (2015) proposed a regularization method that controls the ESS and offers sharper generalization bounds while correcting for covariate shift. However, the authors do not explore how the number of features plays an essential role. Another work that explores the concept of ESS in the context of covariate shift adaptation is Gretton *et al.* (2009). In that work, the authors present the relationship between ESS and generalization bounds in a transductive learning scenario. Besides transductive learning not being as common as inductive learning in practice, the authors also do not explore how dimensionality plays an essential role in the problem.

The idea of features dimensionality being related to ESS is explored in Wang and Rudin (2017), without formalizing the connection to generalization. The authors also motivate how dimensionality reduction can make ESS bigger, however, the central hypothesis adopted in this case is that dimensionality reduction does not depend on the data, which, in most cases, is not valid. In a more recent paper, Stojanov *et al.* (2019) proposes a dimensionality reduction method to make covariate shift adaptation feasible, especially when estimating weights. The authors show how the number of features is indirectly related to transductive generalization bounds and effective sample size when the correction is made by Kernel Mean Matching (Huang *et al.*, 2007). In addition to the results being restricted to a particular case, the authors implicitly assume that the mapping that defines dimensionality reduction is given beforehand and does not depend on the training data, what is not realistic.

In this chapter, we complement previous works by formally articulating the relationship among ESS, generalization of predictive models in the inductive scenario, and dimensionality

as explicitly as possible. We present a unified theory connecting the three concepts, which was not observed by us in the literature. We also show that dimensionality reduction, even considering that the mapping may depend on the data, mitigates low ESS by making the source and target domains less divergent.

5.3 Effective Sample Size (ESS) and Generalization in Covariate Shift Adaptation

5.3.1 Importance Weighting

To keep our discussion as self-contained as possible, we first use this subsection to quickly summarize key points behind importance weighting.

Given a hypothesis class $\mathcal{H}$ and a loss function L, our goal is finding a hypothesis $h^* \in \mathcal{H}$ that minimizes the risk R assessed in the target distribution $P_{\mathbf{x},\mathrm{y}}$ using data from source distribution $Q_{\mathbf{x},\mathrm{y}}$. From now on we assume: (i) $Q_{\mathrm{y}|\mathbf{x}} = P_{\mathrm{y}|\mathbf{x}}$ and $Q_{\mathbf{x}} \neq P_{\mathbf{x}}$; (ii) distributions $P_{\mathbf{x}}$ and $Q_{\mathbf{x}}$ have probability density functions (p.d.f.s) $p_{\mathbf{x}}$ and $q_{\mathbf{x}}$ such that $\mathrm{support}(p_{\mathbf{x}}) \subseteq \mathrm{support}(q_{\mathbf{x}})$. Then, the risk can be written in terms of the source distribution:

$$R(h) = \mathbb{E}_{\mathbf{x}\sim P_{\mathbf{x}}}\mathbb{E}_{\mathrm{y}|\mathbf{x}}\left[L(h(\mathbf{x}),\mathrm{y})\right] \tag{5.1}$$

$$= \int \frac{p_{\mathbf{x}}(\boldsymbol{x})}{q_{\mathbf{x}}(\boldsymbol{x})} q_{\mathbf{x}}(\boldsymbol{x}) \mathbb{E}_{\mathrm{y}|\mathbf{x}}\left[L(h(\boldsymbol{x}),\mathrm{y})\right] \mathrm{d}\boldsymbol{x} \tag{5.2}$$

$$= \mathbb{E}_{\mathbf{x}\sim Q_{\mathbf{x}}}\mathbb{E}_{\mathrm{y}|\mathbf{x}}\left[w(\mathbf{x}) \cdot L(h(\mathbf{x}),\mathrm{y})\right] \tag{5.3}$$

We would like to find a hypothesis $h^{\mathrm{ERM}}_{\hat{w}} \in \mathcal{H}$ that minimizes a weighted version of the empirical risk while also obtaining a low value for R. Assume we have an estimate $\hat{w}$ for the "true" weighting function $w = p_{\mathbf{x}}/q_{\mathbf{x}}$ and that we have pairs $\{(\mathbf{x}_i, \mathrm{y}_i)\}_{i=1}^n$ that are identically and independently (i.i.d.) sampled from $Q_{\mathbf{x},\mathrm{y}}$. The weighted empirical risk is thus given by

$$\widehat{R}_{\hat{w}}(h) = \frac{1}{n}\sum_{i=1}^{n} \hat{w}(\mathbf{x}_i) \cdot L(h(\mathbf{x}_i), \mathrm{y}_i) \tag{5.4}$$

In practice, we might also want to add a regularization term $\Omega(h)$ to penalize for the complexity of the hypothesis h.

5.3.2 Relationship of Effective Sample Size (ESS) and Generalization in Covariate Shift Adaptation

To introduce the concept of effective sample size in the context of covariate shift adaptation, we first describe how this heuristic is employed within the importance sampling literature (Martino *et al.*, 2017; Owen, 2013; Robert *et al.*, 2010), where it originally comes from. We assume the "true" importance function (density ratio) is known up to a constant. This assumption enables us to achieve some theoretical results and is also adopted in previous works (Cortes *et al.*, 2010, 2019; Wang and Rudin, 2017). The strategy we use to show the relevance of the effective sample size in covariate shift adaptation is to find an asymptotic approximation for that quantity, and then connect it to a known generalization bound.

The ESS formulation we use is slightly different from the most usual one (Martino *et al.*, 2017; Owen, 2013; Robert *et al.*, 2010) in the sense we are concerned with percentage of effective samples and not with the number of effective samples[2]. Given the two definitions are not very different, the intuitions and some results regarding ESS are easily adaptable. We present our definition in the following.

Consider two probability distributions $P_{\mathbf{z}}$ and $Q_{\mathbf{z}}$ over $\mathcal{Z} \subseteq \mathbb{R}^d$ with probability density functions $p_{\mathbf{z}}$ and $q_{\mathbf{z}}$ such that $\text{support}(p_{\mathbf{z}}) \subseteq \text{support}(q_{\mathbf{z}})$. From now on, we call $P_{\mathbf{z}}$ the *target* distribution and $Q_{\mathbf{z}}$ the *source* distribution. We thus sample from $Q_{\mathbf{z}}$ in order to estimate the integral $\int_{\mathcal{Z}} g(\boldsymbol{z}) p_{\mathbf{z}}(\boldsymbol{z}) d\boldsymbol{z} = \int_{\mathcal{Z}} \frac{p_{\mathbf{z}}(\boldsymbol{z})}{q_{\mathbf{z}}(\boldsymbol{z})} g(\boldsymbol{z}) q_{\mathbf{z}}(\boldsymbol{z}) d\boldsymbol{z}$, with $g : \mathcal{Z} \to \mathbb{R}$ integrable. A key quantity in this problem is the importance function, which is given by $w \propto p_{\mathbf{z}}/q_{\mathbf{z}}$.

Suppose we have an independent and identically distributed (i.i.d.) sample $\{\mathbf{z}_i\}_{i=1}^n$ from the source distribution $Q_{\mathbf{z}}$ and we want to use the (self-normalized[3]) importance sampling estimator $n^{-1} \sum_{i=1}^n \bar{\mathrm{w}}_i g(\mathbf{z}_i)$ in order to estimate the integral of interest. The weights are

[2] In the literature, it is common to present the ESS as $n \cdot \widehat{\text{ESS}}_n$, while we are concerned only with $\widehat{\text{ESS}}_n$ (Equation 5.5).

[3] We show the case of the self-normalized estimator because it returns the most usual definition for the ESS, which is also used in the context of covariate shift (Reddi *et al.*, 2015). In spite of that, we show that this definition for the ESS is still useful for the non normalized case while performing covariate shift adaptation.

given by $\bar{\mathrm{w}}_i = \mathrm{w}_i / \sum_j \mathrm{w}_j$, where $\mathrm{w}_i = w(\mathbf{z}_i) \propto p_{\mathbf{z}}(\mathbf{z}_i)/q_{\mathbf{z}}(\mathbf{z}_i)$, $i \in [n] := \{1, ..., n\}$. Then, the effective sample size is defined as

$$\widehat{\mathrm{ESS}}_n(P_{\mathbf{z}}, Q_{\mathbf{z}}) := \frac{1}{n \sum_{i=1}^n \bar{\mathrm{w}}_i^2} \tag{5.5}$$

$$= \frac{(\sum_{i=1}^n \mathrm{w}_i)^2}{n \sum_{i=1}^n \mathrm{w}_i^2} \tag{5.6}$$

Intuitively, the effective sample size is the percentage of effective samples. For example, if the effective sample size equals $1/2$, then the importance sampling estimator *effectiveness* is the same of a monte carlo estimator with $n/2$ samples. That formulation can be used to approximate, via Delta Method, the ratio of monte carlo estimators' variance and the self-normalized importance sampling estimator' variance, using the derivation made by Elvira *et al.* (2018). While that work motivates the use of the ESS, other approaches can be derived from Owen (2013) and Martino *et al.* (2017). The latter presents the relationship between effective sample size and the euclidean distance between the vector $(\bar{\mathrm{w}}_1, ..., \bar{\mathrm{w}}_n)$ and the "ideal" balanced vector $(1/n, ..., 1/n)$. Furthermore, effective sample size informs about the importance sampling estimator's convergence rate (Agapiou *et al.*, 2017). Said that, the results presented in this section for the covariate shift adaptation case resembles the results presented by Agapiou *et al.* (2017) in a different context.

To move forward, we introduce the concept of Rényi Divergence, which plays a central role in our analysis:

Definition 5.1 (Rényi Divergence (van Erven and Harremoës, 2012))**.** Consider two probability distributions $P_{\mathbf{x}}$ and $Q_{\mathbf{x}}$ over $\mathcal{X} \subseteq \mathbb{R}^d$, with probability density functions $p_{\mathbf{x}}$ and $q_{\mathbf{x}}$ such that $\mathrm{support}(p_{\mathbf{x}}) \subseteq \mathrm{support}(q_{\mathbf{x}})$. The Rényi Divergence of order $\alpha > 1$ of $P_{\mathbf{x}}$ from $Q_{\mathbf{x}}$ is given by:

$$D_\alpha(P_{\mathbf{x}}||Q_{\mathbf{x}}) := \frac{1}{\alpha - 1} \log \mathop{\mathbb{E}}_{\mathbf{x} \sim Q_{\mathbf{x}}} \left[\left(\frac{p_{\mathbf{x}}(\mathbf{x})}{q_{\mathbf{x}}(\mathbf{x})} \right)^\alpha \right] \tag{5.7}$$

Consequently, the Rényi Divergence of order 2 of $P_{\mathbf{x}}$ from $Q_{\mathbf{x}}$ is given by $D_2(P_{\mathbf{x}}||Q_{\mathbf{x}}) = \log \mathbb{E}_{\mathbf{x} \sim P_{\mathbf{x}}}[\frac{p_{\mathbf{x}}(\mathbf{x})}{q_{\mathbf{x}}(\mathbf{x})}]$.

Despite all previous work, the question of how we should transpose the effective sample size concept to the covariate shift adaptation framework remains. In the following, we make

explicit the close relationship between the ESS and generalization bounds under covariate shift adaptation. As we start talking about covariate shift adaptation, we substitute $\mathbf{z}$ by a vector of features $\mathbf{x}$, the set $\mathcal{Z}$ by $\mathcal{X}$ or $\mathcal{X}\times\mathcal{Y}$ and the function g by the loss function L. Before we move on, we must establish that the effective sample size $\widehat{\mathrm{ESS}}_n(P_\mathbf{x}, Q_\mathbf{x})$ converges almost surely to the quantity $\mathrm{ESS}^*(P_\mathbf{x}, Q_\mathbf{x})$, which plays a central role in our analysis. $\mathrm{ESS}^*(P_\mathbf{x}, Q_\mathbf{x})$ can be considered a population version for the effective sample size. From now on, we may call it by population effective sample size or only effective sample size, when it is not ambiguous.

Theorem 5.2. *Consider two probability distributions $P_\mathbf{x}$ and $Q_\mathbf{x}$ over $\mathcal{X} \subseteq \mathbb{R}^d$, with probability density functions $p_\mathbf{x}$ and $q_\mathbf{x}$ such that support$(p_\mathbf{x}) \subseteq$ support$(q_\mathbf{x})$. Suppose we have a random sample $\{\mathbf{x}_i\}_{i=1}^n$, identically and independently sampled from the distribution $Q_\mathbf{x}$, and we define $\mathrm{w}_i = w(\mathbf{x}_i) \propto p_\mathbf{x}(\mathbf{x}_i)/q_\mathbf{x}(\mathbf{x}_i)$. Assume that $0 < \mathbb{E}_{\mathbf{x}\sim Q_\mathbf{x}}\left[w(\mathbf{x})^2\right] < \infty$. Then*

$$\widehat{\mathrm{ESS}}_n(P_\mathbf{x}, Q_\mathbf{x}) \xrightarrow[n\to\infty]{a.s.} \mathrm{ESS}^*(P_\mathbf{x}, Q_\mathbf{x}) \tag{5.8}$$

Where

$$\mathrm{ESS}^*(P_\mathbf{x}, Q_\mathbf{x}) := \exp\left[-D_2(P_\mathbf{x}||Q_\mathbf{x})\right] \tag{5.9}$$

The quantity $D_2(P_\mathbf{x}||Q_\mathbf{x})$ is the Rényi Divergence of order 2 of $P_\mathbf{x}$ from $Q_\mathbf{x}$ (van Erven and Harremoës, 2012).

Proof. Assume the hypothesis stated are valid. Being $c \neq 0$ a real constant, see we can re-wright the ESS as follows:

$$\widehat{\mathrm{ESS}}_n(P_\mathbf{x}, Q_\mathbf{x}) = \frac{\left(\sum_{i=1}^n \mathrm{w}_i\right)^2}{n\sum_{i=1}^n \mathrm{w}_i^2} = \frac{\left[\sum_{i=1}^n c\cdot\frac{p_\mathbf{x}(\mathbf{x}_i)}{q_\mathbf{x}(\mathbf{x}_i)}\right]^2}{n\sum_{i=1}^n\left[c\cdot\frac{p_\mathbf{x}(\mathbf{x}_i)}{q_\mathbf{x}(\mathbf{x}_i)}\right]^2} = \frac{\left[\frac{1}{n}\sum_{i=1}^n\frac{p_\mathbf{x}(\mathbf{x}_i)}{q_\mathbf{x}(\mathbf{x}_i)}\right]^2}{\frac{1}{n}\sum_{i=1}^n\left[\frac{p_\mathbf{x}(\mathbf{x}_i)}{q_\mathbf{x}(\mathbf{x}_i)}\right]^2}$$

Then, by the Strong Law of Large Numbers and almost-sure convergence properties (Roussas, 1997), we verify that $\widehat{\mathrm{ESS}}_n(P_\mathbf{x}, Q_\mathbf{x}) \xrightarrow{a.s.} \frac{\mathbb{E}_{\mathbf{x}\sim Q_\mathbf{x}}\left[\frac{p_\mathbf{x}(\mathbf{x})}{q_\mathbf{x}(\mathbf{x})}\right]^2}{\mathbb{E}_{\mathbf{x}\sim Q_\mathbf{x}}\left[\left(\frac{p_\mathbf{x}(\mathbf{x})}{q_\mathbf{x}(\mathbf{x})}\right)^2\right]}$ when $n \to \infty$. To complete

the proof, we state the following

$$\frac{\mathbb{E}_{\mathbf{x}\sim Q_{\mathbf{x}}}\left[\frac{p_{\mathbf{x}}(\mathbf{x})}{q_{\mathbf{x}}(\mathbf{x})}\right]^2}{\mathbb{E}_{\mathbf{x}\sim Q_{\mathbf{x}}}\left[\left(\frac{p_{\mathbf{x}}(\mathbf{x})}{q_{\mathbf{x}}(\mathbf{x})}\right)^2\right]} = \frac{1}{\mathbb{E}_{\mathbf{x}\sim P_{\mathbf{x}}}\left[\frac{p_{\mathbf{x}}(\mathbf{x})}{q_{\mathbf{x}}(\mathbf{x})}\right]} = \frac{1}{\exp\left[D_2(P_{\mathbf{x}}||Q_{\mathbf{x}})\right]} = \mathrm{ESS}^*(P_{\mathbf{x}}, Q_{\mathbf{x}})$$

□

The last theorem can be seen as a variation of some of the results presented by Agapiou *et al.* (2017). While the authors focus on related but different divergences, we choose to present this result in terms of the Rényi Divergence because, in that way, we can connect it to other results in the literature.

It is essential to state that similar results hold for other effective sample size definitions as, for example, the one used by Wang and Rudin (2017) divided by n, to give the percentage of effective samples considering the non normalized weights for covariate shift adaptation.

It is fascinating how Rényi Divergence naturally emerges when working with the effective sample size. It is a crucial point to understand that, when calculating the effective sample size, we are approximating a quantity inversely proportional to the exponential of Rényi Divergence of order 2 of $P_{\mathbf{x}}$ from $Q_{\mathbf{x}}$.

Now we focus on the understanding of how effective sample size relates to generalization of adapted supervised models. For Theorem 5.3, consider some conditions. Let $\mathcal{X}$ denote the input space, $\mathcal{Y}$ the label set, and let $L : \mathcal{Y}^2 \to [0, 1]$ be a bounded loss function. Denote the *target* distribution of features by $P_{\mathbf{x}}$ and the *source* distribution of features by $Q_{\mathbf{x}}$, such that $P_{\mathbf{x}}$ is dominated by $Q_{\mathbf{x}}$. Consider $\mathcal{H}$ to be the hypothesis class used by the learning algorithm and $f : \mathcal{X} \to \mathcal{Y}$ to be the labeling function we want to learn about. We denote by $\mathrm{Pdim}(U)$ the pseudo-dimension[4] of a real-valued function class U (Vidyasagar, 2002). Pdim is used here to quantify the complexity of a hypothesis class through the loss function. Finally, R is the risk assessed in the target distribution $P_{\mathbf{x}}$ and $\hat{R}_w$ is the weighted empirical error calculated using the true weighting function (density ratio) and samples $\{\mathbf{x}_i\}_{i=1}^n$, identically and independently sampled from the source distribution $Q_{\mathbf{x}}$.

Theorem 5.3 (Adapted from Cortes *et al.* (2010))**.** *Define the function* $L_h(\boldsymbol{x}) := L[h(\boldsymbol{x}), f(\boldsymbol{x})]$

[4] A pseudo-dimension is an extension of VC Dimension for real-valued classes of functions

and let $\mathcal{H}$ *be a hypothesis set such that* $Pdim(\{L_h : h \in \mathcal{H}\}) = p < \infty$. *Assume that* $\text{ESS}^*(P_{\mathbf{x}}, Q_{\mathbf{x}}) = \exp[-D_2(P_{\mathbf{x}}||Q_{\mathbf{x}})]$, $D_2(P_{\mathbf{x}}||Q_{\mathbf{x}}) < \infty$, *and the target/source density ratio* $w > 0$. *Then, for any* $\delta \in (0, 1)$, *with probability at least* $1 - \delta$, *we have that:*

$$\sup_{h \in \mathcal{H}} [R(h) - \hat{R}_w(h)] \leq \frac{2^{\frac{5}{4}}}{\sqrt{\text{ESS}^*(P_{\mathbf{x}}, Q_{\mathbf{x}})}} \cdot \left[\frac{p \cdot \log \frac{2 \cdot e \cdot n}{p} + \log \frac{4}{\delta}}{n} \right]^{\frac{3}{8}} \tag{5.10}$$

See Cortes *et al.* (2010) for the proof, and replace D_2 by ESS^* to get this version of the theorem.

It is clear from Theorem 5.3 that $\text{ESS}^*(P_{\mathbf{x}}, Q_{\mathbf{x}})$ plays a fundamental role when learning f from data. A larger $\text{ESS}^*(P_{\mathbf{x}}, Q_{\mathbf{x}})$ leads to a tighter generalization bound. Consequently, if $\widehat{\text{ESS}}_n(P_{\mathbf{x}}, Q_{\mathbf{x}})$ is a good approximation for $\text{ESS}^*(P_{\mathbf{x}}, Q_{\mathbf{x}})$, the rationale behind using effective sample size as a heuristic for diagnosis of covariate shift adaptation becomes clearer. To conclude, we should mention that Cortes *et al.* (2019) shows a similar result to Theorem 5.3 with less assumptions, namely, assuming the existence of a labeling function f and that $w > 0$. However, we chose the form provided by Cortes *et al.* (2010), as it gives us a more straightforward expression without losing the property that is key to our approach, to say, that a larger $\text{ESS}^*(P_{\mathbf{x}}, Q_{\mathbf{x}})$ leads to a sharper generalization bound.

5.4 The Role of Dimensionality

In Section 5.3, we showed the effective sample size's role in the context of covariate shift adaptation exploring its asymptotic relationship with generalization bounds. However, we still need to understand the role that dimensionality plays during covariate shift adaptation. In Theorem 5.4, we demonstrate that the Rényi Divergence of source and target distributions does not decrease with the number of features, and, consequently, the population effective sample size does not increase with the number of features, which explains potential adaptation problems for high-dimensional data.

Theorem 5.4. *Given two joint probability distributions* $P_{\mathbf{x}_1,\mathbf{x}_2}$ *(target) and* $Q_{\mathbf{x}_1,\mathbf{x}_2}$ *(source) over* $\mathcal{X} \subseteq \mathbb{R}^d$, $D_2(P_{\mathbf{x}_1,\mathbf{x}_2}||Q_{\mathbf{x}_1,\mathbf{x}_2}) < \infty$, *with joint probability density functions* $p_{\mathbf{x}_1,\mathbf{x}_2}$ *and*

$q_{\mathbf{x}_1,\mathbf{x}_2}$, such that support$(p_{\mathbf{x}_1,\mathbf{x}_2}) \subseteq$ support$(q_{\mathbf{x}_1,\mathbf{x}_2})$, we have that

$$D_2(P_{\mathbf{x}_1,\mathbf{x}_2}||Q_{\mathbf{x}_1,\mathbf{x}_2}) \geq D_2(P_{\mathbf{x}_1}||Q_{\mathbf{x}_1}) \tag{5.11}$$

And, consequently,

$$\text{ESS}^*(P_{\mathbf{x}_1}, Q_{\mathbf{x}_1}) \geq \text{ESS}^*(P_{\mathbf{x}_1,\mathbf{x}_2}, Q_{\mathbf{x}_1,\mathbf{x}_2}) \tag{5.12}$$

Proof. Assume the hypothesis are valid and let $d_2(P_{\mathbf{x}_1,\mathbf{x}_2}||Q_{\mathbf{x}_1,\mathbf{x}_2}) = \exp[D_2(P_{\mathbf{x}_1,\mathbf{x}_2}||Q_{\mathbf{x}_1,\mathbf{x}_2})]$. See that:

$$\begin{aligned} d_2(P_{\mathbf{x}_1,\mathbf{x}_2}||Q_{\mathbf{x}_1,\mathbf{x}_2}) &= \mathbb{E}_{P_{\mathbf{x}_1,\mathbf{x}_2}}\left[\frac{p_{\mathbf{x}_1,\mathbf{x}_2}(\mathbf{x}_1,\mathbf{x}_2)}{q_{\mathbf{x}_1,\mathbf{x}_2}(\mathbf{x}_1,\mathbf{x}_2)}\right] = \mathbb{E}_{P_{\mathbf{x}_1}}\left[\frac{p_{\mathbf{x}_1}(\mathbf{x}_1)}{q_{\mathbf{x}_1}(\mathbf{x}_1)} \cdot \mathbb{E}_{P_{\mathbf{x}_2|\mathbf{x}_1}}\left[\frac{p_{\mathbf{x}_2|\mathbf{x}_1}(\mathbf{x}_2|\mathbf{x}_1)}{q_{\mathbf{x}_2|\mathbf{x}_1}(\mathbf{x}_2|\mathbf{x}_1)}\right]\right] = \\ &= \mathbb{E}_{P_{\mathbf{x}_1}}\left[\frac{p_{\mathbf{x}_1}(\mathbf{x}_1)}{q_{\mathbf{x}_1}(\mathbf{x}_1)} \cdot d_2(P_{\mathbf{x}_2|\mathbf{x}_1}||Q_{\mathbf{x}_2|\mathbf{x}_1})\right] \geq \mathbb{E}_{P_{\mathbf{x}_1}}\left[\frac{p_{\mathbf{x}_1}(\mathbf{x}_1)}{q_{\mathbf{x}_1}(\mathbf{x}_1)}\right] = d_2(P_{\mathbf{x}_1}||Q_{\mathbf{x}_1}) \end{aligned}$$

Where the inequality is obtained by the fact that the exponential of the Rényi Divergence must be greater or equals one. To complete the proof, see that $\text{ESS}^*(P_{\mathbf{x}_1,\mathbf{x}_2}, Q_{\mathbf{x}_1,\mathbf{x}_2}) = d_2(P_{\mathbf{x}_1,\mathbf{x}_2}||Q_{\mathbf{x}_1,\mathbf{x}_2})^{-1}$. Therefore

$$\text{ESS}^*(P_{\mathbf{x}_1}, Q_{\mathbf{x}_1}) \geq \text{ESS}^*(P_{\mathbf{x}_1,\mathbf{x}_2}, Q_{\mathbf{x}_1,\mathbf{x}_2})$$

□

This theorem can be seen as a particular case of the Data Processing Inequality (Van Erven and Harremos, 2014).

Combining the results of Theorem 5.3 and Theorem 5.4, we conclude that performing covariate shift adaptation with many features may not be feasible, as we would potentially have loose generalization bounds.

Note that Theorem 5.4 does not necessarily say that by reducing dimensions or selecting the most relevant features we will have a bigger effective sample size. Reducing dimensions or selecting features is a random process that depends on data, and we have ignored this

fact so far. In Section 5.5, we consider the randomness of the dimensionality reduction or feature selection step to prove that we can increase the effective sample size by following these procedures before conducting covariate shift adaptation.

5.4.1 A Toy Experiment

In this section, we present a toy experiment in order to illustrate the relationship between effective sample size, Rényi divergence, dimensionality, and performance of supervised methods.

Assume there are two joint distributions of features and labels P_λ and Q with densities p_λ and q, being the case that Q describes the source/training population and that P_λ describes the target/test population. Moreover, we assume we are facing the classical covariate shift problem, that is, $p_\lambda(y|\boldsymbol{x}) = q(y|\boldsymbol{x}) = p(y|\boldsymbol{x})$ but $p_\lambda(\boldsymbol{x}) \neq q(\boldsymbol{x})$, plus the fact that we cannot sample the labels from the test population. Finally, consider $q(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}|\mathbf{0}, \boldsymbol{I}_d)$ and $p_\lambda(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}|\lambda \cdot \mathbf{1}, \boldsymbol{I}_d)$, for $\lambda \neq 0$, with d indicating the number of dimensions. Suppose $p(y|\boldsymbol{x}) = \mathcal{N}(y|100 \cdot x_1, 1)$, that is, y depends on $\mathbf{x}$ only through its first coordinate x_1.

Firstly, we calculate $D_2(P_\lambda||Q)$ and $\mathrm{ESS}^*(P_\lambda, Q)$ as functions of d and then simulate how the predictive power of a decision tree regressor deteriorates as d increases and $\mathrm{ESS}^*(P_\lambda, Q)$ decreases. We train the trees by minimizing the empirical error weighted by the true weighting function w in the training set, also imposing a minimum of 10 samples per leaf as a regularization strategy. We choose to work with decision trees since they are fast to train and robust against irrelevant features. Thus, it is reasonable to expect that a great part of performance deterioration is not due to noisy features but because of small ESSs.

The first step to calculate $\mathrm{ESS}^*(P_\lambda, Q)$ and $D_2(P_\lambda||Q)$ is to calculate $\exp[D_2(P_\lambda||Q)]$:

$$\exp[D_2(P_\lambda||Q)] = \mathbb{E}_{\mathbf{x}\sim P_\lambda}\left[\frac{p_\lambda(\mathbf{x})}{q(\mathbf{x})}\right] \tag{5.13}$$

$$= \mathbb{E}_{\mathbf{x}\sim P_\lambda}\left\{\frac{\exp[-\frac{1}{2}(\mathbf{x}-\lambda\mathbf{1})^\top(\mathbf{x}-\lambda\mathbf{1})]}{\exp[-\frac{1}{2}\mathbf{x}^\top\mathbf{x}]}\right\} \tag{5.14}$$

$$= \exp\left(-\frac{d\lambda^2}{2}\right) \cdot \mathbb{E}_{\mathbf{x}\sim P_\lambda}\left[\exp\left(\lambda\sum_{j=1}^{d}\mathrm{x}_j\right)\right] \tag{5.15}$$

$$= \exp(d\lambda^2) \tag{5.16}$$

The last equality is true since $\exp(\lambda \sum_{j=1}^{d} \mathrm{x}_j) \sim \text{LogNormal}(d\lambda^2, d\lambda^2)$. Then, $D_2(P_\lambda||Q) = d\lambda^2$ and $\text{ESS}^*(P_\lambda, Q) = \exp(-d\lambda^2)$.

Figure 5.1 depicts the behavior of Rényi Divergence and $\text{ESS}^*(P_\lambda, Q)$ as functions of d. We also vary the value for λ. Given that $D_2(P_\lambda||Q)$ only depends on $|\lambda|$ and not on $\text{sign}(\lambda)$, we consider the case where $\lambda > 0$. When $|\lambda|$ is bigger, the divergence between the source and target distributions also increases. Finally, to check how large d affects performance of a regressor we, for each d, (i) sample 50 training and test sets of size 10^6, (ii) train the trees on the training set minimizing the weighted empirical error and (iii) assess the regressors on the test sets. The third plot of Figure 5.1 represents the average root-mean-square test error ($\pm$ standard deviation). Clearly the regressor deteriorates as the divergence between domains grows and the ESS decreases.

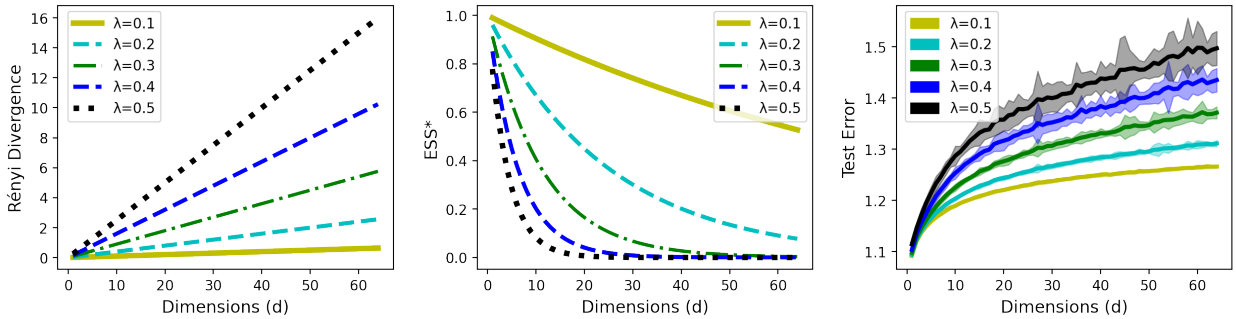

Figure 5.1: *(i) We plot the Rényi Divergence of the target distribution P_λ from the source distribution Q as a function of the number of features. Both distributions are normal with the same covariance matrix but located $\sqrt{d\lambda^2}$ units apart from each other, i.e. the divergence also depends on $|\lambda|$; (ii) We plot the $ESS^*(P_\lambda, Q)$ as a function of d and also varying λ. As expected, $ESS^*(P_\lambda, Q)$ exponentially decays in d as long as the divergence is linearly related with d; (iii) In 50 simulations for each pair (λ, d), we observe how decision trees' performances deteriorate as the divergence between domains grows and the ESS decreases.*

5.5 The use of dimensionality reduction/feature selection to make effective sample size bigger

In this section, we present dimensionality reduction and feature selection as ways to obtain a bigger effective sample size. The two main results of this section are given by Theorems 5.6 and 5.7. We show that linear dimensionality reduction and feature selection, under some conditions, decrease Rényi divergence between the target and source probability

distributions, leading to a bigger effective sample size. This result accounts for the dimensionality reduction or feature selection's randomness; that is, the transformation can depend on data in some specific ways.

To arrive at our main results, we first show the intermediate result given by Lemma 5.5. In the following result, $\boldsymbol{A}$ represents a constant dimensionality reduction matrix and the vector $\boldsymbol{b}$ represents a translation in data before dimensionality reduction, which is common when performing principal components analysis (PCA) (Hastie *et al.*, 2009), for example. When there is no need for considering a translation, we just can adopt $\boldsymbol{b} = \mathbf{0}$. Also, $\boldsymbol{A}$ can represent a feature selector, as we explain in the coming paragraphs.

Lemma 5.5. Consider (i) two absolutely continuous random vectors $\mathbf{x} \sim Q_{\mathbf{x}}$ and $\mathbf{x}' \sim P_{\mathbf{x}}$ of size $d \geq 2$, $D_2(P_{\mathbf{x}}||Q_{\mathbf{x}}) < \infty$, (ii) a nonrandom constant vector $\boldsymbol{b} \in \mathbb{R}^d$, and (iii) a nonrandom constant matrix $\boldsymbol{A} \in \mathbb{R}^{d' \times d}$ with rank d' (and $d' \leq d$). Suppose $Q_{\mathbf{x}}$ and $P_{\mathbf{x}}$ measure events in $\mathcal{X} \subseteq \mathbb{R}^d$, $d \geq 2$, and have probability density functions $q_{\mathbf{x}}$ and $p_{\mathbf{x}}$, such that $\text{support}(p_{\mathbf{x}}) \subseteq \text{support}(q_{\mathbf{x}})$. Also, assume $\boldsymbol{A}(\mathbf{x}-\boldsymbol{b}) \sim Q_{\boldsymbol{A}(\mathbf{x}-\boldsymbol{b})}$ and $\boldsymbol{A}(\mathbf{x}'-\boldsymbol{b}) \sim P_{\boldsymbol{A}(\mathbf{x}-\boldsymbol{b})}$. Then

$$D_2(P_{\mathbf{x}}||Q_{\mathbf{x}}) \geq D_2(P_{\boldsymbol{A}(\mathbf{x}-\boldsymbol{b})}||Q_{\boldsymbol{A}(\mathbf{x}-\boldsymbol{b})}) \tag{5.17}$$

And, consequently,

$$\text{ESS}^*(P_{\boldsymbol{A}(\mathbf{x}-\boldsymbol{b})}, Q_{\boldsymbol{A}(\mathbf{x}-\boldsymbol{b})}) \geq \text{ESS}^*(P_{\mathbf{x}}, Q_{\mathbf{x}}) \tag{5.18}$$

Proof. If $d = d'$, the result is direct, considering the arguments used by Qiao and Minematsu (2010) to prove[5] their Theorem 1, because $\boldsymbol{A}$ represents an invertible linear (and differentiable) transformation. Otherwise, consider a full rank matrix $\boldsymbol{C} = \begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{B} \end{bmatrix} \in \mathbb{R}^{d \times d}$, where $\boldsymbol{B} \in \mathbb{R}^{d-d',d}$. Given that $\boldsymbol{C}$ is full rank, it represents an invertible linear (and differentiable) transformation. If $\boldsymbol{C}(\mathbf{x}-\boldsymbol{b}) = \begin{bmatrix} \boldsymbol{A}(\mathbf{x}-\boldsymbol{b}) \\ \boldsymbol{B}(\mathbf{x}-\boldsymbol{b}) \end{bmatrix} \sim Q_{\boldsymbol{C}(\mathbf{x}-\boldsymbol{b})}$ and $\boldsymbol{C}(\mathbf{x}'-\boldsymbol{b}) = \begin{bmatrix} \boldsymbol{A}(\mathbf{x}'-\boldsymbol{b}) \\ \boldsymbol{B}(\mathbf{x}'-\boldsymbol{b}) \end{bmatrix} \sim$

[5]Even though D_2 is not an f-divergence, the thoughts presented by Qiao and Minematsu (2010) in their proof can readily be applied in this case. Furthermore, we can write $D_2(P_{\mathbf{x}}||Q_{\mathbf{x}}) = \log(\chi^2(P_{\mathbf{x}}||Q_{\mathbf{x}}) + 1)$, where χ^2 is a f-divergence (Sason and Verdú, 2016). This is an another reason on why this is valid.

$P_{\boldsymbol{C}(\mathbf{x}'-\boldsymbol{b})}$, then by the arguments used by Qiao and Minematsu (2010) to prove[5] their Theorem 1, we have that $D_2(P_{\mathbf{x}}||Q_{\mathbf{x}}) = D_2(P_{\boldsymbol{C}(\mathbf{x}-\boldsymbol{b})}||Q_{\boldsymbol{C}(\mathbf{x}-\boldsymbol{b})})$. Discarding $\boldsymbol{B}(\mathbf{x}-\boldsymbol{b})$ and $\boldsymbol{B}(\mathbf{x}'-\boldsymbol{b})$ from random vectors $\boldsymbol{C}(\mathbf{x}-\boldsymbol{b})$ and $\boldsymbol{C}(\mathbf{x}'-\boldsymbol{b})$, by Theorem 5.4, we have that

$$D_2(P_{\mathbf{x}}||Q_{\mathbf{x}}) \geq D_2(P_{\boldsymbol{A}(\mathbf{x}-\boldsymbol{b})}||Q_{\boldsymbol{A}(\mathbf{x}-\boldsymbol{b})})$$

Therefore

$$\mathrm{ESS}^*(P_{\boldsymbol{A}(\mathbf{x}-\boldsymbol{b})}, Q_{\boldsymbol{A}(\mathbf{x}-\boldsymbol{b})}) = \exp\left[-D_2(P_{\boldsymbol{A}(\mathbf{x}-\boldsymbol{b})}||Q_{\boldsymbol{A}(\mathbf{x}-\boldsymbol{b})})\right] \geq \exp\left[-D_2(P_{\mathbf{x}}||Q_{\mathbf{x}})\right] = \mathrm{ESS}^*(P_{\mathbf{x}}, Q_{\mathbf{x}})$$

□

Like Theorem 5.4, this result can be seen as a particular case of the Data Processing Inequality (Van Erven and Harremos, 2014).

Although Lemma 5.5 gives us a way out in cases which the dimensionality reduction is not random, this case is not realistic. We know that, in practice, $\boldsymbol{A}$ and $\boldsymbol{b}$ are obtained using data.

In the next results, linear dimensionality reduction and feature selection are represented by the random matrix $\mathbf{A}$. If we assume in advance that $\mathbf{A}$ is absolutely continuous, then it represents an ordinary dimensionality reduction matrix. On the other hand, if $\mathbf{A}$ is composed of zeros except for a single entry in each of its columns, which is given by one, then it represents a feature selector. Also, we can consider a random data translator $\mathbf{b}$ instead of the deterministic $\boldsymbol{b}$.

Theorem 5.6 (Linear dimensionality reduction)**.** *Firstly, consider the training random samples of absolutely continuous vectors* $\{\mathbf{x}_i\}_{i=1}^n \overset{iid}{\sim} Q_{\mathbf{x}}$ *and an absolutely continuous random vector from target domain* $\mathbf{x}' \sim P_{\mathbf{x}}$*. Assume* $Q_{\mathbf{x}}$ *and* $P_{\mathbf{x}}$ *measure events in* $\mathcal{X} \subseteq \mathbb{R}^d$*,* $d \geq 2$*, and have probability density functions* $q_{\mathbf{x}}$ *and* $p_{\mathbf{x}}$*, such that* $support(p_{\mathbf{x}}) \subseteq support(q_{\mathbf{x}})$*. Also, assume that* $D_2(P_{\mathbf{x}}||Q_{\mathbf{x}}) < \infty$*. Secondly, consider an absolutely continuous random vector* $\mathbf{b} \in \mathbb{R}^d$ *and an absolutely continuous random matrix* $\mathbf{A} \in \mathbb{R}^{d' \times d}$*,* $rank(\boldsymbol{A}) = d'$*, jointly distributed according to the p.d.f.* $p_{\mathbf{b},\mathbf{A}}$*, such that* $(\mathbf{b}, \mathbf{A}), \mathbf{x}_i$*, and* $\mathbf{x}'$ *are pairwise independent, for every* $i \in [n]$*. Assume that* $d' \leq d$*. Suppose* $\mathbf{A}(\mathbf{x}_i - \mathbf{b}) \sim Q_{\mathbf{A}(\mathbf{x}-\mathbf{b})}$ *and* $\mathbf{A}(\mathbf{x}' - \mathbf{b}) \sim P_{\mathbf{A}(\mathbf{x}-\mathbf{b})}$*,*

for every $i \in [n]$, then

$$D_2(P_{\mathbf{x}}||Q_{\mathbf{x}}) \geq D_2(P_{\mathbf{A}(\mathbf{x}-\mathbf{b})}||Q_{\mathbf{A}(\mathbf{x}-\mathbf{b})}) \tag{5.19}$$

And, consequently,

$$\mathrm{ESS}^*(P_{\mathbf{A}(\mathbf{x}-\mathbf{b})}, Q_{\mathbf{A}(\mathbf{x}-\mathbf{b})}) \geq \mathrm{ESS}^*(P_{\mathbf{x}}, Q_{\mathbf{x}}) \tag{5.20}$$

Proof. Firstly, we define $\mathbf{v} := \mathbf{A}(\mathbf{x}_i - \mathbf{b}) \sim Q_{\mathbf{v}} \equiv Q_{\mathbf{A}(\mathbf{x}-\mathbf{b})}$ and $\mathbf{u} := \mathbf{A}(\mathbf{x}' - \mathbf{b}) \sim P_{\mathbf{u}} \equiv P_{\mathbf{A}(\mathbf{x}-\mathbf{b})}$, for an arbitrary $i \in [n]$. Let $q_{\mathbf{v}}$ and $p_{\mathbf{u}}$ be probability density functions associated with distributions $Q_{\mathbf{v}}$ and $P_{\mathbf{u}}$. From Lemma 5.5, we know that $D_2\left(P_{\mathbf{u}|\mathbf{b}=\boldsymbol{b},\mathbf{A}=\boldsymbol{A}}||Q_{\mathbf{v}|\mathbf{b}=\boldsymbol{b},\mathbf{A}=\boldsymbol{A}}\right) \leq D_2\left(P_{\mathbf{x}}||Q_{\mathbf{x}}\right), \forall \boldsymbol{b} \in \mathbb{R}^d$, $\forall \boldsymbol{A} \in \mathbb{R}^{d' \times d}$ such that $\mathrm{rank}(\boldsymbol{A}) = d'$. That statement implies the following:

$$D_2\left(P_{\mathbf{u}|\mathbf{b}=\boldsymbol{b},\mathbf{A}=\boldsymbol{A}}||Q_{\mathbf{v}|\mathbf{b}=\boldsymbol{b},\mathbf{A}=\boldsymbol{A}}\right) \leq D_2\left(P_{\mathbf{x}}||Q_{\mathbf{x}}\right) \Rightarrow$$

$$\Rightarrow \exp D_2\left(P_{\mathbf{u}|\mathbf{b}=\boldsymbol{b},\mathbf{A}=\boldsymbol{A}}||Q_{\mathbf{v}|\mathbf{b}=\boldsymbol{b},\mathbf{A}=\boldsymbol{A}}\right) \leq \exp D_2\left(P_{\mathbf{x}}||Q_{\mathbf{x}}\right) \Rightarrow$$

$$\Rightarrow \mathbb{E}_{p_{\mathbf{b},\mathbf{A}}}\left[\exp D_2\left(P_{\mathbf{u}|\mathbf{b},\mathbf{A}}||Q_{\mathbf{v}|\mathbf{b},\mathbf{A}}\right)\right] \leq \exp D_2\left(P_{\mathbf{x}}||Q_{\mathbf{x}}\right) \Rightarrow$$

$$\Rightarrow \int p_{\mathbf{b},\mathbf{A}}(\boldsymbol{b},\boldsymbol{A}) \int p_{\mathbf{u}|\mathbf{b},\mathbf{A}}(\boldsymbol{u}|\boldsymbol{b},\boldsymbol{A}) \frac{p_{\mathbf{u}|\mathbf{b},\mathbf{A}}(\boldsymbol{u}|\boldsymbol{b},\boldsymbol{A})}{q_{\mathbf{v}|\mathbf{b},\mathbf{A}}(\boldsymbol{u}|\boldsymbol{b},\boldsymbol{A})} \mathrm{d}\boldsymbol{u}\mathrm{d}\boldsymbol{b}\mathrm{d}\boldsymbol{A} \leq \exp D_2\left(P_{\mathbf{x}}||Q_{\mathbf{x}}\right) \Rightarrow$$

$$\Rightarrow \int p_{\mathbf{u}|\mathbf{b},\mathbf{A}}(\boldsymbol{u}|\boldsymbol{b},\boldsymbol{A}) p_{\mathbf{b},\mathbf{A}}(\boldsymbol{b},\boldsymbol{A}) \frac{p_{\mathbf{u}|\mathbf{b},\mathbf{A}}(\boldsymbol{u}|\boldsymbol{b},\boldsymbol{A})}{q_{\mathbf{v}|\mathbf{b},\mathbf{A}}(\boldsymbol{u}|\boldsymbol{b},\boldsymbol{A})} \frac{p_{\mathbf{b},\mathbf{A}}(\boldsymbol{b},\boldsymbol{A})}{p_{\mathbf{b},\mathbf{A}}(\boldsymbol{b},\boldsymbol{A})} \mathrm{d}\boldsymbol{u}\mathrm{d}\boldsymbol{b}\mathrm{d}\boldsymbol{A} \leq \exp D_2\left(P_{\mathbf{x}}||Q_{\mathbf{x}}\right) \Rightarrow$$

$$\Rightarrow D_2\left(P_{\mathbf{u},\mathbf{b},\mathbf{A}}||Q_{\mathbf{v},\mathbf{b},\mathbf{A}}\right) \leq D_2\left(P_{\mathbf{x}}||Q_{\mathbf{x}}\right) \Rightarrow$$

$$\Rightarrow D_2\left(P_{\mathbf{A}(\mathbf{x}-\mathbf{b})}||Q_{\mathbf{A}(\mathbf{x}-\mathbf{b})}\right) = D_2\left(P_{\mathbf{u}}||Q_{\mathbf{v}}\right) \leq D_2\left(P_{\mathbf{u},\mathbf{b},\mathbf{A}}||Q_{\mathbf{v},\mathbf{b},\mathbf{A}}\right) \leq D_2\left(P_{\mathbf{x}}||Q_{\mathbf{x}}\right)$$

The last step is due to Theorem 5.4 (extending to random matrices). To complete the

proof, we state the following:

$$\mathrm{ESS}^*(P_{\mathbf{A}(\mathbf{x}-\mathbf{b})}, Q_{\mathbf{A}(\mathbf{x}-\mathbf{b})}) = \exp\left[-D_2(P_{\mathbf{A}(\mathbf{x}-\mathbf{b})}||Q_{\mathbf{A}(\mathbf{x}-\mathbf{b})})\right] \geq \exp\left[-D_2(P_{\mathbf{x}}||Q_{\mathbf{x}})\right] = \mathrm{ESS}^*(P_{\mathbf{x}}, Q_{\mathbf{x}})$$

□

Theorem 5.6 tells us that a dimensionality reduction procedure before performing covariate shift adaptation increases the population effective sample size. It is important to state that Theorem 5.6 also holds when disconsidering $\mathbf{b}$ and the proof's adaptation is straightforward. In that case, we would have that $D_2(P_{\mathbf{x}}||Q_{\mathbf{x}}) \geq D_2(P_{\mathbf{Ax}}||Q_{\mathbf{Ax}})$ and $\mathrm{ESS}^*(P_{\mathbf{Ax}}, Q_{\mathbf{Ax}}) \geq \mathrm{ESS}^*(P_{\mathbf{x}}, Q_{\mathbf{x}})$.

Next, in Theorem 5.7, we state a result regarding feature selection.

Theorem 5.7 (Feature selection)**.** *Firstly, consider the training random samples of absolutely continuous vectors* $\{\mathbf{x}_i\}_{i=1}^n \overset{iid}{\sim} Q_{\mathbf{x}}$ *and an absolutely continuous random vector from target domain* $\mathbf{x}' \sim P_{\mathbf{x}}$*. Assume* $Q_{\mathbf{x}}$ *and* $P_{\mathbf{x}}$ *measure events in* $\mathcal{X} \subseteq \mathbb{R}^d$*,* $d \geq 2$*, and have probability density functions* $q_{\mathbf{x}}$ *and* $p_{\mathbf{x}}$*, such that support*$(p_{\mathbf{x}}) \subseteq$ *support*$(q_{\mathbf{x}})$*. Also, assume that* $D_2(P_{\mathbf{x}}||Q_{\mathbf{x}}) < \infty$*. Secondly, consider a discrete random matrix* $\mathbf{A} \in \mathbb{R}^{d' \times d}$*, that represents a feature selector with rank*$(\boldsymbol{A}) = d'$*, distributed according to the probability mass function (p.m.f.)* $p_{\mathbf{A}}$*, such that* $\mathbf{A}, \mathbf{x}_i$*, and* $\mathbf{x}'$ *are pairwise independent, for every* $i \in [n]$*. Assume that* $d' \leq d$*. Suppose* $\mathbf{Ax}_i \sim Q_{\mathbf{Ax}}$ *and* $\mathbf{Ax}' \sim P_{\mathbf{Ax}}$*, for every* $i \in [n]$*, then*

$$D_2(P_{\mathbf{x}}||Q_{\mathbf{x}}) \geq D_2(P_{\mathbf{Ax}}||Q_{\mathbf{Ax}}) \tag{5.21}$$

And, consequently,

$$\mathrm{ESS}^*(P_{\mathbf{Ax}}, Q_{\mathbf{Ax}}) \geq \mathrm{ESS}^*(P_{\mathbf{x}}, Q_{\mathbf{x}}) \tag{5.22}$$

Proof. Firstly, we define $\mathbf{v} := \mathbf{Ax}_i \sim Q_{\mathbf{v}} \equiv Q_{\mathbf{Ax}}$ and $\mathbf{u} := \mathbf{Ax}' \sim P_{\mathbf{u}} \equiv P_{\mathbf{Ax}}$, for an arbitrary $i \in [n]$. Let $q_{\mathbf{v}}$ and $p_{\mathbf{u}}$ be probability density functions associated with distributions $Q_{\mathbf{v}}$ and $P_{\mathbf{u}}$. From Lemma 5.5, we know that $D_2\left(P_{\mathbf{u}|\mathbf{A}=\boldsymbol{A}}||Q_{\mathbf{v}|\mathbf{A}=\boldsymbol{A}}\right) \leq D_2\left(P_{\mathbf{x}}||Q_{\mathbf{x}}\right)$, $\forall \boldsymbol{A} \in \mathbb{R}^{d' \times d}$ such

that $\text{rank}(\boldsymbol{A}) = d'$. That statement implies the following:

$$D_2\left(P_{\mathbf{u}|\mathbf{A}=\boldsymbol{A}}||Q_{\mathbf{v}|\mathbf{A}=\boldsymbol{A}}\right) \leq D_2\left(P_{\mathbf{x}}||Q_{\mathbf{x}}\right) \Rightarrow \exp D_2\left(P_{\mathbf{u}|\mathbf{A}=\boldsymbol{A}}||Q_{\mathbf{v}|\mathbf{A}=\boldsymbol{A}}\right) \leq \exp D_2\left(P_{\mathbf{x}}||Q_{\mathbf{x}}\right) \Rightarrow$$

$$\Rightarrow \mathbb{E}_{p_{\mathbf{A}}}\left[\exp D_2\left(P_{\mathbf{u}|\mathbf{A}}||Q_{\mathbf{v}|\mathbf{A}}\right)\right] \leq \exp D_2\left(P_{\mathbf{x}}||Q_{\mathbf{x}}\right) \Rightarrow$$

$$\Rightarrow \sum_{\boldsymbol{A}} p_{\mathbf{A}}(\boldsymbol{A}) \int p_{\mathbf{u}|\mathbf{A}}(\boldsymbol{u}|\boldsymbol{A}) \frac{p_{\mathbf{u}|\mathbf{A}}(\boldsymbol{u}|\boldsymbol{A})}{q_{\mathbf{v}|\mathbf{A}}(\boldsymbol{u}|\boldsymbol{A})} \mathrm{d}\boldsymbol{u} \leq \exp D_2\left(P_{\mathbf{x}}||Q_{\mathbf{x}}\right) \Rightarrow$$

$$\Rightarrow \sum_{\boldsymbol{A}} \int p_{\mathbf{u}|\mathbf{A}}(\boldsymbol{u}|\boldsymbol{A}) p_{\mathbf{A}}(\boldsymbol{A}) \frac{p_{\mathbf{u}|\mathbf{A}}(\boldsymbol{u}|\boldsymbol{A})}{q_{\mathbf{v}|\mathbf{A}}(\boldsymbol{u}|\boldsymbol{A})} \frac{p_{\mathbf{A}}(\boldsymbol{A})}{p_{\mathbf{A}}(\boldsymbol{A})} \mathrm{d}\boldsymbol{u} \leq \exp D_2\left(P_{\mathbf{x}}||Q_{\mathbf{x}}\right) \Rightarrow$$

$$\Rightarrow D_2\left(P_{\mathbf{u},\mathbf{A}}||Q_{\mathbf{v},\mathbf{A}}\right) \leq D_2\left(P_{\mathbf{x}}||Q_{\mathbf{x}}\right) \Rightarrow$$

$$\Rightarrow D_2\left(P_{\mathbf{A}\mathbf{x}}||Q_{\mathbf{A}\mathbf{x}}\right) = D_2\left(P_{\mathbf{u}}||Q_{\mathbf{v}}\right) \leq D_2\left(P_{\mathbf{u},\mathbf{A}}||Q_{\mathbf{v},\mathbf{A}}\right) \leq D_2\left(P_{\mathbf{x}}||Q_{\mathbf{x}}\right)$$

Given the matrix $\boldsymbol{A}$ represents a feature selector, it can only assume a finite number of values. Thus, the sum is given over a finite number of values of $\boldsymbol{A}$. The last step is due to the Theorem 5.4 (extending to random matrices). To complete the proof, we state the following:

$$\text{ESS}^*(P_{\mathbf{A}\mathbf{x}}, Q_{\mathbf{A}\mathbf{x}}) = \exp\left[-D_2(P_{\mathbf{A}\mathbf{x}}||Q_{\mathbf{A}\mathbf{x}})\right] \geq \exp\left[-D_2(P_{\mathbf{x}}||Q_{\mathbf{x}})\right] = \text{ESS}^*(P_{\mathbf{x}}, Q_{\mathbf{x}})$$

□

Theorems 5.6 and 5.7 hold when the data used to obtain $\mathbf{A}$ and $\mathbf{b}$ do not depend on training data that will be used to train the supervised models or data points that represent the target domain we want to make generalizations for. That does not mean we cannot use some portion of the dataset to obtain $\mathbf{A}$ and $\mathbf{b}$, but it only means the results are not valid for those specific used data points, being from source or target domains.

Before closing this section, it is worth mentioning three points. Firstly, at the same time dimensionality reduction/feature selection solve the problem of low effective sample sizes, it might impose other problems. For example, when performing principal components analysis (PCA) (Hastie *et al.*, 2009) for dimensionality reduction, it is not guaranteed the method

will not discard useful information for the supervised task. Also, it is not even possible to ensure the covariate shift main assumption, that the conditional distribution of the labels are the same in source and target domains, still holds. In this direction, Stojanov *et al.* (2019) offers a clever solution to overcome these specific problems, applying sufficient dimension reduction (SDR), which is a supervised method, to reduce dimensions. Secondly, given that $\mathbf{A}$ and $\mathbf{b}$ are random quantities[6], $\{\mathbf{A}(\mathbf{x}_i - \mathbf{b})\}_{i=1}^n$ or $\{\mathbf{A}\mathbf{x}_i\}_{i=1}^n$ may not form independent samples, even when $\mathbf{x}_i \perp\!\!\!\perp (\mathbf{A}, \mathbf{b}), \forall i \in [n]$, and $\{\mathbf{x}_i\}_{i=1}^n \overset{iid}{\sim} Q_{\mathbf{x}}$. If samples are not independent, then the results presented in Section 5.3 might not hold. Finally, it is true that the results presented in this section can be extended to include more general dimensionality reduction transformations, i.e. non-linear transformations, and the validity of other transformations might be proven using the Data Processing Inequality (Van Erven and Harremos, 2014). Unfortunately, exploring the two last points is beyond the scope of the present chapter and might be treated in future work.

5.6 Numerical experiments with real data

In this section, we present regression and classification experiments in which we perform feature selection before covariate shift adaptation. When designing the experiments, we choose to work with the least possible number of assumptions, searching for evidence that the theoretical results presented so far can be extended to more general cases, which will be treated in future work. Namely, we did not assume (i) the true importance function is always known, (ii) that training data is independent of the feature selector, and (iii) that training data are formed with independent data points after the feature selection procedure.

For the following experiments, 10 regression datasets with no missing values were selected[7]. Each experiment consisted of (i) introducing covariate shift[8], (ii) estimating the weights, (iii) correcting the shift by the importance weighting method, and finally (iv) assessing the performance of the predictors and the effective sample size. We also studied the classification

[6] This is not true when $\mathbf{A}$ and $\mathbf{b}$ are fixed.

[7] From www.dcc.fc.up.pt/~ltorgo/Regression/DataSets.html and https://archive.ics.uci.edu/ml/datasets.php.

[8] Similarly to previous research, e.g., (Huang *et al.*, 2007; Reddi *et al.*, 2015; Stojanov *et al.*, 2019; Wang and Rudin, 2017).

case by binarizing the target variables using their medians as a threshold. We use the same datasets for both regression and classification experiments to make comparisons easier. For each one of the 10 datasets, we repeated the following preprocessing steps: (i) we kept up to 8,000 data points per dataset[9], (ii) generated new features using independent standard gaussian noise and (iii) standardized each column in every dataset. By augmenting the dataset to 32 features using noise, we can explore a scenario in which performance deterioration is mainly due to small effective sample sizes. We give more details on this point in the next paragraph.

The following procedure is used to create divergent training and test sets after the preprocessing steps. For each of the datasets, we sampled a sequence of vectors uniformly from $[-1, 1]^d$. We projected the data points onto the subspace generated by each vector, resulting in only one feature $\boldsymbol{x}_i^{(j)}$ per sample i for each subspace/simulation j. For each $\boldsymbol{x}_i^{(j)}$, we calculated the score $s_{ij} = \Phi\big([x_i^{(j)} - \text{median}(\boldsymbol{x}^{(j)})]/\sigma_j\big)$, which is the probability that the data point i from simulation j is in the training set. According to that score, we randomly allocated each data point in either the training or test set in simulation j. The constant σ_j was adjusted until the empirical effective sample size, as defined in Section 5.3, is less than 0.01. Following this procedure, the training and test sets are approximately of the same sizes in each simulation j. Then, we fit two decision trees for each of the training/test sets: one in the training set and one in a subset of the test set. Then, we tested both decision trees in the unused portion of the test set and compared their performance according to the mean squared error for regression and classification error (1 - accuracy) for classification. We selected the 75 simulations[10] in which decision trees trained in the test sets did best, relatively to the training set trees. We chose decision trees because they are fast to train and robust against irrelevant features. Thus, the noisy features added in the datasets are not likely to directly affect predictive power but only by making the effective sample size smaller. It is important to state that, during the whole experimenting phase, decisions trees were 2-fold cross-validated in order to choose the minimum number of samples per leaf[11].

[9]The datasets "Abalone," "Delta Ailerons," and "Wine Quality" had 4177, 7129, and 6497 data points, respectively. All the others were undersampled to have 8,000 data points.

[10]From the total of 7,200 simulations.

[11]More details on hyperparameter tuning can be found in Section 5.6.1

For the feature selection step, we were inspired by Stojanov *et al.* (2019) and the idea of Sufficient Dimension Reduction (Suzuki and Sugiyama, 2010), which is a supervised approach to dimensionality reduction and feature selection, contrasting to Principal Component Analysis, for example. Supervised approaches to dimensionality reduction and feature selection are preferable since we are able to keep important information for a supervised task performed afterwards. Using training data, we apply a combination of the methods described by Eirola *et al.* (2014); Lan *et al.* (2006) and the *Forward Selection* algorithm (Guyon and Elisseeff, 2003). The approach uses gaussian mixture models (GMMs) to estimate, using the whole training set, the mutual information between a subset of features and the target variable. In this case, the number of GMMs' components are chosen evenly splitting the training data and performing a simple holdout set hyperparameter tuning phase[12]. After training the GMMs, the procedure follows these steps: we start by choosing the feature that has the largest estimated mutual information with the target variable, and, at each subsequent step, we select the feature that marginally maximizes the estimated mutual information of target variable and selected features. We repeat the process until we reach a stop criteria. Our stopping criteria is that we should stop selecting features when the marginal improvement in the empirical mutual information is less than 1% relative to the last level or when we select the first 15 features. An implementation of the feature selection method is available in the Python package *InfoSelect*[13].

To estimate the weighting function for covariate shift adaptation, we use the probabilistic classification approach (Sugiyama and Kawanabe, 2012; Sugiyama *et al.*, 2012b) with a logistic regression model combined with a quadratic polynomial expansion of the original features. We choose to work with this approach since it is simple and fast to implement, besides being promising for high-dimensional settings. Others approaches are possible though (Sugiyama and Kawanabe, 2012). In order to prepare the data for training the logistic regression model, we first append the whole training set and randomly select rows (80%) from the test set, and create the artificial labels for both groups. Then, we randomly/evenly split that dataset in order to choose the best value for the $l1$ regularization hyperparameter

[12]More details on hyperparameter tuning can be found in Section 5.6.1

[13]See https://github.com/felipemaiapolo/infoselect or https://pypi.org/project/infoselect/

of the logistic regression, using the simple holdout validation approach[14]. After getting the optimal values for the hyperparameter, we train a final model using the whole appended dataset.

In the experiments, we work with four training scenarios. In the first one, we use the whole set of features and no weighting method. In the second one, we use the entire set of features and importance weighting combined with the "true" weights $(1 - s_{ij})/s_{ij}$. In the third, we use the whole set of features and estimated weights using the probabilistic classification approach. In the fourth scenario, we use only selected features and estimated weights using the probabilistic classification approach. Comparing the four scenarios enables us to see how importance weighting may fail in high-dimensional settings due to low ESS, even when we know the "true" weighting function.

Table 5.1 shows, for each one of the employed datasets, (i) the original number of features, (ii) the augmented number of features, (iii) the average number ($\pm$ standard deviation) of selected features for the regression and (iv) classification experiments.

Dataset	Original	Augmented	Selected (Reg.)	Selected (Class.)
abalone	7	32	4.19 ± 1.26	9.87 ± 5.64
ailerons	40	40	5.16 ± 0.54	3.79 ± 0.64
bank32nh	32	32	10.00 ± 1.82	13.91 ± 0.61
cal housing	8	32	5.29 ± 1.29	7.45 ± 4.92
cpu act	21	32	9.88 ± 1.20	2.56 ± 0.72
delta ailerons	5	32	3.16 ± 0.49	3.75 ± 0.63
elevators	18	32	7.97 ± 1.11	13.08 ± 2.16
fried delve	10	32	4.45 ± 0.50	5.00 ± 0.00
puma32H	32	32	1.88 ± 0.32	14.00 ± 0.00
winequality	11	32	9.60 ± 1.02	14.00 ± 0.00

Table 5.1: *Average Numbers of features ($\pm$ standard deviation) - in this table we compare the numbers of original, augmented and selected features for regression (reg.) and classification (class.) tasks. It is possible to note that, on average, we select small subsets of features, even smaller than the original sets.*

In Figure 5.2, one can see the distribution of effective sample sizes in all the weighted approaches, calculated in the entire set of experiments. It is possible to notice how small the ESSs can be by adopting the pure weighting strategy. The feature selection step allows bigger ESSs.

In Table 5.2, we see the average test errors ($\pm$ standard deviation). To compute the errors,

[14]More details on hyperparameter tuning can be found in Section 5.6.1

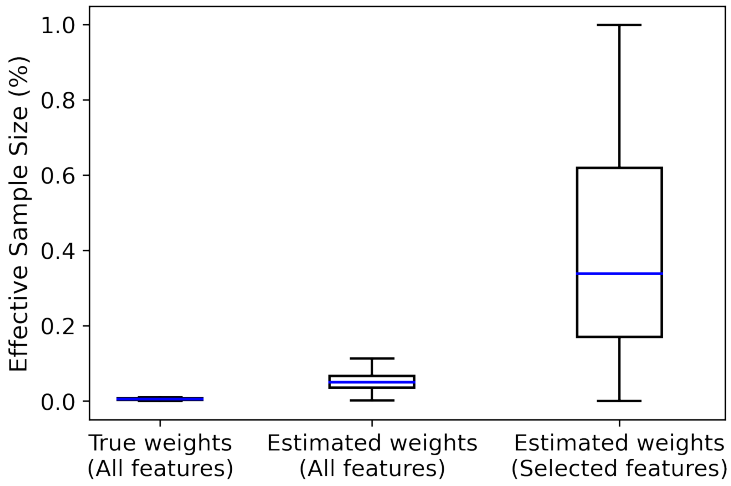

Figure 5.2: *Effective Sample Size distributions across all experiments. Notice higher ESSs can be achieved by a prior feature selection stage.*

we use the test set portion (20%) not used to train the importance function. The errors reported are the mean squared error and classification error relative to the first scenario. From Table 5.2, it is noticeable that our feature selection approach and posterior weighting systematically outperforms all the other benchmarks, especially the pure weighting method when the whole set of features is used. Even the benchmarks that used true weights are often beaten by large margins. That suggests that the degradation in the model performances is mainly due to small effective sample sizes instead of difficulties estimating the weighting function.

Through our experiments, we were able to verify that the feature selection stage tends to increase the effective sample size, consequently allowing better performance of supervised methods.

5.6.1 Some details of the experiments

In the experiments section, we tune three hyperparameters: (i) $l1$ regularization parameter used to train the logistic regression model when estimating w, (ii) the minimum number of samples per leaf in each regression/classification tree, and (iii) number of GMM components. We use the Scikit-Learn (Pedregosa *et al.*, 2011) implementations to train the logistic regressions, regression/classification trees and GMMs. Firstly, we choose the $l1$ logistic regression regularization parameter C from values in $[10^{-4}, 5]$, in order to minimize the log loss in a holdout dataset.

		All features			Selected features
	Dataset	Unweighted	True weights	Estimated weights	Estimated weights
Regression	abalone	1.00	1.42 ± 0.24	1.25 ± 0.19	$\mathbf{0.92 \pm 0.07}$
	ailerons	1.00	1.01 ± 0.13	0.99 ± 0.11	$\mathbf{0.87 \pm 0.11}$
	bank32nh	1.00	1.29 ± 0.14	1.20 ± 0.11	$\mathbf{0.98 \pm 0.06}$
	cal housing	1.00	1.50 ± 0.24	1.35 ± 0.20	$\mathbf{0.84 \pm 0.09}$
	cpu act	1.00	0.52 ± 0.55	0.55 ± 0.59	$\mathbf{0.15 \pm 0.21}$
	delta ailerons	1.00	1.39 ± 0.18	1.25 ± 0.12	$\mathbf{0.92 \pm 0.06}$
	elevators	1.00	1.10 ± 0.15	1.05 ± 0.13	$\mathbf{0.85 \pm 0.15}$
	fried delve	1.00	1.60 ± 0.22	1.40 ± 0.15	$\mathbf{0.90 \pm 0.11}$
	puma32H	**1.00**	2.24 ± 1.18	1.45 ± 0.22	1.77 ± 2.42
	winequality	1.00	1.31 ± 0.12	1.24 ± 0.11	$\mathbf{0.97 \pm 0.04}$
Classification	abalone	**1.00**	1.29 ± 0.19	1.22 ± 0.16	1.05 ± 0.15
	ailerons	1.00	1.03 ± 0.27	1.01 ± 0.20	$\mathbf{0.86 \pm 0.13}$
	bank32nh	**1.00**	1.25 ± 0.13	1.20 ± 0.13	$\mathbf{1.00 \pm 0.09}$
	cal housing	1.00	1.43 ± 0.23	1.36 ± 0.19	$\mathbf{0.87 \pm 0.14}$
	cpu act	1.00	1.09 ± 0.16	1.06 ± 0.16	$\mathbf{0.99 \pm 0.15}$
	delta ailerons	1.00	1.38 ± 0.40	1.25 ± 0.31	$\mathbf{0.84 \pm 0.12}$
	elevators	1.00	1.07 ± 0.15	1.04 ± 0.14	$\mathbf{0.89 \pm 0.13}$
	fried delve	1.00	1.34 ± 0.22	1.22 ± 0.18	$\mathbf{0.85 \pm 0.09}$
	puma32H	**1.00**	1.73 ± 0.59	1.22 ± 0.18	1.10 ± 0.42
	winequality	**1.00**	1.20 ± 0.13	1.13 ± 0.10	1.07 ± 0.10

Table 5.2: *Average Test Errors (± std. deviation) - here we compared the predictive performance of decision trees in the test set of 75 different simulations for each dataset. We have four basic scenarios: (i) whole set of features and no weighting method; (ii) whole set of features and use of "true" weights; (iii) whole set of features and estimated weights; (iv) selected features and estimated weights. The numbers reported are the mean squared error and classification error averages and their std. deviations. All the results were normalized w.r.t. the first scenario.*

Secondly, we choose the minimum number of samples per leaf in each regression/classification tree from values in $[5, 15, 25, 40, 50]$, in order to minimize the mean squared error or classification error within a 2-fold cross-validation procedure. Finally, we maximize the log-likelihood in a holdout dataset to choose the number of GMM components, varying the possible number of components within the list $[1, 3, 5, 7, 9, 11, 13, 15]$.

5.7 Conclusion

In this chapter, we have made two main contributions. The first is that we explicitly and formally connected three key concepts in the context of covariate shift adaptation: (i) effective sample size, (ii) dimensionality of data, and (iii) generalization of a supervised model. Since, to the best of our knowledge, there is no unified and rigorous view on how the

three key concepts connect to each other, we consider this to be the first contribution of the chapter. The second contribution of the chapter is that we show dimensionality reduction or feature selection, even considering data dependent mappings, corrects small effective sample sizes by making the source and target distributions less divergent. This suggests that it is a good practice to perform dimensionality reduction or feature selection before covariate adaptation. We also present numerical experiments using real and artificial data to complement our theoretical results.

Regarding possible future research paths and improvements, we point to Sections 5.3 and 5.5. Concerning Section 5.3, perhaps the three most relevant points to be considered for future research relate to the following assumptions: the first one is assuming the importance function is known up to a constant, the second is assuming the sample ESS is close to its population version, and the third is assuming independent samples. While the first hardly applies in practice, the second may hold in many situations, and the third could be relaxed to include dependent samples, thus solving one of the problems discussed in Section 5.5. Considering Section 5.5, we think there is one main point to be explored in future work, which is extending the theorems to include more general transformations, i.e., non-linear or training data dependent transformations. Said that, future work and improvements of this chapter could focus on relaxing assumptions or exploring cases in which they are valid.

5.8 Code and data

All the datasets used are open datasets and are downloaded while running the code.

The code and material used can be found on https://github.com/felipemaiapolo/ess_dimensionality_covariate_shift. Also, we have made our feature selection implementation available as a Python package called *InfoSelect* https://github.com/felipemaiapolo/infoselect.

Chapter 6

Decomposing Dataset Shift into Covariate and Concept Shifts

6.1 Introduction

In this chapter, we propose a way to characterize Dataset Shifts in supervised learning tasks. Our proposal is to decompose the Kullback-Leibler (KL) divergence (Polyanskiy and Wu, 2019) between the joint distributions of features and labels, which represents the total dataset shift, into a part that depends only on the divergence of the marginal distributions of the features (covariate shift) and another that only depends on the average divergence of the conditional distributions of the labels (average concept shift[1]). In addition, we want to estimate these quantities from data.

The decomposition of the total dataset shift into covariate shift and average concept shift allows, among other things, the machine learning practitioner to better understand the data he/she is working with. For example, using dataset shift decomposition, it is possible to better understand how a certain population evolves over time and answer questions such as "what percentage of the total shift is due to the covariate shift?", "is it possible to detect relevant concept shift comparing population A with population B?", or "during an economic and health crisis, such as COVID-19's, how is the financial profile of Brazilians affected? How can this change in profile be decomposed into covariate shift and concept shift?".

[1] In the domain adaptation literature, it is more common to use the expression "concept drift", however, we chose to use the word "shift" in order to make the language more uniform.

This chapter is organized as follows. First, in Section 6.2 we briefly review the concepts of dataset and covariate shifts, seen in more depth in Chapter 3, and introduce the concept of concept shift/drift. Secondly, in Section 6.3 we present how an idea known in the Information Theory literature can be used to characterize the dataset shift, through the decomposition of KL divergences. In this same section, we present a toy experiment in order to consolidate an intuition behind the concepts presented. Finally, in Section 6.5 we present an application of the ideas presented in a real problem involving credit data for the Brazilian population.

6.2 Dataset Shift, Covariate Shift, and Concept Shift

Dataset shift is the situation in which the joint distribution of features and labels from which we can sample (source/train distribution) is different from the one we are interested in (target/test distribution). In this chapter, we name the source distribution as $Q_{\mathbf{x},\mathrm{y}}$ and the target distribution as $P_{\mathbf{x},\mathrm{y}}$. If there is a dataset shift, then $Q_{\mathbf{x},\mathrm{y}} \neq P_{\mathbf{x},\mathrm{y}}$.

Covariate shift is a type of dataset shift in which the training/source joint distribution $Q_{\mathbf{x},\mathrm{y}}$ differs from the test/target distribution $P_{\mathbf{x},\mathrm{y}}$ only by their features' marginals. That is, features and labels are sampled according to the same conditional distribution $Q_{\mathrm{y}|\mathbf{x}} = P_{\mathrm{y}|\mathbf{x}}$ but different marginals $Q_{\mathbf{x}} \neq P_{\mathbf{x}}$ (Sugiyama and Kawanabe, 2012). On the other hand, concept shift is another type of dataset shift often characterized by the situation in which training/source joint distribution $Q_{\mathbf{x},\mathrm{y}}$ differs from the test/target distribution $P_{\mathbf{x},\mathrm{y}}$ only by their labels' conditional distributions (Kull and Flach, 2014; Moreno-Torres *et al.*, 2012). That is, features are sampled from the same marginal distribution $Q_{\mathbf{x}} = P_{\mathbf{x}}$ but labels are sampled from different conditional distributions $Q_{\mathrm{y}|\mathbf{x}} \neq P_{\mathrm{y}|\mathbf{x}}$.

6.3 Decomposing Dataset Shift

In this section, we will explore a possible way to decompose dataset shift into covariate shift and average concept shift. We will use the Kullback-Leibler divergence to express each of these quantities concretely and quantitatively. In this way, depending on the context, we can refer to the quantities $\mathrm{D}_{\mathrm{KL}}(P_{\mathbf{x},\mathrm{y}}||Q_{\mathbf{x},\mathrm{y}})$, $\mathbb{E}_{\mathbf{x}\sim P_{\mathbf{x}}}\left[\mathrm{D}_{\mathrm{KL}}(P_{\mathrm{y}|\mathbf{x}}||Q_{\mathrm{y}|\mathbf{x}})\right]$, and $\mathrm{D}_{\mathrm{KL}}(P_{\mathbf{x}}||Q_{\mathbf{x}})$ as

(total) dataset shift, expected/average concept shift, and covariate shift, respectively. The greater those quantities, the greater the shifts.

Below we present a theoretical result of the Information Theory (Polyanskiy and Wu, 2019) central to our approach. For this theorem, we present a proof for the case where labels and features are discrete or absolutely continuous with respect to the Lebesgue measure. Polyanskiy and Wu (2019) discussions in the section "How to avoid measurability problems?" how can this idea be expanded to more general cases.

Theorem 6.1 (Extracted from Polyanskiy and Wu (2019))**.** *Being $P_{\mathbf{x},\mathrm{y}}$ the target joint distribution of features and labels and $Q_{\mathbf{x},\mathrm{y}}$ the source joint distribution of features and labels, we have that*

$$D_{KL}(P_{\mathbf{x},\mathrm{y}}||Q_{\mathbf{x},\mathrm{y}}) = \mathbb{E}_{\mathbf{x}\sim P_{\mathbf{x}}}\left[D_{KL}(P_{\mathrm{y}|\mathbf{x}}||Q_{\mathrm{y}|\mathbf{x}})\right] + D_{KL}(P_{\mathbf{x}}||Q_{\mathbf{x}}) \tag{6.1}$$

Proof. If $p_{\mathrm{y}|\mathbf{x}}, q_{\mathrm{y}|\mathbf{x}}, p_{\mathbf{x}}$, and $q_{\mathbf{x}}$ are p.d.f.s or p.m.f.s, then

$$\begin{aligned}
\mathrm{D}_{\mathrm{KL}}(P_{\mathbf{x},\mathrm{y}}||Q_{\mathbf{x},\mathrm{y}}) &= \mathbb{E}_{(\mathbf{x},\mathrm{y})\sim P_{\mathbf{x},\mathrm{y}}}\left[\log \frac{p_{\mathrm{y}|\mathbf{x}}(\mathrm{y}|\mathbf{x})p_{\mathrm{x}}(\mathbf{x})}{q_{\mathrm{y}|\mathbf{x}}(\mathrm{y}|\mathbf{x})q_{\mathrm{x}}(\mathbf{x})}\right] && (6.2)\\
&= \mathbb{E}_{(\mathbf{x},\mathrm{y})\sim P_{\mathbf{x},\mathrm{y}}}\left[\log \frac{p_{\mathrm{y}|\mathbf{x}}(\mathrm{y}|\mathbf{x})}{q_{\mathrm{y}|\mathbf{x}}(\mathrm{y}|\mathbf{x})}\right] + \mathbb{E}_{\mathbf{x}\sim P_{\mathbf{x}}}\left[\log \frac{p_{\mathrm{x}}(\mathbf{x})}{q_{\mathrm{x}}(\mathbf{x})}\right] && (6.3)\\
&= \mathbb{E}_{\mathbf{x}\sim P_{\mathbf{x}}}\left[\mathrm{D}_{\mathrm{KL}}(P_{\mathrm{y}|\mathbf{x}}||Q_{\mathrm{y}|\mathbf{x}})\right] + \mathrm{D}_{\mathrm{KL}}(P_{\mathbf{x}}||Q_{\mathbf{x}}) && (6.4)
\end{aligned}$$

□

This result shows that the concept shift is only relevant in regions of the feature space where we can sample points with positive probability considering the distribution $P_{\mathbf{x}}$; that is, if the region R of the input space $\mathcal{X}$ has null measure, i.e., $P_{\mathbf{x}}(R) = 0$, then the average $\mathrm{D}_{\mathrm{KL}}(P_{\mathrm{y}|\mathbf{x}}||Q_{\mathrm{y}|\mathbf{x}})$ in R is negligible and do not contribute to $\mathrm{D}_{\mathrm{KL}}(P_{\mathbf{x},\mathrm{y}}||Q_{\mathbf{x},\mathrm{y}})$.

6.4 Estimating the shifts

Our main objective in this section is to describe the procedure for estimating the following quantities of interest $\mathrm{D}_{\mathrm{KL}}(P_{\mathbf{x},\mathrm{y}}||Q_{\mathbf{x},\mathrm{y}})$, $\mathbb{E}_{\mathbf{x}\sim P_{\mathbf{x}}}\left[\mathrm{D}_{\mathrm{KL}}(P_{\mathrm{y}|\mathbf{x}}||Q_{\mathrm{y}|\mathbf{x}})\right]$, and $\mathrm{D}_{\mathrm{KL}}(P_{\mathbf{x}}||Q_{\mathbf{x}})$. The quantities

of interest are difficult to estimate as we do not know the distributions that generated the data and, besides that $\mathbf{x}$ and y can be multidimensional and composed of continuous, discrete, or mixed random variables. An approach that potentially handles these challenges well is outlined below. First, we use the (Sugiyama *et al.*, 2012b) probabilistic classification method to estimate density ratios $\frac{p_{\mathbf{x},\mathrm{y}}}{q_{\mathbf{x},\mathrm{y}}}$ and $\frac{p_{\mathbf{x}}}{q_{\mathbf{x}}}$, described in Section 4.2. Second, as suggested by Sønderby *et al.* (2016) and Tiao (2018) in the generative models literature, we obtain estimates of $\mathrm{D}_{\mathrm{KL}}(P_{\mathbf{x},\mathrm{y}}||Q_{\mathbf{x},\mathrm{y}})$ and $\mathrm{D}_{\mathrm{KL}}(P_{\mathbf{x}}||Q_{\mathbf{x}})$ using empirical averages. Finally, we get an estimate for $\mathbb{E}_{\mathbf{x}\sim P_{\mathbf{x}}}\left[\mathrm{D}_{\mathrm{KL}}(P_{\mathrm{y}|\mathbf{x}}||Q_{\mathrm{y}|\mathbf{x}})\right]$ subtracting the two quantities already estimated. Below, we explain in more detail the ideas presented in this paragraph.

Suppose our goal is to estimate $\mathrm{D}_{\mathrm{KL}}(P_{\mathbf{z}}||Q_{\mathbf{z}})$, given that $\mathbf{z} = \mathbf{x}$ or $\mathbf{z} = (\mathbf{x}, \mathrm{y})$ in our case. We will use the probabilistic classification method (Sugiyama *et al.*, 2012b) for this purpose, so we describe it below. Consider a binary random variable t that indicates if a random data point $\mathbf{z}$ is sampled from the target distribution $P_{\mathbf{z}}$ and not from the source distribution $Q_{\mathbf{z}}$. Thus, we can write $p_{\mathbf{z}}(\boldsymbol{z}) = f_{\mathbf{z}|\mathrm{t}}(\boldsymbol{z}|\mathrm{t}=1)$, $q_{\mathbf{z}}(\boldsymbol{z}) = f_{\mathbf{z}|\mathrm{t}}(\boldsymbol{z}|\mathrm{t}=0)$, if $f_{\mathbf{z}}(\boldsymbol{z}) = \mathbb{P}(\mathrm{t}=0)q_{\mathbf{z}}(\boldsymbol{z}) + \mathbb{P}(\mathrm{t}=1)p_{\mathbf{z}}(\boldsymbol{z})$. Using Bayes' Theorem, it is possible to write w as follows:

$$w(\boldsymbol{z}) = \frac{p_{\mathbf{z}}(\boldsymbol{z})}{q_{\mathbf{z}}(\boldsymbol{z})} \tag{6.5}$$

$$= \frac{f_{\mathbf{z}|\mathrm{t}}(\boldsymbol{z}|\mathrm{t}=1)}{f_{\mathbf{z}|\mathrm{t}}(\boldsymbol{z}|\mathrm{t}=0)} \tag{6.6}$$

$$= \frac{\mathbb{P}(\mathrm{t}=0)}{\mathbb{P}(\mathrm{t}=1)} \frac{\mathbb{P}(\mathrm{t}=1|\mathbf{z}=\boldsymbol{z})}{[1-\mathbb{P}(\mathrm{t}=1|\mathbf{z}=\boldsymbol{z})]} \tag{6.7}$$

Suppose we have the data points $\{\boldsymbol{z}'_i\}_{i=1}^m$ sampled from $P_{\mathbf{z}}$, and $\{\boldsymbol{z}_i\}_{i=1}^n$ sampled from $Q_{\mathbf{z}}$. We can append the datasets $\{(\boldsymbol{z}'_i, 1)\}_{i=1}^m$ and $\{(\boldsymbol{z}_i, 0)\}_{i=1}^n$, and then train a probabilistic classifier, e.g., XGBoost (Chen and Guestrin, 2016), to discriminate samples according to labels 1 and 0. The probabilistic classifier returns the function $\widehat{\mathbb{P}}(\mathrm{t}=1|\mathbf{z}=\boldsymbol{z})$ that approximates $\mathbb{P}(\mathrm{t}=1|\mathbf{z}=\boldsymbol{z})$. On the other hand, the quantity $\mathbb{P}(\mathrm{t}=0)/\mathbb{P}(\mathrm{t}=1)$ is estimated by n/m. Our estimator for w is given by

$$\hat{w}(\boldsymbol{z}) = \frac{n}{m} \frac{\widehat{\mathbb{P}}(\mathrm{t}=1|\mathbf{z}=\boldsymbol{z})}{[1-\widehat{\mathbb{P}}(\mathrm{t}=1|\mathbf{z}=\boldsymbol{z})]} \tag{6.8}$$

Finally, if we have extra samples $\{\tilde{\boldsymbol{z}}'_i\}_{i=1}^{\tilde{m}}$ sampled independently from $P_{\mathbf{z}}$, our estimator for $\mathrm{D}_{\mathrm{KL}}(P_{\mathbf{z}}||Q_{\mathbf{z}})$ is given by

$$\widehat{\mathrm{D}}_{\mathrm{KL}}(P_{\mathbf{z}}||Q_{\mathbf{z}}) = \frac{1}{\tilde{m}} \sum_{i=1}^{\tilde{m}} \log \hat{w}(\tilde{\boldsymbol{z}}'_i) \tag{6.9}$$

Following the steps described in this section, we can directly obtain $\widehat{\mathrm{D}}_{\mathrm{KL}}(P_{\mathbf{x},\mathrm{y}}||Q_{\mathbf{x},\mathrm{y}})$ and $\widehat{\mathrm{D}}_{\mathrm{KL}}(P_{\mathbf{x}}||Q_{\mathbf{x}})$. Consequently, if $\gamma := \mathbb{E}_{\mathbf{x}\sim P_{\mathbf{x}}}\left[\mathrm{D}_{\mathrm{KL}}(P_{\mathrm{y}|\mathbf{x}}||Q_{\mathrm{y}|\mathbf{x}})\right]$, we use the estimator $\widehat{\gamma} = \widehat{\mathrm{D}}_{\mathrm{KL}}(P_{\mathbf{x},\mathrm{y}}||Q_{\mathbf{x},\mathrm{y}}) - \widehat{\mathrm{D}}_{\mathrm{KL}}(P_{\mathbf{x}}||Q_{\mathbf{x}})$ to estimate γ.

It is worth mentioning that, given $\{\tilde{\boldsymbol{z}}'_i\}_{i=1}^{\tilde{m}}$ forms an i.i.d. sample from $P_{\mathbf{z}}$, $\widehat{\mathrm{D}}_{\mathrm{KL}}(P_{\mathbf{z}}||Q_{\mathbf{z}})$'s standard errors and asymptotic confidence intervals, from Central Limit Theorem (Roussas, 1997), are straightforward.

6.4.1 A Toy Experiment

In this experiment, (i) we use training and test sets with 100K data points each and (ii) we employ the approach presented in Section 6.4 in conjunction with a XGBoost classifier (Chen and Guestrin, 2016) for estimating the quantities $\mathrm{D}_{\mathrm{KL}}(P_{\mathbf{x},\mathrm{y}}||Q_{\mathbf{x},\mathrm{y}})$, $\mathbb{E}_{\mathbf{x}\sim P_{\mathbf{x}}}\left[\mathrm{D}_{\mathrm{KL}}(P_{\mathrm{y}|\mathbf{x}}||Q_{\mathrm{y}|\mathbf{x}})\right]$, and $\mathrm{D}_{\mathrm{KL}}(P_{\mathbf{x}}||Q_{\mathbf{x}})$.

Consider the source probability density functions:

$$q_{\mathrm{x}}(x) = \mathcal{N}(x|0,1) \qquad q_{\mathrm{y}|\mathrm{x}}(y|x) = \mathcal{N}(y|x,1) \tag{6.10}$$

And then consider the target probability density functions:

$$p_{\mathrm{x}}(x;\lambda) = \mathcal{N}(x|\lambda,1) \qquad p_{\mathrm{y}|\mathrm{x}}(y|x;\theta) = \mathcal{N}(y|\theta+x,1) \tag{6.11}$$

We can now decompose the joint distributions KL divergence (total dataset shift) as

$$\mathrm{D}_{\mathrm{KL}}(P_{\mathbf{x},\mathrm{y}}||Q_{\mathbf{x},\mathrm{y}}) = \mathbb{E}_{\mathbf{x}\sim P_{\mathbf{x}}}\left[\mathrm{D}_{\mathrm{KL}}(P_{\mathrm{y}|\mathbf{x}}||Q_{\mathrm{y}|\mathbf{x}})\right] + \mathrm{D}_{\mathrm{KL}}(P_{\mathbf{x}}||Q_{\mathbf{x}}) \tag{6.12}$$

$$= \frac{\theta^2}{2} + \frac{\lambda^2}{2} \tag{6.13}$$

Given that we are dealing with gaussian distributions, $\mathrm{D}_{\mathrm{KL}}(P_{\mathrm{y}|\mathbf{x}}||Q_{\mathrm{y}|\mathbf{x}})$ and $\mathrm{D}_{\mathrm{KL}}(P_{\mathbf{x}}||Q_{\mathbf{x}})$ are promptly obtained.

In order to present a first idea about the problem, we will adopt $\lambda = \theta = 1$ and plot below, in Figure 6.1, the empirical joint distributions obtained from the data:

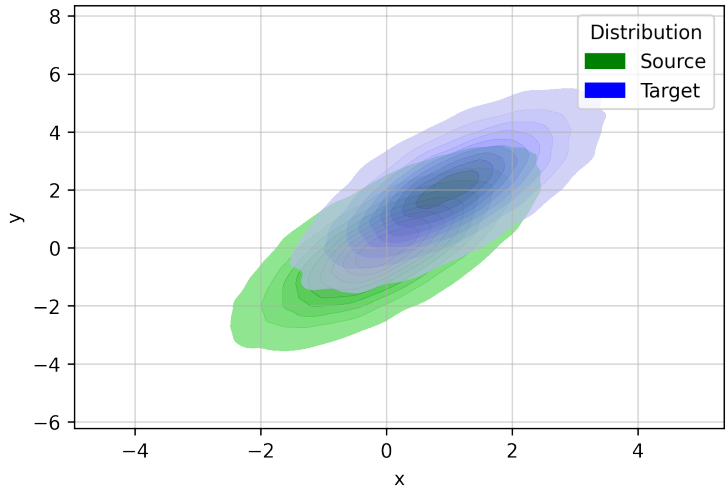

Figure 6.1: *Source and target distributions when adopting $\lambda = \theta = 1$. In this specific case, we have that $\mathbb{E}_{\mathbf{x}\sim P_{\mathbf{x}}}\left[D_{KL}(P_{\mathrm{y}|\mathbf{x}}||Q_{\mathrm{y}|\mathbf{x}})\right] = D_{KL}(P_{\mathbf{x}}||Q_{\mathbf{x}}) = 1/2$.*

In this specific case, we have that $\mathbb{E}_{\mathbf{x}\sim P_{\mathbf{x}}}\left[\mathrm{D}_{\mathrm{KL}}(P_{\mathrm{y}|\mathbf{x}}||Q_{\mathrm{y}|\mathbf{x}})\right] = \mathrm{D}_{\mathrm{KL}}(P_{\mathbf{x}}||Q_{\mathbf{x}}) = 1/2$, and consequently $\mathrm{D}_{\mathrm{KL}}(P_{\mathbf{x},\mathrm{y}}||Q_{\mathbf{x},\mathrm{y}}) = 1$. Even though this example gives an initial idea of the problem we want to tackle, it is very restrictive as it assumes fixed values for λ and θ. To get a more general idea about the problem, we use the equation 6.13 and plot in Figure 6.2 the values that our quantities of interest assume for values of λ and θ in a 15×15 grid of $[0.3]^2$.

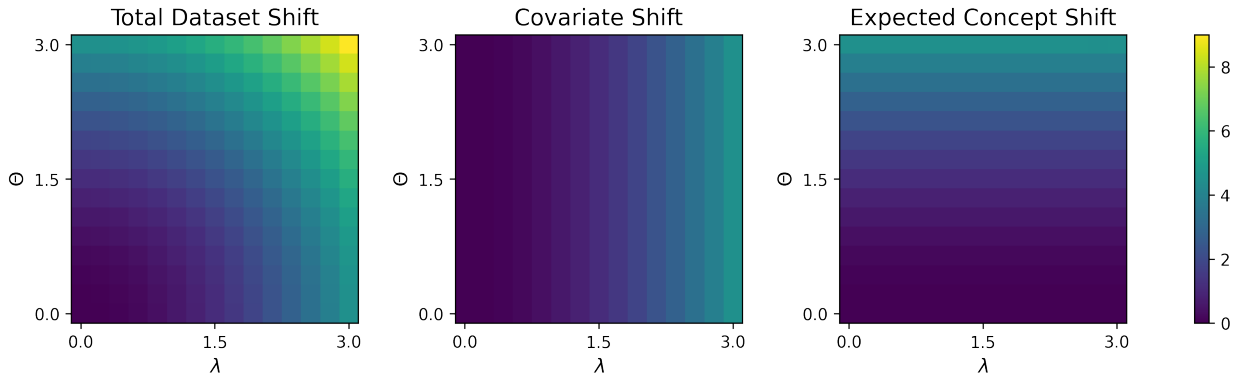

Figure 6.2: *Theoretical values of $D_{KL}(P_{\mathbf{x},\mathrm{y}}||Q_{\mathbf{x},\mathrm{y}})$, $D_{KL}(P_{\mathbf{x}}||Q_{\mathbf{x}})$, and $\mathbb{E}_{\mathbf{x}\sim P_{\mathbf{x}}}\left[D_{KL}(P_{\mathrm{y}|\mathbf{x}}||Q_{\mathrm{y}|\mathbf{x}})\right]$ varying λ and θ in a 15×15 grid of $[0, 3]^2$.*

In addition to the result presented in Figure 6.2, we present in Figure 6.3 the empirical

results obtained using the solution proposed in Section 6.4.

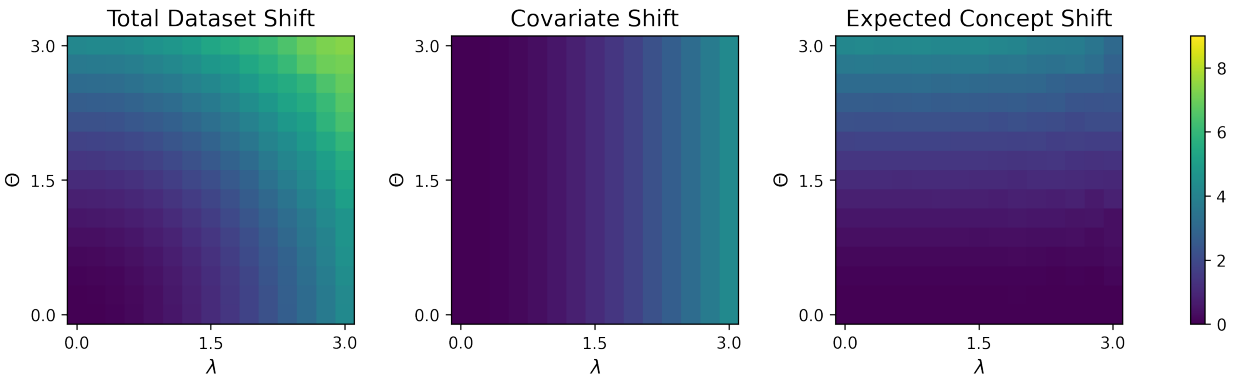

Figure 6.3: *Empirical values (XGBoost) of* $D_{KL}(P_{\mathbf{x},\mathrm{y}}||Q_{\mathbf{x},\mathrm{y}})$, $D_{KL}(P_{\mathbf{x}}||Q_{\mathbf{x}})$, *and* $\mathbb{E}_{\mathbf{x}\sim P_{\mathbf{x}}}\left[D_{KL}(P_{\mathrm{y}|\mathbf{x}}||Q_{\mathrm{y}|\mathbf{x}})\right]$ *varying* λ *and* θ *in a* 15×15 *grid of* $[0,3]^2$.

We can see from Figure 6.3 that the empirical solution is, in general, very accurate. However, it is possible to notice that this solution can start to fail when the divergences reach their greatest values. Figure 6.4 compares the theoretical and empirical values for the quantities of interest and makes this observation more apparent.

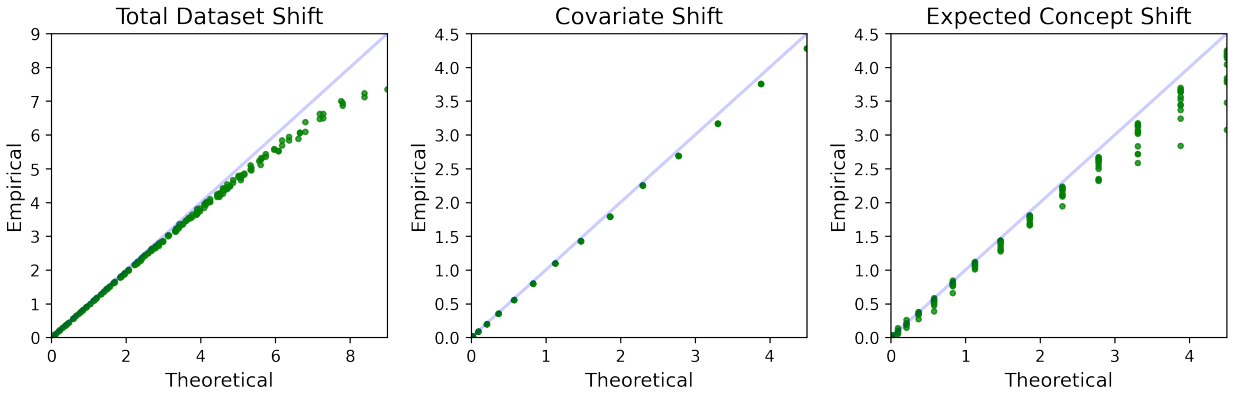

Figure 6.4: *Comparing theoretical and empirical values of* $D_{KL}(P_{\mathbf{x},\mathrm{y}}||Q_{\mathbf{x},\mathrm{y}})$, $D_{KL}(P_{\mathbf{x}}||Q_{\mathbf{x}})$, *and* $\mathbb{E}_{\mathbf{x}\sim P_{\mathbf{x}}}\left[D_{KL}(P_{\mathrm{y}|\mathbf{x}}||Q_{\mathrm{y}|\mathbf{x}})\right]$ *varying* λ *and* θ *in a* 15×15 *grid of* $[0,3]^2$.

In the next section, we apply the ideas presented here to real data used in credit analysis.

6.5 Application with real Credit Data

6.5.1 Objective and its practical importance

The objective of this application is to better understand the dynamics of some characteristics of the Brazilian population frequently used in credit analysis. It is expected that the characteristics

and behaviors of the population will change over time, but it is necessary to quantify these changes to better understand, for example, how and why credit models can have their performance degraded over time. To accomplish this goal, we will use the concepts presented in the previous sections in order to decompose the total dataset shift into covariate shift and expected concept shift.

6.5.2 Data

The dataset used in the experiments of this section was provided by the Serasa Experian Datalab, the Latin American Experian Datalab[2] based in São Paulo, Brazil. The dataset contains longitudinal data for one million Brazilians over a ten-month period, which runs from August/2019 to May/2020. Each of the ten time slices contains 1788 features and a label variable, often used for training credit models. In this work, we randomly partition the sample into ten mutually exclusive parts, thus keeping 100,000 Brazilians in each of the months.

Features and label variable

First, in all time slices we had access to the same set of 1789 variables, being 1788 features and 1 label variable. Second, the variables that we will use in these experiments are often used to train credit models, that is, to predict which people are more likely not to pay their debts in the near future. That said, our label variable is an indicator variable that assumes 1 if the person has been at least 30 days late in paying any debt in a future 3 month period. In turn, the features are composed of variables that may have statistical dependence with the variable of interest, and these may, for example, relate to the amount of loans and financing made in the last year or the amount of credit card bills not paid on time in the last few months.

Many of the features provided are redundant and many are simply poor predictors. Also, the dataset contains too many missing values. Because of this, we filter rows and columns in each of the datasets, in order to make our analysis computationally less expensive. In the next subsection, we explain in more detail how the row filtering and feature selection

[2]See https://www.experian.com/big-data/datalabs

procedure was done.

Feature selection and row filtering

In order to make our tasks computationally less expensive, we will work with a subset of the data in our analyses. Column selection and row filtering are done in the following order:

1. Firstly, we trained a XGBoost algorithm for classifying people in August/2019, with no hyperparameter tuning phase but using early stopping. Then, we kept in all the data slices only the $d = 75$ features with the greatest feature importance, which is "the average gain across all splits the feature is used in"[3], according to the August/2019 dataset;

2. Secondly, we kept in each of the time slices only those people who had at most 75% of missing values in their d features. In order to simplify the experiments, we randomly undersampled each of the data slices in order to maintain all of them with the same sample size, which is roughly 54K;

Following the two steps above, we arrive at our final datasets, each of which is made up of approximately 54K rows and exactly 75 columns. One last detail is that missing values were encoded as -1, while all the features can naturally admit only non-negative values.

6.5.3 Methodology

In this section, we describe how we achieve the objective described in Section 6.5.1 with the knowledge built in Sections 6.3 and 6.4.

First, it is necessary to introduce some new definitions and notations:

- $F^{(t)}_{\mathbf{x}_k,\mathrm{y}}$: joint distribution, in month t, of the label variable y and the most important k features according to the metric commented on in Section 6.5.2 represented by $\mathbf{x}_k$;

- $F^{(t)}_{\mathbf{x}_k}$: marginal distribution, in month t, of the most important k features according to the metric commented on in Section 6.5.2 represented by $\mathbf{x}_k$;

[3]See https://xgboost.readthedocs.io/en/latest/python/python_api.html - Accessed in 24/06/2021

- $F^{(t)}_{\mathrm{y}|\mathbf{x}_k}$: conditional distribution, in month t, of the label variable y given the most important k features according to the metric commented on in Section 6.5.2 represented by $\mathbf{x}_k$;

In this work, we have that $t \in \{0, ..., 9\}$ and we choose to work with $k \in \{1, 2, 3, 5, 10, 15, 25, 50, 75\}$.

Our objective, described in Section 6.5.1, can be summarized in the estimation of the following quantities: (i) $\mathrm{D_{KL}}(F^{(t)}_{\mathbf{x}_k,\mathrm{y}}||F^{(0)}_{\mathbf{x}_k,\mathrm{y}})$ (total dataset shift), (ii) $\mathrm{D_{KL}}(F^{(t)}_{\mathbf{x}_k}||F^{(0)}_{\mathbf{x}_k})$ (covariate shift), and (iii) $\mathbb{E}_{\mathbf{x}_k \sim F^{(t)}_{\mathbf{x}_k}}\left[\mathrm{D_{KL}}(F^{(t)}_{\mathrm{y}|\mathbf{x}_k}||F^{(0)}_{\mathrm{y}|\mathbf{x}_k})\right]$ (expected concept shift) for all possible combinations of t and k. After the estimations, if we fix k, it is possible to draw curves that represent the temporal evolution of the quantities of interest from August/2019 to May/2020. By observing the temporal evolution of the quantities of interest, we can better understand (i) whether there is a temporal dataset shift, (ii) what the intensity of the shift is, and (iii) how much of the total dataset shift is due to the covariate shift and how much is due to the concept shift.

To estimate the quantities of interest, we use the approach suggested in Sections 6.3 and 6.4. As the estimation process has a direct dependence on the training and test sets[4], in addition to also depending on the random seed of our probabilistic classifier, we performed the same experiment 75 times. Each time, we set a different seed for both the dataset splitting procedure and the probabilistic classifier training. As a probabilistic classifier, we use the XGBoost binary classifier (Chen and Guestrin, 2016) without hyperparameter tuning, but using early-stopping in order to minimize the log loss in a validation set[5]. In each of the 75 simulations, we train the XGBoost classifier in the training set and use the test set to estimate the divergences, as described in Section 6.4. At the end, we worked with the average of the 75 estimates in addition to using their standard deviation as a measure of uncertainty.

6.5.4 Results

In Figure 6.5, we can see the results of this application with real credit data. The solid and dashed lines were obtained by taking the experiments average and the error bars are

[4]Training and test sets are composed by 90% and 10% portions of the original dataset. We use the training set to train the probabilistic classifier and the test set to estimate the KL divergence.

[5]Randomly formed by 10% of the training set.

the size of one standard deviation from the 75 different experiments.

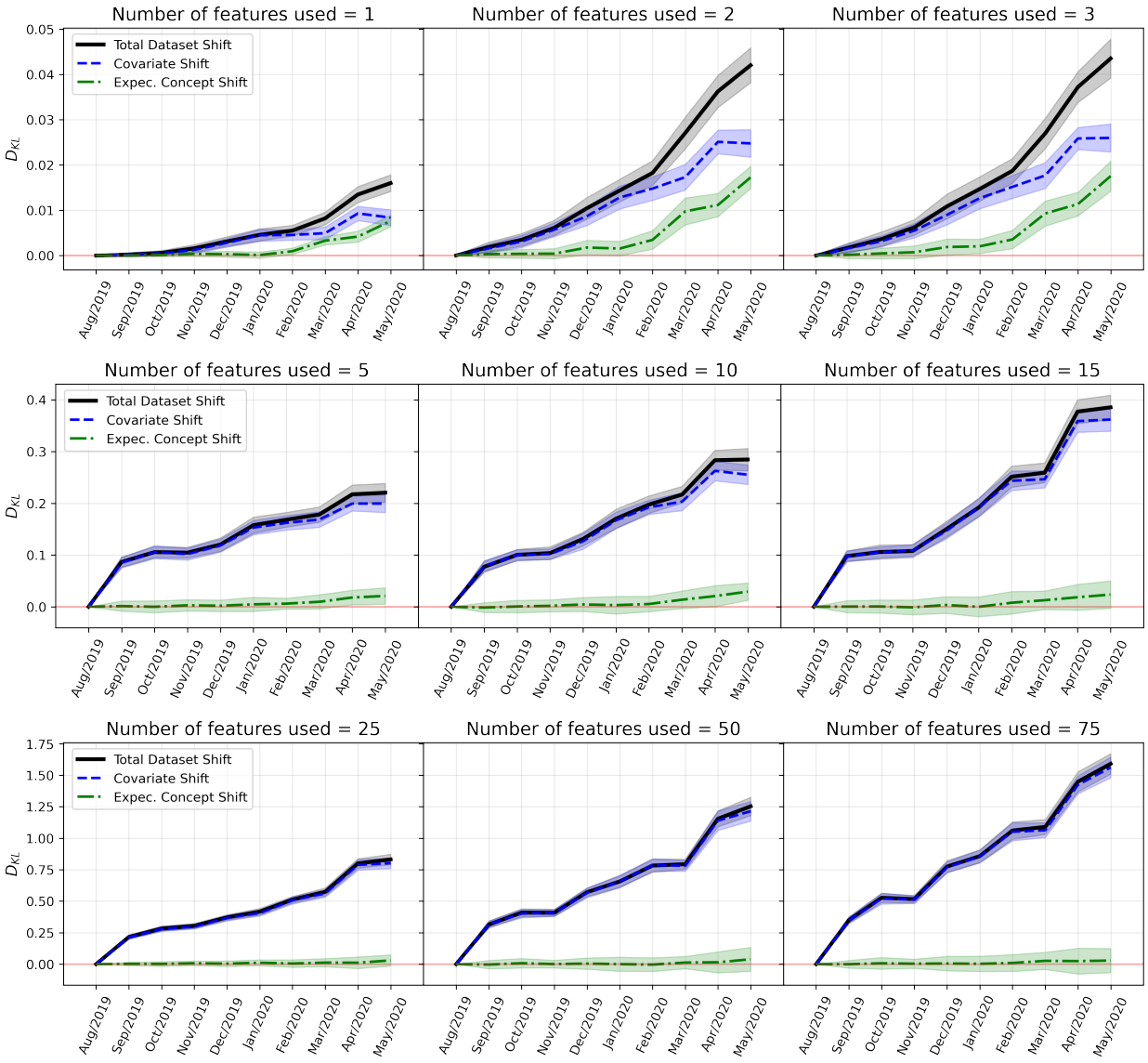

Figure 6.5: *Temporal total dataset, covariate, and expected concept shifts considering August/2019 as baseline. As intuition suggests, shifts increase with time and the number of variables. The sharp increase in expected concept shift might reflect people's behavior change due to COVID-19 pandemic.*

The first observation we can make with respect to Figure 6.5 is that the dataset shift tends to increase over time, regardless of the number of variables we are analyzing. A second point, which was already expected, is that the greater the number of variables considered, the greater the total dataset shift and covariate shift. This is due to the monotonicity of the KL divergence, analogous to what we saw for the Rényi divergence in Section 5.4. Thirdly, and which is related to the previous observation, is that as we consider more features, the total dataset shift becomes almost entirely explained by the covariate shift. This is because the conditional distribution represents a univariate variable. Lastly, and perhaps

most interestingly: it is clear that the expected concept shift, at least in the first six plots, becomes more relevant from February/2020 onwards. Since the label variable y tells whether people had payment delay problems in the next three months, we have information until mid-May/2020. One possible explanation for the sudden change in Brazilians' behavior, represented by the conditional distribution of labels, is the economic crisis triggered by the COVID-19 pandemic[6]. If this hypothesis could be verified in practice, it would enable us to conjecture important theories for understanding the behavior of Brazilian borrowers, as it seems that the conditional distribution of y given $\mathbf{x}_k$ is stable in time, relative to the distribution of $\mathbf{x}_k$, until an extraordinary event occurs. Unfortunately, with the analyzes we have so far, we could not assert causal relationships, however, this observation opens a gap for future research.

In the following, we try to better understand what happens to the conditional distribution of labels after January/2020.

Better understanding concept shift after January/2020

To better understand what happens with the conditional distribution of labels given the features after January/2020, we performed the following analysis. Firstly, we split the data of each month (54K rows) in a training (81%), validation (9%), and test sets (10%). Secondly, for each month, we trained a XGBoost classifier in order to predict problems with payment (label=1) given a set of features. We used the training set portion to train the classifiers and the validation part for early-stopping purposes. Thirdly, we built two time series called (i) factual and (ii) counterfactual. The factual one is given by the average predicted probability of payment delay problems in the next three months calculated in the test set, when using classifiers and features from the same month. For example, in order to calculate the average predicted probability for December/2019, we used the XGBoost classifier trained in that month to predict probabilities using test samples from that same month, and then we calculate the average predicted probability for that sample. On the other

[6]On March 22, 2020, the governor of the State of São Paulo (Brazil) decreed state quarantine (See https://www.saopaulo.sp.gov.br/wp-content/uploads/2020/03/decreto-quarentena.pdf - Accessed in 06/26/2021) closing many commercial establishments for a few months, which also ended up happening in other Brazilian regions a few days later.

hand, the counterfactual one is given by the average predicted probability of payment delay problems in the next three months calculated in the test set, when using the classifier trained in August/2019 and features from the month of interest. For example, in order to calculate the average predicted probability for December/2019, we used the XGBoost classifier trained in August/2019 to predict probabilities using test samples from December/2019, and then we calculate the average predicted probability for that sample. Given that we fix the predicted conditional distribution of labels in the counterfactual scenario, the different between curves in Figure 6.6 can be interpreted as an isolated effect of concept shift.

As done before, we performed the experiment 75 times, given that in each one, we change the random seed related to the classifiers training and dataset splitting. In Figure 6.6, we report the average across experiments while the error bars are the size of one standard deviation from the 75 different experiments.

As it can be seen in Figure 6.6, after January/2020, there is a sharp decrease in the average predicted probabilities of payment delay problems. We can infer that the sharp decrease when looking at the black line is mainly due to concept shift given that the gap between the factual and counterfactual time series also dramatically increases after January/2020. Another observation is that the blue line is more or less constant over time, and that means covariate shift by itself cannot cause big changes in the payment observed behavior.

Regarding the possible connection between rapid changes in the conditional distribution of the label variable and the economic crisis triggered by the COVID-19 pandemic, one explanation for the sharp decreasing behavior of the black line after the arrival of the pandemic COVID-19 in Brazil could be that many measures taken by banks and credit bureaus help consumers during the pandemic. Some of the measures include, but are not limited to, lower interest rates and longer payment intervals[7].

[7]See https://www.serasa.com.br/ensina/seu-credito/credito-pessoal/ - Accessed in 27/07/2021.

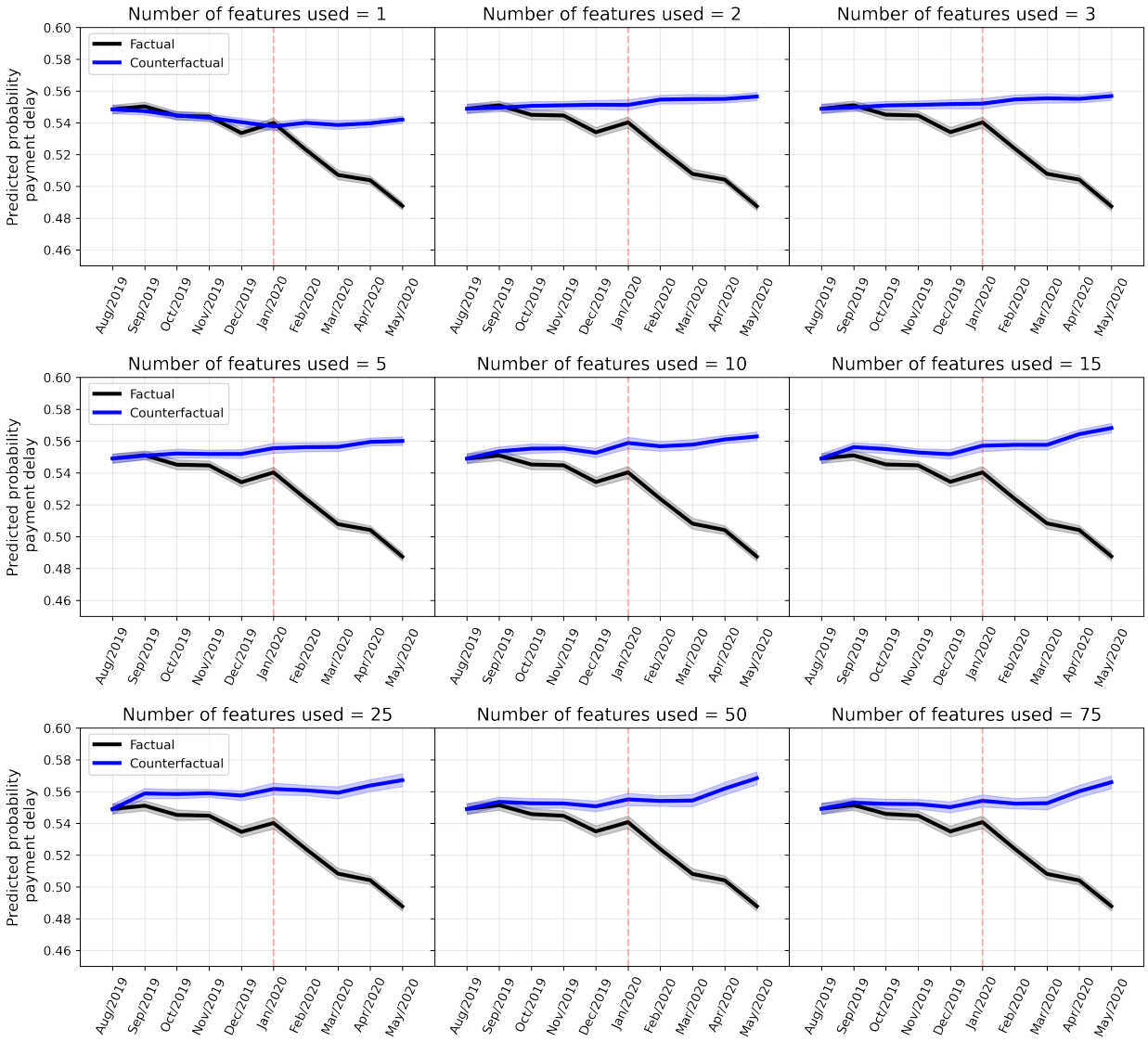

Figure 6.6: *The role of concept shift. The factual time series is given by the average predicted probability of payment delay problems in the next three months calculated in the test set, when using classifiers and features from the same month. On the other hand, the counterfactual one is given by the average predicted probability of payment delay problems in the next three months calculated in the test set, when using the classifier trained in August/2019 and features from the month of interest. Given that we fix the predicted conditional distribution of labels in the counterfactual scenario, the different between curves can be interpreted as an isolated effect of concept shift. The sharp increase in the gap of the two lines after January/2020 might reflect concept shift due to COVID-19 pandemic.*

6.6 Conclusion

In this chapter, we proposed a new way to characterize dataset shift in supervised learning tasks. Our approach permits the researcher to quantify and decompose the Kullback-Leibler (KL) divergence between the joint distributions of features and labels, which represents the total dataset shift, into a part that represents the covariate shift and another that represents

the concept shift. With that, we can quantify each term separately and better understand the nature shifting data. Future directions of research could be studying how to predict dataset shift and model degradation, or using the results presented by Rhodes *et al.* (2020) to obtain better divergences' estimates.

6.7 Code

The codes used to generate the results of this chapter can be found in https://github.com/felipemaiapolo/decomposing_dataset_shift. The datasets are private and could not be shared.

6.8 Acknowledgement

I would like to thank the Serasa Experian DataLab for sharing their datasets and computing infrastructure, so I could run the experiments presented in this chapter.

Chapter 7

Conclusion

During this work, we went from an extensive covariate shift adaptation review to our own original contributions to dataset shift literature. In Chapters 2 3 and 4, we reviewed the fundamental concepts of statistical learning theory, covariate shift problem and its solution, and density ratio/importance estimation. In Chapter 5, we proposed a new unifying theory that connects effective sample size (ESS), features dimensionality and generalization, which are concepts present in the modern literature of covariate shift adaptation. We show that (i) bigger ESSs lead to sharper generalization bounds, (ii) data dimensionality is directly linked to the ESS, and (iii) dimensionality reduction can make the ESS bigger. Finally, in Chapter 6, we proposed and applied a new approach to better understand data under dataset shift, decomposing it into covariate shift and expected concept drift/shift. With that, one can quantify each term separately and better understand the nature shifting data.

Possible future research directions regarding the work presented in Chapters 5 and 6 of this dissertation are the following. With respect to the Chapter 5, loosening some assumptions made in Sections 5.3 and 5.5 are the most direct ones, as pointed out in the end of that chapter. Concerning Chapter 6, work could be done in studying how to predict dataset shift and model degradation, or using the results presented by Rhodes *et al.* (2020) to obtain better divergences' estimates.

Appendix A

Some proofs

A.1 Proof of Theorem 2.8

Proof. First we prove that, when n gets large, there is only a little probability that exists a hypothesis in $\mathcal{H}$ which the empirical error is far away from the statistical risk. Mathematically, we first prove that for a given $\epsilon' > 0$, there is a constant $C > 0$ that satisfies:

$$F_{\mathcal{S}}\left\{\mathcal{S} : \exists h \in \mathcal{H} \text{ that } \left|\hat{R}_L(h,\mathcal{S}) - R_L(h)\right| \geq \epsilon'\right\} \leq C\exp(-2n\epsilon'^2) \tag{A.1}$$

Consider $\mathcal{H} = \{h_1, ..., h_{|\mathcal{H}|}\}$, and note that

$$F_{\mathcal{S}}\left\{\mathcal{S} : \exists h \in \mathcal{H} \text{ that } \left|\hat{R}_L(h,\mathcal{S}) - R_L(h)\right| \geq \epsilon'\right\} \tag{A.2}$$

$$= F_{\mathcal{S}}\bigcup_{i=1}^{|\mathcal{H}|}\left\{\mathcal{S} : \left|\hat{R}_L(h_i,\mathcal{S}) - R_L(h_i)\right| \geq \epsilon'\right\} \tag{A.3}$$

$$\leq \sum_{i=1}^{|\mathcal{H}|} F_{\mathcal{S}}\left[\mathcal{S} : \left|\hat{R}_L(h_i,\mathcal{S}) - R_L(h_i)\right| \geq \epsilon'\right] \tag{A.4}$$

$$\leq 2|\mathcal{H}|\exp(-2n\epsilon'^2) \tag{A.5}$$

Then, in our case $C = 2|\mathcal{H}|$. The inequality A.4 is obtained by the subadditivity property of the probability measure and the inequality A.5 is obtained by the result presented in the Example 2.5. Now, note that the event $\left\{\mathcal{S} : \exists h \in \mathcal{H} \text{ that } \left|\hat{R}_L(h,\mathcal{S}) - R_L(h)\right| \geq \epsilon'\right\}^c = \left\{\mathcal{S} : \forall h \in \mathcal{H} \text{ we have } \left|\hat{R}_L(h,\mathcal{S}) - R_L(h)\right| < \epsilon'\right\} = E$ has a probability of at least

$1 - 2|\mathcal{H}| \exp(-2n\epsilon'^2)$. Moreover, note that

$$E \subseteq \left\{\mathcal{S} : \left|\hat{R}_L(h_{\mathcal{S}}^{\mathrm{ERM}}, \mathcal{S}) - R_L(h_{\mathcal{S}}^{\mathrm{ERM}})\right| < \epsilon'\right\} \bigcap \left\{\mathcal{S} : \left|\hat{R}_L(h^*, \mathcal{S}) - R_L(h^*)\right| < \epsilon'\right\} \tag{A.6}$$

Where $h^* \in \mathrm{argmin}_{h \in \mathcal{H}} R_L(h)$. Then with probability of at least $1 - 2|\mathcal{H}| \exp(-2n\epsilon'^2)$, it is true that:

$$R_L(h_{\mathcal{S}}^{\mathrm{ERM}}) \leq \hat{R}_L(h_{\mathcal{S}}^{\mathrm{ERM}}, \mathcal{S}) + \epsilon' \leq \hat{R}_L(h^*, \mathcal{S}) + \epsilon' \leq R_L(h^*) + 2\epsilon' = \min_{h \in \mathcal{H}} R_L(h) + 2\epsilon' \tag{A.7}$$

Where the first and third inequalities are due to the fact stated above and the second one is due to the definition of $h_{\mathcal{S}}^{\mathrm{ERM}}$. Adopting $\epsilon = 2\epsilon'$,

$$F_{\mathcal{S}}\left[R_L(h_{\mathcal{S}}^{\mathrm{ERM}}) - \min_{h \in \mathcal{H}} R_L(h) < \epsilon\right] > 1 - 2|\mathcal{H}| \exp\left(-\frac{n\epsilon^2}{2}\right) \tag{A.8}$$

Rewriting,

$$F_{\mathcal{S}}\left[R_L(h_{\mathcal{S}}^{\mathrm{ERM}}) - \min_{h \in \mathcal{H}} R_L(h) \geq \epsilon\right] \leq 2|\mathcal{H}| \exp\left(-\frac{n\epsilon^2}{2}\right) \tag{A.9}$$

Setting $\delta \geq 2|\mathcal{H}| \exp\left(-\frac{n\epsilon^2}{2}\right)$ and solving for n, we can see that

$$n \geq \frac{2}{\epsilon^2} \log\left(\frac{2|\mathcal{H}|}{\delta}\right) \tag{A.10}$$

We then conclude the proof affirming that, in this case, the minimal sample complexity is bounded from above by $\frac{2}{\epsilon^2} \log\left(\frac{2|\mathcal{H}|}{\delta}\right)$.

□

Appendix B

Other Methods for Importance Estimation

We refer to the source/training distribution of features as $Q_{\mathbf{x}}$ and the target/test distribution of features as $P_{\mathbf{x}}$. Also, for the sake of simplicity, we assume both distributions are absolutely continuous with probability density functions $q_{\mathbf{x}}$ and $p_{\mathbf{x}}$, such that $\text{support}(p_{\mathbf{x}}) \subseteq \text{support}(q_{\mathbf{x}})$. Furthermore, we always consider $\boldsymbol{x}$ and $\{\boldsymbol{x}_i\}_{i=1}^{n}$ to be data points independently sampled from $Q_{\mathbf{x}}$, and $\boldsymbol{x}$ and $\{\boldsymbol{x}'_i\}_{i=1}^{n'}$ to be data points independently sampled from $P_{\mathbf{x}}$, i.e., realizations of $\mathbf{x} \sim Q_{\mathbf{x}}$ and $\mathbf{x}' \sim P_{\mathbf{x}}$, respectively.

B.1 Kernel Mean Matching (KMM)

The weighting strategy that we present in this section is known as "Kernel Mean Matching" (Huang *et al.*, 2007). It proposes estimating the weighting function w applied to the training by matching the means of the two groups (training/test or source/target) in the feature space associated with a universal kernel (Steinwart, 2001). First, we need to understand better the Reproducing Kernel Hilbert Space (RKHS)[1].

Definition B.1. (***Positive Semidefinite (PSD) Kernel***): a kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is positive semidefinite if for $n \in \mathbb{N}$ and a sequence of data points $\{\boldsymbol{x}_i\}_{i=1}^{n} \in \mathcal{X}$, the matrix $\boldsymbol{K} = \{K_{i,j}\}_{i,j=1}^{n}$ defined as $K_{i,j} = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is positive semidefinite.

[1] See (Wainwright, 2019) for more details.

Consider the following map $\boldsymbol{x} \mapsto K(.,\boldsymbol{x}) \in \mathbb{H}$, where $\mathbb{H}$ is a Hilbert Space[2] of real functions with domain in $\mathcal{X}$. If K is a positive semidefinite kernel, there is only one set $\mathbb{H}$ where K satisfies the reproducing property.

Theorem B.2. ***(Reproducing Kernel Hilbert Space (RKHS))****: Given a positive semidefinite kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, there exists a unique Hilbert Space $\mathbb{H}$ of functions with domain $\mathcal{X}$ in which the kernel K satisfies the reproducing property:*

$$\langle f, K(.,\boldsymbol{x})\rangle_{\mathbb{H}} = f(\boldsymbol{x}),\ \forall f \in \mathbb{H} \tag{B.1}$$

$\mathbb{H}$ is the Reproducing Kernel Hilbert Space associated with kernel K.

Proof. See Theorem 12.11 from Wainwright (2019). □

In this way, we can calculate inner products $\langle K(.,\boldsymbol{x}_i), K(.,\boldsymbol{x}_j)\rangle_{\mathbb{H}}$ in $\mathbb{H}$, which is a higher dimensional space relative to $\mathcal{X}$, simply by evaluating $K(\boldsymbol{x}_i,\boldsymbol{x}_j)$. To move forward, we will say that $f : \mathcal{X} \to \mathbb{R}$ is induced by the K kernel if there is $g \in \mathbb{H}$ such that $f(\boldsymbol{x}) = \langle g, K(.,\boldsymbol{x})\rangle_{\mathbb{H}},\ \forall \boldsymbol{x} \in \mathcal{X}$. From Theorem B.2, it is straightforward that if K is a positive semidefinite kernel and $\mathbb{H}$ its RKHS, then every $f \in \mathbb{H}$ function is induced by K. Now that we understand what a RKHS is, let us discuss the universal kernel (Micchelli *et al.*, 2006; Steinwart, 2001)[3].

Definition B.3. (*Universal Kernel*): $(\mathcal{X}, d)$ is a metric space. The continuous kernel K defined in the compact set $(\mathcal{Z}, d)$, $\mathcal{Z} \subseteq \mathcal{X}$, is called universal kernel if the space of all functions induced by K is dense in $C(\mathcal{Z})$, that is, in the space of all continuous functions with domain $\mathcal{Z}$. In other words, for all $\epsilon > 0$ and $f \in C(\mathcal{Z})$, there exists g induced by K that satisfies

$$\sup_{\boldsymbol{z}\in\mathcal{Z}} |f(\boldsymbol{z}) - g(\boldsymbol{z})| \leq \epsilon \tag{B.2}$$

[2] A Hilbert space is a complete inner product space, i.e., a vector space with an inner product in which every Cauchy sequence converges to an element of the set.

[3] Definition 4 in the original article.

From now on, we will discuss the method of our interest, presented by Huang *et al.* (2006, 2007). Let $\phi : \mathcal{X} \to \mathbb{H}$ be a feature map, with $\mathbb{H}$ being a Hilbert Space representing the "Feature Space". In addition, we define $\mu(F) = \mathbb{E}_{\mathbf{x}\sim F}[\phi(\mathbf{x})]$ as the expected value operator. The Theorem B.4 is fundamental to understand why the method works.

Theorem B.4. ***(Bijection of the Operator** μ**)**: *the operator μ is a bijection if $\mathbb{H}$ is a RKHS induced by an universal kernel $K(\boldsymbol{x}, \boldsymbol{x}') = \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{x}') \rangle_{\mathbb{H}}$.*

Proof. See Theorem 1 from Huang *et al.* (2006). □

In practice, it is common to define our feature map being given by the Gaussian kernel, that is, $\phi(\boldsymbol{x}) = K_\sigma(., \boldsymbol{x})$. Given that, we will adopt this choice from now on. This choice is based on the fact that kernels invariant to translation, i.e., $K(\boldsymbol{x}, \boldsymbol{y}) = k(\boldsymbol{x} - \boldsymbol{y}),\ \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$ for a continuous function k, are universal kernels (Micchelli *et al.*, 2006). The Theorem B.4 implies that, if we adopt $\phi(\boldsymbol{x}) = K_\sigma(., \boldsymbol{x})$, we have $\mu(F) = \mu(F') \Rightarrow F = F'$, F and F' being distributions over $\mathcal{X} \subseteq \mathbb{R}^d$. Given that, the problem of choosing a weighting function $w : \mathcal{X} \to \mathbb{R}_+$ is given by the following minimization problem:

$$\min_{w} \left\| \mathop{\mathbb{E}}_{\mathbf{x}'\sim P_{\mathbf{x}}} \Big[K_\sigma\big(., \mathbf{x}'\big) \Big] - \mathop{\mathbb{E}}_{\mathbf{x}\sim Q_{\mathbf{x}}} \Big[w\big(\mathbf{x}\big) K_\sigma\big(., \mathbf{x}\big) \Big] \right\|_{\mathbb{H}}^2 \tag{B.3}$$

$$\text{s.t.} \mathop{\mathbb{E}}_{\mathbf{x}\sim Q_{\mathbf{x}}} \Big[w\big(\mathbf{x}\big) \Big] = 1 \text{ and } w \geq 0 \tag{B.4}$$

Where $||.||_{\mathbb{H}}$ denotes the norm induced by the inner product. If we have the samples

$\{\boldsymbol{x}'_i\}_{i=1}^{n'}$ and $\{\boldsymbol{x}_i\}_{i=1}^{n}$, we will rewrite the objective function in its empirical version as follows:

$$\left\| \frac{1}{n'} \sum_{i=1}^{n'} K_\sigma\big(., \boldsymbol{x}'_i\big) - \frac{1}{n} \sum_{j=1}^{n} w(\boldsymbol{x}_k) K_\sigma\big(., \boldsymbol{x}_k\big) \right\|_{\mathbb{H}}^2 = \tag{B.5}$$

$$= \left\langle \frac{1}{n'} \sum_{i=1}^{n'} K_\sigma\big(., \boldsymbol{x}'_i\big) - \frac{1}{n} \sum_{j=1}^{n} w(\boldsymbol{x}_k) K_\sigma\big(., \boldsymbol{x}_k\big),\ \frac{1}{n'} \sum_{i=1}^{n'} K_\sigma\big(., \boldsymbol{x}'_i\big) - \frac{1}{n} \sum_{j=1}^{n} w(\boldsymbol{x}_k) K_\sigma\big(., \boldsymbol{x}_k\big) \right\rangle_{\mathbb{H}} \tag{B.6}$$

$$= \frac{1}{n'^2} \sum_{i,j=1}^{n'} K_\sigma\big(\boldsymbol{x}'_i, \boldsymbol{x}'_j\big) - \frac{2}{nn'} \sum_{i=1}^{n'} \sum_{j=1}^{n} w(\boldsymbol{x}_k) K_\sigma\big(\boldsymbol{x}'_i, \boldsymbol{x}_k\big) + \frac{1}{n^2} \sum_{i,j=1}^{n} w(\boldsymbol{x}_i) w(\boldsymbol{x}_k) K_\sigma\big(\boldsymbol{x}_i, \boldsymbol{x}_k\big) \tag{B.7}$$

$$= \frac{1}{n^2} \sum_{i,j=1}^{n} w(\boldsymbol{x}_i) w(\boldsymbol{x}_k) K_\sigma\big(\boldsymbol{x}_i, \boldsymbol{x}_k\big) - \frac{2}{nn'} \sum_{j=1}^{n} w(\boldsymbol{x}_k) \sum_{i=1}^{n'} K_\sigma\big(\boldsymbol{x}'_i, \boldsymbol{x}_k\big) + C \tag{B.8}$$

$$= \frac{1}{n^2} \boldsymbol{w}^\top \boldsymbol{K} \boldsymbol{w} - \frac{2}{nn'} \boldsymbol{w}^\top \mathbf{k} + C \tag{B.9}$$

Where the vector $\boldsymbol{w}$ has entries $\boldsymbol{w}_i = w(\boldsymbol{x}_i)$, the matrix $\boldsymbol{K}$ has entries $K_{i,j} = K_\sigma\big(\boldsymbol{x}_i, \boldsymbol{x}_j\big)$, the vector $\boldsymbol{k}$ has entries $k_j = \sum_{i=1}^{n'} K_\sigma\big(\boldsymbol{x}'_i, \boldsymbol{x}_j\big)$, and C is a constant which does not depend on w. Thus, we present an empirical version of our minimization problem (Huang *et al.*, 2006, 2007). We end up with a problem of quadratic programming problem:

$$\min_{\boldsymbol{w}} \quad \frac{1}{n^2} \boldsymbol{w}^\top \boldsymbol{K} \boldsymbol{w} - \frac{2}{nn'} \boldsymbol{w}^\top \boldsymbol{k} \tag{B.10}$$

$$\text{s.t} \ \left| \boldsymbol{w}^\top \mathbf{1}/n - 1 \right| \leq \epsilon \text{ and } \boldsymbol{w} \in [0, B]^n \tag{B.11}$$

The problem can be solved using standard techniques. Here $\mathbf{1}$ is a vector of ones and the hyperparameters $\epsilon > 0$ and $B > 0$ work as regularization terms (Sugiyama and Kawanabe, 2012). To conclude this section, we present another interesting result (Huang *et al.*, 2006, 2007):

Theorem B.5. ***(Relationship between** B **and sample size)**: Admit* $\text{Var}[w(\mathbf{x})] > 0$ *if* $\mathbf{x} \sim Q_{\mathbf{x}}$, $\{\mathbf{x}_i\}_{i=1}^{n} \overset{iid}{\sim} Q_{\mathbf{x}}$, $\{\mathbf{x}'_i\}_{i=1}^{n'} \overset{iid}{\sim} P_{\mathbf{x}}$, $K_\sigma(\boldsymbol{x}, \boldsymbol{x}) \leq R^2$ *e* $w(\boldsymbol{x}) \in [0, B]$ *for* $\forall \boldsymbol{x} \in \mathcal{X}$. *Then,*

with probability of at least $1-\delta$*, we have:*

$$\left\| \frac{1}{n'} \sum_{i=1}^{n'} K_\sigma(.,\mathbf{x}'_i) - \frac{1}{n} \sum_{j=1}^{n} w(\mathbf{x}_j) K_\sigma(.,\mathbf{x}_j) \right\|_{\mathbb{H}}^2 \leq R^2 \left[1 + \sqrt{-2\log(\delta/2)}\right]^2 \left[B^2/n + 1/n'\right] \tag{B.12}$$

Proof. See Lemma 4 from Huang *et al.* (2006). □

Where w is the "correct" weighting function in the population sense. Theorem B.5 tells us that given B, if we have a big enough sample, KMM is theoretically feasible.

The hyperparameters $\epsilon > 0$ and $B > 0$ are not optimizable by cross validation as we calculate the weights only for the examples in the training set. Despite that, a good choice of ϵ is a function of order $O(B/\sqrt{n})$ (Huang *et al.*, 2006, 2007). The Gaussian kernel bandwidth σ can be fixed as the average distance between sample points (Song *et al.*, 2007). In the literature, one can find another version of the KMM algorithm in which the hyperparameters can be tuned by cross validation (Kanamori *et al.*, 2009b).

B.2 Density Matching methods

The following methods are extensions or modifications of KLIEP, presented in Section 4.4.

B.2.1 Gaussian Mixture Kullback-Leibler Importance Estimation Procedure (GM-KLIEP)

This method is a particular case of the KLIEP method in which non-spherical Gaussian kernels are adopted (Sugiyama *et al.*, 2012a; Yamada and Sugiyama, 2009). Despite being a particular case of KLIEP, GM-KLIEP is still more general than the case in which we use

spherical kernels. In this case, we model the weighting function as

$$w_{\boldsymbol{\beta}}(\boldsymbol{x}) = \sum_{t=1}^{T} \beta_t K(\boldsymbol{x}; \boldsymbol{\mu_t}, \sigma_t) \tag{B.13}$$

$$= \sum_{t=1}^{T} \beta_t \exp\left[-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu_t})^\top \boldsymbol{\Sigma}_t^{-1}(\boldsymbol{x} - \boldsymbol{\mu_t})\right] \tag{B.14}$$

Where the parameters $\{(\beta_t, \boldsymbol{\mu_t}, \boldsymbol{\Sigma}_t)\}_{t=1}^{T}$ are empirically learnt. Given the fact that we could write $p_{\mathbf{x}}(\boldsymbol{x}; \boldsymbol{\beta}) = w_{\boldsymbol{\beta}}(\boldsymbol{x}) q_{\mathbf{x}}(\boldsymbol{x})$, we admit $\beta_t \geq 0$ for $t = 1, ..., T$. We can transpose the optimization problem presented in Section 4.4 to this case and obtain:

$$\max_{\{(\beta_t, \boldsymbol{\mu_t}, \boldsymbol{\Sigma}_t)\}_{t=1}^{T}} \quad \sum_{i=1}^{n'} \log\left[\sum_{t=1}^{T} \beta_t K_\sigma\left(\boldsymbol{x}_i'; \boldsymbol{\mu_t}, \boldsymbol{\Sigma}_t\right)\right] \tag{B.15}$$

$$\text{s.t.} \ \sum_{i=1}^{n} \sum_{t=1}^{T} \beta_t K_\sigma\left[\boldsymbol{x}_i; \boldsymbol{\mu_t}, \boldsymbol{\Sigma}_t\right] = n \text{ and } \boldsymbol{\beta} \geq 0 \tag{B.16}$$

The maximization is made using the EM Algorithm (Bishop, 2006; Yamada and Sugiyama, 2009) or even a fixed point method (Sugiyama *et al.*, 2012a). As the optimization problem is non-convex, these procedures may yield local optima (Sugiyama *et al.*, 2012a).

The model selection is straightforward since this is a hyperparameter tuning phase. As our objective is to maximize an objective function, Yamada and Sugiyama (2009) suggests researchers use an ordinary K-fold cross validation procedure to tune regularization parameters or the number of kernels.

B.2.2 Principal Mixture Kullback-Leibler Importance Estimation Procedure (PM-KLIEP)

Even though GM-KLIEP works well in the experiments presented by Yamada and Sugiyama (2009), the authors state that the method may be unstable in some regions of space, where the data can induce ill-conditioned covariance matrices. Thus, it would be hard to invert such matrices with precision (Yamada *et al.*, 2010). To correct this point, the authors make an adaptation that reduces the dimensions using the Probabilistic Principal

Component Analysis (Tipping and Bishop, 1999; Yamada *et al.*, 2010). In this case, we model the weighting function as follows:

$$w_{\boldsymbol{\beta}}(\boldsymbol{x}) = \sum_{t=1}^{T} \beta_t K(\boldsymbol{x}; \sigma_t^2, \boldsymbol{\mu_t}, \boldsymbol{W}_t) \tag{B.17}$$

Given

$$K(\boldsymbol{x}; \sigma_t^2, \boldsymbol{\mu_t}, \boldsymbol{W}_t) = (2\pi\sigma_t^2)^{-\frac{d}{2}} \det(\boldsymbol{C}_t)^{-\frac{1}{2}} \exp\left[-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu_t})^\top \boldsymbol{C}_t^{-1}(\boldsymbol{x} - \boldsymbol{\mu_t})\right] \tag{B.18}$$

$$\boldsymbol{C}_t = \boldsymbol{W}_t \boldsymbol{W}_t^\top + \sigma_t^2 \boldsymbol{I}_d \tag{B.19}$$

Where d is the original number of variables and $\boldsymbol{W}_t$ is a projection matrix with dimensions $d \times m$, with $m \leq d$. If $d = m$, we have GM-KLIEP. The parameters $\{(\beta_t, \sigma_t, \boldsymbol{\mu_t}, \boldsymbol{W}_t)\}_{t=1}^T$ are empirically learnt. Given the fact we could write $p_{\mathbf{x}}(\boldsymbol{x}; \boldsymbol{\beta}) = w_{\boldsymbol{\beta}}(\boldsymbol{x}) q_{\mathbf{x}}(\boldsymbol{x})$, we admit $\beta_t \geq 0$ para $t = 1, ..., T$. Translating the optimization problems presented in Sections 4.4 and B.2.1 to this case, we get:

$$\max_{\{(\beta_t, \sigma_t, \boldsymbol{\mu_t}, \boldsymbol{W}_t)\}_{t=1}^T} \sum_{i=1}^{n'} \log\left[\sum_{t=1}^{T} \beta_t K(\boldsymbol{x}_i'; \boldsymbol{\sigma_t^2}, \boldsymbol{\mu_t}, \boldsymbol{W}_t)\right] \tag{B.20}$$

$$\text{s.t.} \sum_{i=1}^{n} \sum_{t=1}^{T} \beta_t K\left(\boldsymbol{x}_i; \sigma_t^2, \boldsymbol{\mu_t}, \boldsymbol{W}_t\right) = n \text{ and } \boldsymbol{\beta} \geq 0 \tag{B.21}$$

The maximization is made using the EM Algorithm (Bishop, 2006; Yamada *et al.*, 2010). Unfortunately, there is no unique solution and then we could get stuck in a local optima (Sugiyama *et al.*, 2012a).

As our objective is to maximize an objective function, Yamada *et al.* (2010) suggests researchers use an ordinary K-fold cross validation procedure to tune regularization parameters or the number of kernels.

B.2.3 Log-Linear Kullback-Leibler Importance Estimation Procedure (LL-KLIEP)

This approach is an adaptation of KLIEP to large scale applications (Tsuboi *et al.*, 2009). The importance function w is modelled as follows:

$$w_{\boldsymbol{\beta}}(\boldsymbol{x}) = \frac{\exp\left[\sum_{t=1}^{T} \beta_t \varphi_t(\boldsymbol{x})\right]}{\mathop{\mathbb{E}}_{\mathbf{x}\sim Q_{\mathbf{x}}}\left\{\exp\left[\sum_{t=1}^{T} \beta_t \varphi_t(\mathbf{x})\right]\right\}} \tag{B.22}$$

$$= \frac{\exp\left[\boldsymbol{\beta}^\top \boldsymbol{\varphi}(\boldsymbol{x})\right]}{\mathop{\mathbb{E}}_{\mathbf{x}\sim Q_{\mathbf{x}}}\left\{\exp\left[\boldsymbol{\beta}^\top \boldsymbol{\varphi}(\mathbf{x})\right]\right\}} \tag{B.23}$$

Where the basis functions $\boldsymbol{\varphi}(\boldsymbol{x}) = (\varphi_1(\boldsymbol{x}), ...\varphi_t(\boldsymbol{x}))$ are hyperparameters and the vector $\boldsymbol{\beta} = (\beta_1, ..., \beta_T)$ is learnt empirically. Note that $w_{\boldsymbol{\beta}} \geq 0$ and there is no need to constrain $\boldsymbol{\varphi}$ or $\boldsymbol{\beta}$ as in KLIEP (Sugiyama *et al.*, 2008). Furthermore, because LL-KLIEP is modelled like in Equation B.22, there is no need to reinforce the condition 4.34. In practice, we do not have access to the expected value in the denominator of $w_{\boldsymbol{\beta}}$, then we use the empirical mean. If we have the samples $\{\boldsymbol{x}_i\}_{i=1}^{n}$ and $\{\boldsymbol{x}'_i\}_{i=1}^{n'}$, we can rewrite the function above as follows:

$$w_{\boldsymbol{\beta}}(\boldsymbol{x}) = \frac{\exp\left[\boldsymbol{\beta}^\top \boldsymbol{\varphi}(\boldsymbol{x})\right]}{\frac{1}{n}\sum_{j=1}^{n} \exp\left[\boldsymbol{\beta}^\top \boldsymbol{\varphi}(\boldsymbol{x}_j)\right]} \tag{B.24}$$

Adapting the optimization problem of KLIEP (4.35) to this case, we get:

$$\max_{\boldsymbol{\beta}\in\mathbb{R}^T} \sum_{i=1}^{n'} \log\left\{\frac{\exp\left[\boldsymbol{\beta}^\top \boldsymbol{\varphi}(\boldsymbol{x}'_i)\right]}{\frac{1}{n}\sum_{j=1}^{n} \exp\left[\boldsymbol{\beta}^\top \boldsymbol{\varphi}(\boldsymbol{x}_j)\right]}\right\} \tag{B.25}$$

We could also rewrite it as follows:

$$\max_{\boldsymbol{\beta}\in\mathbb{R}^T} \sum_{i=1}^{n'} \boldsymbol{\beta}^\top \boldsymbol{\varphi}(\boldsymbol{x}'_i) - \log \sum_{j=1}^{n} \exp\left[\boldsymbol{\beta}^\top \boldsymbol{\varphi}(\boldsymbol{x}_j)\right] \tag{B.26}$$

The objective function can be maximized using the gradient ascent algorithm (Tsuboi *et al.*, 2009). According to the authors, we can still add a regularization term that constrains the norm of $\boldsymbol{\beta}$. We have two main computational advantages of LL-KLIEP over KLIEP: (i) the

optimization problem is unconstrained and (ii) once we calculate the gradient for the first time, we no longer need to use the samples $\{\boldsymbol{x}'_i\}_{i=1}^{n'}$ for later iterations. To check this fact, we write the gradient:

$$\nabla_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{n'} \boldsymbol{\beta}^\top \boldsymbol{\varphi}(\boldsymbol{x}'_i) - \log \sum_{j=1}^{n} \exp\left[\boldsymbol{\beta}^\top \boldsymbol{\varphi}(\boldsymbol{x}_j)\right] \right\} \tag{B.27}$$

$$= \nabla_{\boldsymbol{\beta}} \sum_{i=1}^{n'} \boldsymbol{\beta}^\top \boldsymbol{\varphi}(\boldsymbol{x}'_i) - \nabla_{\boldsymbol{\beta}} \log \sum_{j=1}^{n} \exp\left[\boldsymbol{\beta}^\top \boldsymbol{\varphi}(\boldsymbol{x}_j)\right] \tag{B.28}$$

$$= \sum_{i=1}^{n'} \boldsymbol{\varphi}(\boldsymbol{x}'_i)^\top - \nabla_{\boldsymbol{\beta}} \log \sum_{j=1}^{n} \exp\left[\boldsymbol{\beta}^\top \boldsymbol{\varphi}(\boldsymbol{x}_j)\right] \tag{B.29}$$

$$= \boldsymbol{\phi} - \nabla_{\boldsymbol{\beta}} \log \sum_{j=1}^{n} \exp\left[\boldsymbol{\beta}^\top \boldsymbol{\varphi}(\boldsymbol{x}_j)\right] \tag{B.30}$$

Note that once we calculate $\boldsymbol{\phi}$, there is no need to use $\{\boldsymbol{x}'_i\}_{i=1}^{n'}$ anymore.

As our objective is to maximize an objective function, Tsuboi *et al.* (2009) suggests researchers use an ordinary K-fold cross validation procedure to tune parameters like the number of kernels, kernels' hyperparameters or regularization strength. Tsuboi *et al.* (2009) also present alternatives for the objective function B.25 in order to obtain a more efficient model selection procedure.

B.2.4 Trimmed Density Ratio Estimator

This approach was introduced by Liu *et al.* (2017), with the objective of having a robust method for estimating the weighting functions w, using density matching methods in the presence of some "pathological" data points, that is, those data points $\boldsymbol{x} \in \{\boldsymbol{x}'_i\}_{i=1}^{n'}$ such that $w(\boldsymbol{x})$ assume large values. The presence of few outliers is capable to undermine the performance of the estimators of w, as shown in some experiments (Liu *et al.*, 2017). Then, the authors propose a method that, when estimating w, we automatically disconsider potentially problematic data points, obtaining a more robust estimator. Having the samples $\{\boldsymbol{x}_i\}_{i=1}^{n}$ and $\{\boldsymbol{x}'_i\}_{i=1}^{n'}$, Liu *et al.* (2017) models w in the same way it was made for the LL-KLIEP

(Sugiyama *et al.*, 2008) method:

$$w_{\boldsymbol{\beta}}(\boldsymbol{x}) = \frac{\exp\left[\boldsymbol{\beta}^\top \boldsymbol{\varphi}(\boldsymbol{x})\right]}{\frac{1}{n}\sum_{j=1}^{n} \exp\left[\boldsymbol{\beta}^\top \boldsymbol{\varphi}(\boldsymbol{x}_j)\right]} \tag{B.31}$$

Where the basis functions $\boldsymbol{\varphi}(\boldsymbol{x}) = (\varphi_1(\boldsymbol{x}), ...\varphi_t(\boldsymbol{x}))$ are hyperparameters and the vector $\boldsymbol{\beta} = (\beta_1, ..., \beta_T)$ is learnt empirically. As shown in Section B.2.3, there is no need to constrain $\boldsymbol{\varphi}$ or $\boldsymbol{\beta}$. Adpating the objective function B.25, we get the following optimizations problem:

$$\max_{\boldsymbol{\beta}\in\mathbb{R}^T} \sum_{i=1}^{n'} \min\left\{0,\ \log\left[w_{\boldsymbol{\beta}}(\boldsymbol{x}'_i)\right] - t_0\right\} \quad \Leftrightarrow \quad \min_{\boldsymbol{\beta}\in\mathbb{R}^T} \sum_{i=1}^{n'} \max\left\{0,\ t_0 - \log\left[w_{\boldsymbol{\beta}}(\boldsymbol{x}'_i)\right]\right\} \tag{B.32}$$

Where t_0 is threshold for the log of the importance functions. That is, if a data point has a weight that exceeds the threshold, we discard/trim that data point in the estimation procedure and then we are able to get rid of problems caused by outliers. We will now handle the above problem as follows:

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^T} \sum_{i=1}^{n'} \max\left\{0,\ t_0 - \log\left[w_{\boldsymbol{\beta}}(\boldsymbol{x}'_i)\right]\right\} \quad \Leftrightarrow \quad \min_{\boldsymbol{\beta}\in\mathbb{R}^T,\ \boldsymbol{\xi}\in\mathbb{R}^{n'}} \sum_{i=1}^{n'} \xi_i \tag{B.33}$$

$$\text{s.t.} \quad \xi_i \geq t_0 - \log\left[w_{\boldsymbol{\beta}}(\boldsymbol{x}'_i)\right],\ \forall i \in [n'] \tag{B.34}$$

$$\xi_i \geq 0,\ \forall i \in [n'] \tag{B.35}$$

Liu *et al.* (2017) reformulates the problem in the following way:

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^T,\ \boldsymbol{\xi}\in\mathbb{R}^{n'},\ t\geq 0} \quad \frac{1}{n'}\sum_{i=1}^{n'} \xi_i - \nu t + \lambda\Omega(\boldsymbol{\beta}) \tag{B.36}$$

$$\text{s.t.} \quad \xi_i \geq t - \log\left[w_{\boldsymbol{\beta}}(\boldsymbol{x}'_i)\right],\ \forall i \in [n'] \tag{B.37}$$

$$\xi_i \geq 0,\ \forall i \in [n'] \tag{B.38}$$

Where $\Omega(\boldsymbol{\beta})$ is a regularization term and $\lambda \geq$ e $\nu \in (0, 1]$ are hyperparameters. Rewriting

the problem in this way is interesting because it allows the researcher to control the proportion of trimmed points. Intuitively, $1-\nu$ is the proportion of trimmed points, i.e., if $\nu = 1$, then we do not trim any point. To solve the optimization problem, Liu *et al.* (2017) suggests a version of the Gradient Ascent algorithm, the "Gradient Ascent and Trimming."

Liu *et al.* (2017) does not comment about model selection and how to choose λ, for example.

B.3 Least-Squares Importance Fitting methods

The following methods are extensions or modifications of LSIF, presented in Section 4.5.

B.3.1 Unconstrained Least-Squares Importance Fitting (uLSIF)

The method presented in this section is an approximated method for LSIF, presented in Section 4.5. Using this approach can be useful because it returns similar results to the original method but being much more computationally efficient (Kanamori *et al.*, 2009a). The main change in the previous method's configuration is that, in this case, we remove the non-negativity condition from $\boldsymbol{\beta}$ during the optimization. In this way, we model the weighting function as follows:

$$w_{\boldsymbol{\beta}}(\boldsymbol{x}) = \sum_{t=1}^{T} \beta_t \varphi_t(\boldsymbol{x}) \tag{B.39}$$

$$= \boldsymbol{\beta}^\top \boldsymbol{\varphi}(\boldsymbol{x}) \tag{B.40}$$

Then we solve the following problem imposing no constraints on $\boldsymbol{\beta}$:

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^T} \quad \left[\frac{1}{2} \boldsymbol{\beta}^\top \widehat{\boldsymbol{H}} \boldsymbol{\beta} - \boldsymbol{\beta}^\top \hat{\boldsymbol{h}} + \frac{\lambda}{2} ||\boldsymbol{\beta}||_2^2 \right] \tag{B.41}$$

$$\text{s.t. } \boldsymbol{\varphi} \geq 0 \tag{B.42}$$

It is possible to prove that the optimization problem is strictly convex, and therefore its solution returns the global minimum. It can be seen that in this case, we have adopted the

regularization of the type l_2, which precisely will make it possible to obtain an analytical solution to the problem. Assuming $\boldsymbol{\varphi} \geq 0$ by the nature of the basis functions (e.g., Gaussian kernels) and replacing $||\boldsymbol{\beta}||_2^2$ with $\boldsymbol{\beta}^\top \boldsymbol{\beta}$, we can obtain the solution if we equals the gradient of the objective function to the null vector and solve for $\boldsymbol{\beta}$:

$$\nabla_{\boldsymbol{\beta}} \left[\frac{1}{2} \boldsymbol{\beta}^\top \widehat{\boldsymbol{H}} \boldsymbol{\beta} - \boldsymbol{\beta}^\top \hat{\boldsymbol{h}} + \frac{\lambda}{2} \boldsymbol{\beta}^\top \boldsymbol{\beta} \right] = 0 \tag{B.43}$$

$$\Rightarrow \frac{1}{2} \nabla_{\boldsymbol{\beta}} \, \boldsymbol{\beta}^\top \widehat{\boldsymbol{H}} \boldsymbol{\beta} - \nabla_{\boldsymbol{\beta}} \, \boldsymbol{\beta}^\top \hat{\boldsymbol{h}} + \frac{\lambda}{2} \nabla_{\boldsymbol{\beta}} \boldsymbol{\beta}^\top \boldsymbol{\beta} \tag{B.44}$$

$$= \widehat{\boldsymbol{H}} \boldsymbol{\beta} - \hat{\boldsymbol{h}} + \lambda \boldsymbol{\beta} = 0 \tag{B.45}$$

$$\Rightarrow \tilde{\boldsymbol{\beta}} = \left(\widehat{\boldsymbol{H}} + \lambda \boldsymbol{I}_T \right)^{-1} \hat{\boldsymbol{h}} \tag{B.46}$$

Where $\tilde{\beta}$ is an intermediate approximation for $\boldsymbol{\beta}$. In order to obtain its final approximation, we truncate $\tilde{\beta}$ as follows:

$$\hat{\boldsymbol{\beta}} = \max(0, \tilde{\boldsymbol{\beta}}) \tag{B.47}$$

In practice, the uLSIF method gives solutions close to those when using the LSIF method, especially when λ and σ (Gaussian kernel bandwidth) are big (Kanamori *et al.*, 2009a). The reasoning behind the basis functions is the same as presented in Section 4.4.

The advantage of uLSIF over LSIF resides on statistical risk estimation. For this evaluation regarding the use of uLSIF, the most used method is the Leave-One-Out-cross validation (LOOCV). That is because using the Sherman-Woodbury-Morrison Formula (Golub and Van Loan, 2012), it is possible to get the LOOCV score in closed form (Kanamori *et al.*, 2009a). That is, using the LOOCV approach is as cheap as the computation of one solution for the problem B.41. Kanamori *et al.* (2009a) also presents an algorithm that could be used to tune λ and σ (Gaussian kernel bandwidth) efficiently.

B.3.2 Relative Unconstrained Least-Squares Importance Fitting (RuLSIF)

The RuLSIF method is a generalization of uLSIF and was introduced by Yamada *et al.* (2013). If we fix $\alpha \in [0, 1]$, our objective would be to estimate:

$$\tilde{w}_\alpha(\boldsymbol{x}) = \frac{p_{\mathbf{x}}(\boldsymbol{x})}{\alpha p_{\mathbf{x}}(\boldsymbol{x}) + (1-\alpha) q_{\mathbf{x}}(\boldsymbol{x})} \tag{B.48}$$

Note that the denominator is given by a mixture of distributions, and if $\alpha = 0$ we return to uLSIF's case. An interesting property that this function has is that $\tilde{w}_\alpha \leq 1/\alpha$, while the importance function we have seen so far, w, is not bounded from above. As usual, we model $\tilde{w}_\alpha(\boldsymbol{x})$ as a linear combination of basis functions:

$$\tilde{w}_{\alpha,\boldsymbol{\beta}}(\boldsymbol{x}) = \sum_{t=1}^{T} \beta_t \varphi_t(\boldsymbol{x}) \tag{B.49}$$

$$= \boldsymbol{\beta}^\top \boldsymbol{\varphi}(\boldsymbol{x}) \tag{B.50}$$

Defining the distribution function $A_\alpha = \alpha P_{\mathbf{x}} + (1-\alpha) Q_{\mathbf{x}}$, we introduce and manipulates the objective function $J_\alpha(\boldsymbol{\beta})$, which we want to minimize w.r.t. $\boldsymbol{\beta}$:

$$J_\alpha(\boldsymbol{\beta}) = \frac{1}{2} \mathop{\mathbb{E}}_{\mathbf{x} \sim A_\alpha} \left\{ \left[\tilde{w}_{\alpha,\boldsymbol{\beta}}(\mathbf{x}) - \tilde{w}_\alpha(\mathbf{x}) \right]^2 \right\} \tag{B.51}$$

$$= \frac{\alpha}{2} \mathop{\mathbb{E}}_{\mathbf{x}' \sim P_{\mathbf{x}}} \left[\tilde{w}^2_{\alpha,\boldsymbol{\beta}}(\mathbf{x}') \right] + \frac{1-\alpha}{2} \mathop{\mathbb{E}}_{\mathbf{x} \sim Q_{\mathbf{x}}} \left[\tilde{w}^2_{\alpha,\boldsymbol{\beta}}(\mathbf{x}) \right] - \mathop{\mathbb{E}}_{\mathbf{x}' \sim P_{\mathbf{x}}} \left[\tilde{w}_{\alpha,\boldsymbol{\beta}}(\mathbf{x}') \right] + C \tag{B.52}$$

$$= \frac{\alpha}{2} \mathop{\mathbb{E}}_{\mathbf{x}' \sim P_{\mathbf{x}}} \left\{ \left[\boldsymbol{\beta}^\top \boldsymbol{\varphi}(\mathbf{x}') \right]^2 \right\} + \frac{1-\alpha}{2} \mathop{\mathbb{E}}_{\mathbf{x} \sim Q_{\mathbf{x}}} \left\{ \left[\boldsymbol{\beta}^\top \boldsymbol{\varphi}(\mathbf{x}) \right]^2 \right\} - \mathop{\mathbb{E}}_{\mathbf{x}' \sim P_{\mathbf{x}}} \left[\boldsymbol{\beta}^\top \boldsymbol{\varphi}(\mathbf{x}') \right] + C \tag{B.53}$$

$$= J'_\alpha(\boldsymbol{\beta}) + C \tag{B.54}$$

Where C is constant term which does not depend on $\tilde{w}_{\alpha,\boldsymbol{\beta}}$ and we can simply ignore it if our objective is to minimize $J_\alpha(\boldsymbol{\beta})$ w.r.t. $\boldsymbol{\beta}$. If there are samples $\{\boldsymbol{x}_i\}_{i=1}^{n}$ and $\{\boldsymbol{x}'_i\}_{i=1}^{n'}$ sampled from $Q_{\mathbf{x}}$ (source) and $P_{\mathbf{x}}$ (target), and going through the computation of the objective's

function empirical form, we arrive at

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^T} \quad \left[\frac{1}{2}\boldsymbol{\beta}^\top \widehat{\boldsymbol{H}}\beta - \boldsymbol{\beta}^\top \hat{\boldsymbol{h}} + \frac{\lambda}{2}||\boldsymbol{\beta}||_2^2\right] \tag{B.55}$$

$$\text{s.t. } \boldsymbol{\varphi} \geq 0 \tag{B.56}$$

Where $\widehat{\boldsymbol{H}}$ is a $T \times T$ matrix and $\hat{\boldsymbol{h}}$ is a vector of size T. The entry (t, t') of $\widehat{\boldsymbol{H}}$ is given by:

$$\widehat{H}_{t,t'} = \frac{\alpha}{n'}\sum_{i=1}^{n'} \varphi_t\left(\boldsymbol{x}'_i\right)\varphi_{t'}\left(\boldsymbol{x}'_i\right) + \frac{1-\alpha}{n}\sum_{i=1}^{n} \varphi_t\left(\boldsymbol{x}_i\right)\varphi_{t'}\left(\boldsymbol{x}_i\right) \tag{B.57}$$

The entry t of $\hat{\boldsymbol{h}}$ is given by:

$$\hat{h}_t = \frac{1}{n'}\sum_{i=1}^{n'} \varphi_t\left(\boldsymbol{x}'_j\right) \tag{B.58}$$

Like in the uLSIF's case, we can obtain the solution for $\boldsymbol{\beta}$ given analytically by:

$$\hat{\beta} = \max\left[0, \left(\widehat{\boldsymbol{H}} + \lambda \boldsymbol{I}_T\right)^{-1}\hat{\boldsymbol{h}}\right] \tag{B.59}$$

In the RuLSIF's case, the reasoning behind the choice of the basis functions is the same as presented in Section 4.4. One advantage of RuLSIF ($\alpha > 0$) over uLSIF is that the convergence of its algorithm is faster - that is, the sample size needed is smaller. On the other hand, it is not clear how to choose α, since the higher this value, the greater the bias we will have when estimating the statistical risk in the covariate shift framework, despite decreasing the variance of the weighted empirical risk estimator. Once again, the bias-variance trade-off comes into play.

The evaluation of the model and choice of hyperparameters is done in the same way as mentioned in Section B.3.1.

B.3.3 Kernel Unconstrained Least-Squares Importance Fitting (KuLSIF)

Another method that minimizes the mean squared error of a model w.r.t. the true importance function w is a kernel version of uLSIF (Kanamori *et al.*, 2012). This method's main idea is to assume that our function of interest w can be approximated by elements of a Reproducing Kernel Hilbert Space (RKHS) $\mathbb{H}$ associated with a positive semidefinite kernel K. Assuming that our model $w_{\mathbb{H}}$ is an element of $\mathbb{H}$, then we would like to minimize the following quadratic error function w.r.t. $w_{\mathbb{H}}$:

$$J(w_{\mathbb{H}}) = \frac{1}{2}\mathop{\mathbb{E}}_{\mathbf{x}\sim Q_{\mathbf{x}}}\left\{\left[w_{\mathbb{H}}(\mathbf{x}) - w(\mathbf{x})\right]^2\right\} \tag{B.60}$$

$$= \frac{1}{2}\mathop{\mathbb{E}}_{\mathbf{x}\sim Q_{\mathbf{x}}}\left[w_{\mathbb{H}}^2(\mathbf{x})\right] - \mathop{\mathbb{E}}_{\mathbf{x}\sim Q_{\mathbf{x}}}\left[w_{\mathbb{H}}(\mathbf{x})\, w(\mathbf{x})\right] + \frac{1}{2}\mathop{\mathbb{E}}_{\mathbf{x}\sim Q_{\mathbf{x}}}\left[w^2(\mathbf{x})\right] \tag{B.61}$$

$$= \frac{1}{2}\mathop{\mathbb{E}}_{\mathbf{x}\sim Q_{\mathbf{x}}}\left[w_{\mathbb{H}}^2(\mathbf{x})\right] - \mathop{\mathbb{E}}_{\mathbf{x}'\sim P_{\mathbf{x}}}\left[w_{\mathbb{H}}(\mathbf{x}')\right] + C \tag{B.62}$$

$$= J'(w_{\mathbb{H}}) + C \tag{B.63}$$

Where C is a constant term that does not depende on $w_{\mathbb{H}}$ and we can simply ignore it. Therefore, the main objetive here is to solve the following theoretical problem:

$$\min_{w_{\mathbb{H}}\in\mathbb{H}} \ \frac{1}{2}\mathop{\mathbb{E}}_{\mathbf{x}\sim Q_{\mathbf{x}}}\left[w_{\mathbb{H}}^2(\mathbf{x})\right] - \mathop{\mathbb{E}}_{\mathbf{x}'\sim P_{\mathbf{x}}}\left[w_{\mathbb{H}}(\mathbf{x}')\right] \tag{B.64}$$

$$\text{s.t. } w_{\mathbb{H}} \geq 0 \tag{B.65}$$

If there are samples $\{\boldsymbol{x}_i\}_{i=1}^{n}$ and $\{\boldsymbol{x}'_i\}_{i=1}^{n'}$ sampled from $Q_{\mathbf{x}}$ (source) and $P_{\mathbf{x}}$ (target), we write the empirical form of the previous problem:

$$\min_{w_{\mathbb{H}}\in\mathbb{H}} \ \frac{1}{2n}\sum_{i=1}^{n}\left[w_{\mathbb{H}}(\boldsymbol{x}_i)\right]^2 - \frac{1}{n'}\sum_{j=1}^{n'} w_{\mathbb{H}}\left(\boldsymbol{x}'_j\right) + \frac{\lambda}{2}||w_{\mathbb{H}}||_{\mathbb{H}}^2 \tag{B.66}$$

Note that as we did with uLSIF in Section B.3.1, we removed the constraint on the "variable" that minimizes the objective function of interest and put a regularization term

on it. Using the Representer Theorem (Kimeldorf and Wahba, 1971; Schölkopf *et al.*, 2002), a classical result of Statistics and Functional Analysis, we present the following theorem, stated by Kanamori *et al.* (2012), in order to solve the above optimization problem:

Theorem B.6. ***(Analytic solution of KuLSIF)****: Suppose $\lambda > 0$. Then, the KuLSIF estimator - the solution for problem B.66 - is given by:*

$$\tilde{w_{\mathbb{H}}}(.) = \sum_{i=1}^{n} \hat{\beta}_i K(., \boldsymbol{x}_i) + \frac{1}{n'\lambda} \sum_{j=1}^{n'} K(., \boldsymbol{x}_j) \tag{B.67}$$

Where the solution vector $\hat{\boldsymbol{\beta}} = (\hat{\beta}_1 \ s \ \hat{\beta}_n)$ is given as one solves the following linear system of equations:

$$\left(\frac{1}{n} \boldsymbol{K}_{11} + \lambda \boldsymbol{I}_n \right) \hat{\boldsymbol{\beta}} = -\frac{1}{\lambda n' n} \boldsymbol{K}_{12} \mathbf{1}_{n'} \tag{B.68}$$

Where $\boldsymbol{K}_{11}$ is a matrix with entries $(K_{11})_{i,j} = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and $\boldsymbol{K}_{12}$ is a matrix with entries $(K_{12})_{i,j} = K(\boldsymbol{x}_i, \boldsymbol{x}'_j)$. Furthermore, $\boldsymbol{I}_n$ is an identity matrix and $\mathbf{1}_{n'}$ is a vector of 1s of size n'. The linear system above can be solved by the matrix inversion method as well as by other methods - the best method may depend on the sample size.

Proof. Check Kanamori *et al.* (2012). □

The final approximation for w is given by:

$$\hat{w}_{\mathbb{H}} = \max(0, \tilde{w}_{\mathbb{H}}) \tag{B.69}$$

Kanamori *et al.* (2012) also shows there is another way to obtain the solution for KuLSIF, which is given by a non-analytic form as the one presented in Theorem B.6 - the alternative path would use a numerical optimization algorithm to obtain the solution.

As in the previous cases, when choosing the hyperparameters linked to kernels and regularization, one can efficiently use a Leave-One-Out-cross validation (LOOCV) approach, as presented in Section B.3.1 (Kanamori *et al.*, 2012).

Bibliography

Agapiou *et al.***(2017)** S Agapiou, Omiros Papaspiliopoulos, D Sanz-Alonso, AM Stuart *et al.* Importance sampling: Intrinsic dimension and computational cost. *Statistical Science*, 32(3):405–431. Cited on pages: 40, 42

Bishop(2006) Christopher M Bishop. *Pattern recognition and machine learning*. springer. Cited on pages: 84, 85

Chen and Guestrin(2016) Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. Em *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, páginas 785–794. Cited on pages: 25, 62, 63, 68

Cortes *et al.***(2010)** Corinna Cortes, Yishay Mansour and Mehryar Mohri. Learning bounds for importance weighting. Em *Advances in neural information processing systems*, páginas 442–450. Cited on pages: 39, 42, 43

Cortes *et al.***(2019)** Corinna Cortes, Spencer Greenberg and Mehryar Mohri. Relative deviation learning bounds and generalization with unbounded loss functions. *Annals of Mathematics and Artificial Intelligence*, 85(1):45–70. Cited on pages: 39, 43

Drineas and Mahoney(2005) Petros Drineas and Michael W Mahoney. Approximating a gram matrix for improved kernel-based learning. Em *International Conference on Computational Learning Theory*, páginas 323–337. Springer. Cited on pages: 27

Eirola *et al.***(2014)** Emil Eirola, Amaury Lendasse and Juha Karhunen. Variable selection for regression problems using gaussian mixture models to estimate mutual information. Em *2014 International Joint Conference on Neural Networks (IJCNN)*, páginas 1606–1613. IEEE. Cited on pages: 54

Elvira *et al.***(2018)** Víctor Elvira, Luca Martino and Christian P Robert. Rethinking the effective sample size. *arXiv preprint arXiv:1809.04129*. Cited on pages: 40

Golub and Van Loan(2012) Gene H Golub and Charles F Van Loan. *Matrix computations*, volume 3. JHU press. Cited on pages: 90

Goodfellow *et al.***(2016)** Ian Goodfellow, Yoshua Bengio and Aaron Courville. *Deep Learning*. MIT Press. Cited on pages: xvii

Gretton *et al.***(2009)** Arthur Gretton, Alex Smola, Jiayuan Huang, Marcel Schmittfull, Karsten Borgwardt and Bernhard Schölkopf. Covariate shift by kernel mean matching. *Dataset shift in machine learning*, 3(4):5. Cited on pages: 37

Guyon and Elisseeff(2003) Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182. Cited on pages: 54

Hastie *et al.*(**2009**) Trevor Hastie, Robert Tibshirani and Jerome Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media. Cited on pages: 7, 25, 47, 51

Huang *et al.*(**2006**) Jiayuan Huang, Arthur Gretton, Karsten Borgwardt, Bernhard Schölkopf and Alex J Smola. Correcting sample selection bias by unlabeled data. *Technical report, CS-2006-44, University of Waterloo*. Cited on pages: 81, 82, 83

Huang *et al.*(**2007**) Jiayuan Huang, Arthur Gretton, Karsten Borgwardt, Bernhard Schölkopf and Alex J Smola. Correcting sample selection bias by unlabeled data. Em *Advances in neural information processing systems*, páginas 601–608. Cited on pages: 16, 36, 37, 52, 79, 81, 82, 83

Izbicki *et al.*(**2014**) Rafael Izbicki, Ann Lee and Chad Schafer. High-dimensional density ratio estimation with extensions to approximate likelihood computation. Em *Artificial Intelligence and Statistics*, páginas 420–429. Cited on pages: 26, 27, 28, 36, 37

Kanamori *et al.*(**2009a**) Takafumi Kanamori, Shohei Hido and Masashi Sugiyama. A least-squares approach to direct importance estimation. *Journal of Machine Learning Research*, 10(Jul):1391–1445. Cited on pages: 16, 36, 89, 90

Kanamori *et al.*(**2009b**) Takafumi Kanamori, Taiji Suzuki and Masashi Sugiyama. Condition number analysis of kernel-based density ratio estimation. *arXiv preprint arXiv:0912.2800*. Cited on pages: 31, 33, 83

Kanamori *et al.*(**2012**) Takafumi Kanamori, Taiji Suzuki and Masashi Sugiyama. Statistical analysis of kernel-based least-squares density-ratio estimation. *Machine Learning*, 86(3):335–367. Cited on pages: 93, 94

Kimeldorf and Wahba(1971) George Kimeldorf and Grace Wahba. Some results on tchebycheffian spline functions. *Journal of mathematical analysis and applications*, 33(1): 82–95. Cited on pages: 94

Kull and Flach(2014) Meelis Kull and Peter Flach. Patterns of dataset shift. Em *First International Workshop on Learning over Multiple Contexts (LMCE) at ECML-PKDD*. Cited on pages: 60

Kullback(1997) Solomon Kullback. *Information theory and statistics*. Courier Corporation. Cited on pages: 29

Lan *et al.*(**2006**) Tian Lan, Deniz Erdogmus, Umut Ozertem and Yonghong Huang. Estimating mutual information using gaussian mixture model for feature ranking and selection. Em *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, páginas 5034–5039. IEEE. Cited on pages: 54

Liu *et al.*(**2017**) Song Liu, Akiko Takeda, Taiji Suzuki and Kenji Fukumizu. Trimmed density ratio estimation. Em *Advances in Neural Information Processing Systems*, páginas 4518–4528. Cited on pages: 36, 37, 87, 88, 89

Martino *et al.*(**2017**) Luca Martino, Víctor Elvira and Francisco Louzada. Effective sample size for importance sampling based on discrepancy measures. *Signal Processing*, 131:386–401. Cited on pages: 39, 40

Micchelli *et al.***(2006)** Charles A Micchelli, Yuesheng Xu and Haizhang Zhang. Universal kernels. *Journal of Machine Learning Research*, 7(Dec):2651–2667. Cited on pages: 80, 81

Minh(2010) Ha Quang Minh. Some properties of gaussian reproducing kernel hilbert spaces and their implications for function approximation and learning theory. *Constructive Approximation*, 32(2):307–338. Cited on pages: 27

Mohri *et al.***(2012)** Mehryar Mohri, Afshin Rostamizadeh and Ameet Talwalkar. Foundations of machine learning. adaptive computation and machine learning. *MIT Press*, 31:32. Cited on pages: 3, 7, 8

Moreno-Torres *et al.***(2012)** Jose G Moreno-Torres, Troy Raeder, Rocío Alaiz-Rodríguez, Nitesh V Chawla and Francisco Herrera. A unifying view on dataset shift in classification. *Pattern recognition*, 45(1):521–530. Cited on pages: 1, 25, 60

Owen(2013) Art B. Owen. *Monte Carlo theory, methods and examples.* . Cited on pages: 39, 40

Pedregosa *et al.***(2011)** Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg *et al.* Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830. Cited on pages: 56

Polo and Vicente(2021) Felipe Maia Polo and Renato Vicente. Effective sample size, dimensionality, and generalization in covariate shift adaptation. *arXiv preprint arXiv:2010.01184*. Cited on pages: 35

Polyanskiy and Wu(2019) Y Polyanskiy and Y Wu. Lecture notes on information theory. Cited on pages: 59, 61

Qiao and Minematsu(2010) Yu Qiao and Nobuaki Minematsu. A study on invariance of f-divergence and its application to speech recognition. *IEEE Transactions on Signal Processing*, 58(7):3884–3890. Cited on pages: 47, 48

Quionero-Candela *et al.***(2009)** Joaquin Quionero-Candela, Masashi Sugiyama, Anton Schwaighofer and Neil D Lawrence. *Dataset shift in machine learning*. The MIT Press. Cited on pages: 1, 11

Reddi *et al.***(2015)** Sashank Jakkam Reddi, Barnabas Poczos and Alex Smola. Doubly robust covariate shift correction. Em *Twenty-Ninth AAAI Conference on Artificial Intelligence*. Cited on pages: 37, 39, 52

Rhodes *et al.***(2020)** Benjamin Rhodes, Kai Xu and Michael U Gutmann. Telescoping density-ratio estimation. *arXiv preprint arXiv:2006.12204*. Cited on pages: 73, 75

Robert *et al.***(2010)** Christian P Robert, George Casella and George Casella. *Introducing monte carlo methods with r*, volume 18. Springer. Cited on pages: 39

Rosasco *et al.***(2010)** Lorenzo Rosasco, Mikhail Belkin and Ernesto De Vito. On learning with integral operators. *Journal of Machine Learning Research*, 11(30):905–934. Cited on pages: 26

Roussas(1997) George G Roussas. *A course in mathematical statistics*. Elsevier. Cited on pages: 41, 63

Sason and Verdú(2016) Igal Sason and Sergio Verdú. f-divergence inequalities. *IEEE Transactions on Information Theory*, 62(11):5973–6006. Cited on pages: 47

Schölkopf *et al.***(2002)** Bernhard Schölkopf, Alexander J Smola, Francis Bach *et al. Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press. Cited on pages: 20, 94

Shalev-Shwartz and Ben-David(2014) Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press. Cited on pages: 3, 6, 7

Shimodaira(2000) Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244. Cited on pages: 13, 16, 23, 35, 36

Sønderby *et al.***(2016)** Casper Kaae Sønderby, Jose Caballero, Lucas Theis, Wenzhe Shi and Ferenc Huszár. Amortised map inference for image super-resolution. *arXiv preprint arXiv:1610.04490*. Cited on pages: 62

Song *et al.***(2007)** Le Song, Alex Smola, Arthur Gretton, Karsten M Borgwardt and Justin Bedo. Supervised feature selection via dependence estimation. Em *Proceedings of the 24th international conference on Machine learning*, páginas 823–830. Cited on pages: 83

Steinwart(2001) Ingo Steinwart. On the influence of the kernel on the consistency of support vector machines. *Journal of machine learning research*, 2(Nov):67–93. Cited on pages: 79, 80

Stojanov *et al.***(2019)** Petar Stojanov, Mingming Gong, Jaime G Carbonell and Kun Zhang. Low-dimensional density ratio estimation for covariate shift correction. *Proceedings of machine learning research*, 89:3449. Cited on pages: 37, 52, 54

Sugiyama and Kawanabe(2012) Masashi Sugiyama and Motoaki Kawanabe. *Machine learning in non-stationary environments: Introduction to covariate shift adaptation*. MIT press. Cited on pages: 1, 13, 35, 36, 54, 60, 82

Sugiyama *et al.***(2007)** Masashi Sugiyama, Matthias Krauledat and Klaus-Robert MÃžller. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 8(May):985–1005. Cited on pages: 20, 36

Sugiyama *et al.***(2008)** Masashi Sugiyama, Taiji Suzuki, Shinichi Nakajima, Hisashi Kashima, Paul von Bünau and Motoaki Kawanabe. Direct importance estimation for covariate shift adaptation. *Annals of the Institute of Statistical Mathematics*, 60(4):699–746. Cited on pages: 16, 28, 30, 31, 36, 86, 88

Sugiyama *et al.***(2012a)** Masashi Sugiyama, Taiji Suzuki and Takafumi Kanamori. Density-ratio matching under the bregman divergence: a unified framework of density-ratio estimation. *Annals of the Institute of Statistical Mathematics*, 64(5):1009–1044. Cited on pages: 83, 84, 85

Sugiyama *et al.***(2012b)** Masashi Sugiyama, Taiji Suzuki and Takafumi Kanamori. *Density ratio estimation in machine learning*. Cambridge University Press. Cited on pages: 25, 54, 62

Suzuki and Sugiyama(2010) Taiji Suzuki and Masashi Sugiyama. Sufficient dimension reduction via squared-loss mutual information estimation. Em *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, páginas 804–811. Cited on pages: 54

Tiao(2018) Louis C Tiao. Density Ratio Estimation for KL Divergence Minimization between Implicit Distributions. *tiao.io*. URL https://tiao.io/post/density-ratio-estimation-for-kl-divergence-minimization-between-implicit-distributions/. Cited on pages: 62

Tipping and Bishop(1999) Michael E Tipping and Christopher M Bishop. Mixtures of probabilistic principal component analyzers. *Neural computation*, 11(2):443–482. Cited on pages: 85

Tsuboi *et al.*(2009) Yuta Tsuboi, Hisashi Kashima, Shohei Hido, Steffen Bickel and Masashi Sugiyama. Direct density ratio estimation for large-scale covariate shift adaptation. *Journal of Information Processing*, 17:138–155. Cited on pages: 86, 87

van Erven and Harremoës(2012) Tim van Erven and Peter Harremoës. Rényi divergence and kullback-leibler divergence. *arXiv preprint arXiv:1206.2459*. Cited on pages: 40, 41

Van Erven and Harremos(2014) Tim Van Erven and Peter Harremos. Rényi divergence and kullback-leibler divergence. *IEEE Transactions on Information Theory*, 60(7):3797–3820. Cited on pages: 44, 48, 52

Vershynin(2019) Roman Vershynin. High-dimensional probability, 2019. Cited on pages: 5

Vidyasagar(2002) Mathukumalli Vidyasagar. *A theory of learning and generalization*. Springer-Verlag. Cited on pages: 42

Wainwright(2019) Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press. Cited on pages: 26, 79, 80

Wang and Rudin(2017) Fulton Wang and Cynthia Rudin. Extreme dimension reduction for handling covariate shift. *arXiv preprint arXiv:1711.10938*. Cited on pages: 36, 37, 39, 42, 52

Wasserman(2006) Larry Wasserman. *All of nonparametric statistics*. Springer Science & Business Media. Cited on pages: 24

Yamada and Sugiyama(2009) Makoto Yamada and Masashi Sugiyama. Direct importance estimation with gaussian mixture models. *IEICE transactions on information and systems*, 92(10):2159–2162. Cited on pages: 83, 84

Yamada *et al.*(2010) Makoto Yamada, Masashi Sugiyama, Gordon Wichern and Jaak Simm. Direct importance estimation with a mixture of probabilistic principal component analyzers. *IEICE Transactions on Information and Systems*, 93(10):2846–2849. Cited on pages: 84, 85

Yamada *et al.*(2013) Makoto Yamada, Taiji Suzuki, Takafumi Kanamori, Hirotaka Hachiya and Masashi Sugiyama. Relative density-ratio estimation for robust distribution comparison. *Neural computation*, 25(5):1324–1370. Cited on pages: 91