**Eduardo Akira Kinto**

# Otimização e análise das máquinas de vetores de suporte aplicadas à classificação de documentos

Tese apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Doutor em Engenharia Elétrica.

São Paulo
2011

**Eduardo Akira Kinto**

# Otimização e análise das máquinas de vetores de suporte aplicadas à classificação de documentos

Tese apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Doutor em Engenharia Elétrica.

Área de concentração:
Sistemas Eletrônicos

Orientador:
Prof. Dr. Emilio Del Moral Hernandez

São Paulo
2011

Este exemplar foi revisado e alterado em relação à versão original, sob responsabilidade única do autor com a anuência de seu orientador.

São Paulo, 15 de Julho de 2011

_____
Eduardo Akira Kinto

_____
Emilio Del Moral Hernandez

## FICHA CATALOGRÁFICA

Kinto, Eduardo Akira.
Otimização e análise das máquinas de vetores de suporte aplicadas à classificação de documentos (**Edição Revisada**) / E.A. Kinto. - – São Paulo, 2011.
145 p.

Tese (Doutorado) – Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Elétrica. Área de concentração Sistemas Eletrônicos.

1. Aprendizado computacional. 2. Recuperação da informação. 3. Inteligência artificial. 4. Redes Neurais. I. Universidade de São Paulo. Departamento de Engenharia Elétrica. II. T.

# Agradecimentos

Gostaria de agradecer aos meus pais, pela ajuda e conselhos.

Aos meus irmãos que sempre me apoiaram.

Ao meu orientador Professor Emilio, pelo auxílio, dedicação e paciência constante durante todo o período de Doutorado.

Aos colegas de pesquisa, pelas idéias, discussões e ajuda em várias atividades.

Aos colegas de trabalho, pela flexibilidade nos horários e compreensão da importância desta pesquisa para o meu crescimento profissional.

Aos funcionários da POLI - Elétrica, pela atenção, respeito e serviços prestados em diversas ocasiões.

À Universidade de São Paulo, pela oportunidade e qualidade no ensino e na pesquisa.

A todos aqueles que, direta ou indiretamente, contribuíram para a realização desta tese.

## RESUMO

A análise das informações armazenadas é fundamental para qualquer tomada de decisão, mas para isso ela deve estar organizada e permitir fácil acesso. Quando temos um volume de dados muito grande, esta tarefa torna-se muito mais complicada do ponto de vista computacional. É fundamental, então, haver mecanismos eficientes para análise das informações.

As Redes Neurais Artificiais (RNA), as Máquinas de Vetores-Suporte (Support Vector Machine - SVM) e outros algoritmos são frequentemente usados para esta finalidade. Neste trabalho, iremos explorar o SMO (*Sequential Minimal Optimization*) e alterá-lo, com a finalidade de atingir um tempo de treinamento menor, mas, ao mesmo tempo manter a capacidade de classificação. São duas as alterações propostas, uma, no seu algoritmo de treinamento e outra, na sua arquitetura.

A primeira modificação do SMO proposta neste trabalho é permitir a atualização de candidatos ao vetor suporte no mesmo ciclo de atualização de um coeficiente de Lagrange. Dos algoritmos que codificam o SVM, o SMO é um dos mais rápidos e um dos que menos consome memória. A complexidade computacional do SMO é menor com relação aos demais algoritmos porque ele não trabalha com inversão de uma matriz de *kernel*. Esta matriz, que é quadrada, costuma ter um tamanho proporcional ao número de amostras que compõem os chamados vetores-suporte.

A segunda proposta para diminuir o tempo de treinamento do SVM consiste na subdivisão ordenada do conjunto de treinamento, utilizando-se a dimensão de maior entropia. Esta subdivisão difere das abordagens tradicionais pelo fato de as amostras não serem constantemente submetidas repetidas vezes ao treinamento do SVM.

Finalmente, é aplicado o SMO proposto para classificação de documentos ou textos por meio de uma abordagem nova, a classificação de uma-classe usando classificadores binários.

Como toda classificação de documentos, a análise dos atributos é uma etapa fundamental, e aqui uma nova contribuição é apresentada. Utilizamos a correlação total ponto a ponto para seleção das palavras que formam o vetor de índices de palavras.

## Abstract

Stored data analysis is very important when taking a decision in every business, but to accomplish this task data must be organized in a way it can be easily accessed. When we have a huge amount of information, data analysis becomes a very computational hard job. So, it is essential to have an efficient mechanism for information analysis.

Artificial neural networks (ANN), support vector machine (SVM) and other algorithms are frequently used for information analysis, and also in huge volume information analysis. In this work we will explore the sequential minimal optimization (SMO) algorithm, a learning algorithm for the SVM. We will modify it aiming for a lower training time and also to maintaining its classification generalization capacity. Two modifications are proposed to the SMO, one in the training algorithm and another in its architecture.

The first modification to the SMO enables more than one Lagrange coefficient update by choosing the neighbor samples of the updating pair (current working set). From many options of SVM implementation, SMO was chosen because it is one of the fastest and less memory consuming one. The computational complexity of the SMO is lower than other types of SVM because it does not require handling a huge Kernel matrix. Matrix inversion is one of the most time consuming step of SVM, and its size is as bigger as the number of support vectors of the sample set.

The second modification to the SMO proposes the creation of an ordered subset using as a reference one of the dimensions; entropy measure is used to choose the dimension. This subset creation is different from other division based SVM architectures because samples are not used in more than one training pair set.

All this improved SVM is used on a one-class like classification task of documents. Every document classification problem needs a good feature vector (feature selection and dimensionality reduction); we propose in this work a novel feature indexing mechanism using the pointwise total correlation.

# Índice


**1. Introdução** ..... 1
1.1 Resumo ..... 1
1.2 A importância da Classificação de Documentos ..... 2
1.3 Aprendizado de Máquina ..... 3
1.4 Reconhecimento de Padrão Estatístico ..... 3
1.5 Objetivos do Trabalho ..... 4
1.6 Considerações Finais do Capítulo ..... 5
1.7 Organização da Tese ..... 6
**2. *Support Vector Machine* – SVM** ..... 7
2.1 Introdução ..... 8
2.2 Função kernel ..... 14
2.3 Exemplos de funções *kernel* ..... 15
2.4 Otimização sequencial mínima ..... 16
2.4.1 O parâmetro de regularização ..... 18
2.5 Extensão do SMO de Platt ..... 21
2.6 SMO – Classificação de uma-classe ..... 24
2.7 Exemplos de Aplicação do SVM ..... 27
2.7.1 Recuperação de Imagem ..... 27
2.7.2 Diagnóstico Médico Auxiliado por Computador ..... 29
**3. Aprendizado com SVM** ..... 31
3.1 Aprendizado baseado em regras ..... 32
3.2 Aprendizado Incremental ..... 35
**4. Descrição do Domínio de Dados usando Vetores-suporte - SVDD** ..... 42
4.1 Classificação de Uma-Classe segundo Schölkopf ..... 45
4.2 Exemplo de Classificação de Uma-Classe – Detecção de Intrusão ..... 48
**5. Análise dos Atributos** ..... 50
5.1 Base de dados ..... 50
5.1.1 Reuters-21578 – base de dados textual ..... 50
5.1.2 20 Newsgroup – base de dados textual ..... 52
5.1.3 UCI – Adult dataset ..... 53
5.1.4 UCI – Mushroom dataset ..... 53
5.2 Representação dos Documentos ..... 53
5.3 Análise dos Atributos – Seleção dos Atributos ..... 58
5.4 Agrupamento de Textos e Atributos ..... 63
5.5 Recursive Feature Elimination - RFE ..... 66
**6 Indexação das palavras usando correlação total ponto a ponto** ..... 68
6.1 Correlação total ponto-a-ponto - PTC ..... 69
6.2 A abordagem uma-classe ..... 71
6.3 Experimentos – classificação de uma-classe – Reuters-21578 ..... 73
6.4 Experimentos – classificação de uma-classe – 20 Newsgroup ..... 76
6.4.1 Resultados - 20Newsgroup – Pointwise-total-correlation ..... 78
**7 SVM baseado na atualização dos coeficientes de Lagrange vizinhos - NSVM** ..... 80
7.1 Experimentos com NSMO ..... 83
7.2 Experimentos - Classificação de uma-classe, usando classificadores binários 85
**8 Maximal trace subset SVM** ..... 90
8.1 Cascade SVM ..... 90
8.2 Maximal trace subset SVM – MTS-SVM ..... 93
8.3 Experimentos - Resultados do MTS-SVM com dados da 20Newsgroup ..... 101

8.4 Experimentos - Resultados do MTS-SVM – Mushroom dataset ..... 104
**9 Conclusão e comentários finais** ..... 105
9.1 Comentários e Limitações ..... 108
9.2 Trabalhos Futuros ..... 109
9.3 Considerações Finais ..... 110
**Apêndice A - A teoria da otimização** ..... 118
O problema de otimização (*primal*) ..... 120
Teorema de Fermat ..... 122
Método dos multiplicadores de Lagrange ..... 122
O problema de otimização (dual) ..... 124
**Apêndice B – Lógica Proposicional** ..... 126
**Apêndice C – Formula Sherman-Morrison-Woodbury** ..... 127
**Apêndice D – Termos técnicos usados em redes de comunicação** ..... 128
Análise n-grama ..... 128
**Apêndice E – Representação dos atributos – 20Newsgroup** ..... 130
**Apêndice F – Teste de Hipóteses utilizando a distribuição t de Student** ..... 143

# Lista de Abreviaturas

| | |
|---|---|
| ANN | *Artificial Neural Network* |
| ATN | Notação SMART Capítulo 5. TF modificado, IDF e sem normalização |
| BOW | *Bag of Words* |
| CBIR | *Content Based Image Retrieval* |
| CNF | *Conjunctive Normal Form* |
| EM | *Expectation Maximization* |
| ERM | *Empirical Risk Minimization* |
| $F_1$ | Média geométrica entre precisão e revocação |
| FP/FN | Falso positivo/Falso negativo |
| GR | *Gain Ratio* |
| ICF | *Inverse Category Frequency* |
| IDS | *Intrusion Detection System* |
| KKT | Karush-Kuhn-Tucker |
| LDA | Linear Discriminate Analysis |
| MA | Máquina de Aprendizado |
| MI | *Mutual Information* |
| MVP | Maximal Violating Pair |
| NNN | Notação SMART Capítulo 5. Equivale ao TF na representação de um termo |
| NTC | Notação SMART Capítulo 5. TF*IDF é normalizado |
| OCSVM | *One-class SVM* |
| PCA | *Principal Component Analysis* |
| PL | Programação Linear |
| PQ | Programação Quadrática |
| QBPE | *Query by Pictorial Example* |
| RBF | *Radial Basis Function* |
| RF | *Relevance Feedback* |
| RNA | Redes Neurais Artificiais |
| RPE | Reconhecimento de Padrões Estatísticos |
| SMO | *Sequential Minimal Optimization* – Otimização Sequencial Mínima |
| SOM | *Self-Organizing Map* |
| SVDD | *Support Vector Data Description* |
| SRM | *Structural Risk Minimization* |
| SVM | *Support Vector Machine* – Máquina de Vetor Suporte |
| SV | *Support Vectors* – Vetores-suporte |
| SVC | *Support Vector Clustering* |
| TF | *Term Frequency* – frequência de ocorrência de um termo |
| VC | Vapnik e Chervonenkis – dimensão VC |
| VP/VN | Verdadeiro positivo/Verdadeiro negativo |
| VS | *Vector space* – Representação vetorial dos documentos através das palalvras |
| WTA | *Winner-Takes-All* |
| 20NG | 20 Newsgroup – Amostras para treinamento do SVM |

# Convenções e Lista de Símbolos

Na notação das fórmulas, há as seguintes convenções:

- Em negrito, um vetor, por exemplo, $\mathbf{x}_t$ , onde t ∈ T, T um conjunto qualquer

Os seguintes símbolos são utilizados nesta tese:

| | |
|---|---|
| $\alpha_i$ | Multiplicadores de Lagrange de uma amostra i |
| a | Centro de uma hiper-esfera numa SVC ou SVDD |
| b | *Bias* (tendência), parâmetro de ajuste do SVM |
| $\beta$ | Parâmetro de ajuste do bias b no algoritmo do SMO |
| C | Parâmetro de *tradeoff* numa SVDD ou parâmetro de regularização no SVM |
| $\xi$ | Parâmetro de ajuste – *slack variable* – usada no SVC, SVM |
| $\varepsilon$ | Parâmetro de ajuste – *slack variable* |
| $\gamma$ | Indica a largura da função de base radial gaussiana |
| h | Dimensão VC |
| $I_{up/low}$ | Subconjuntos criados e usados no treinamento de um SMO |
| $\rho$ | Margem a ser ajustado no SVDD ou One-class SVM |
| $y_t$ | Identificação da classe à qual pertence $\mathbf{x}_t$ |
| t | índice utilizado para identificar amostras de uma classe, t=1,2,..., N |
| $f(\mathbf{x})$ | Função que realiza o mapeamento de amostras, **x**, de uma dimensão para outra |
| $L$ | Função Lagrangeano do problema de otimização primal |
| $K(\mathbf{a},\mathbf{b})$ | Função *kernel* calculado usando os vetores **a** e **b** |
| K | Quantidade de classes distintas em um processo de multi-classificação |
| M | Margem de separação entre os hiperplanos positivos e negativos |
| N | Quantidade de amostras de treino |
| n | Dimensão das amostras |
| $\eta$ | Passo de ajuste dos coeficientes de Lagrange utilizado no SMO |
| $P(x,y)$ | É uma função distribuição desconhecida das amostras x |
| R | Raio de uma SVDD ou SVC |
| $R(\boldsymbol{q})$ | Função Risco associado ao parâmetro $\mathbf{q}=(\mathbf{w},b)$ |
| $R^{emp}(q)$ | Função Risco empírico associado ao parâmetro $\mathbf{q}=(\mathbf{w},b)$ |
| $\Re$ | Conjunto dos números reais |
| S | Conjunto de pares formados pelas amostras **x** e sua identificação y |
| sgn($f(\mathbf{x})$ | Função sinal: +1 se $f(x)\geq 0$ e -1 caso contrário |
| $\nu$ | Usada no SVC, SVDD. É um parâmetro de controle ou ajuste. |
| xp | Amostras da classe positiva |
| xn | Amostras da classe negativa |
| X | Espaço das amostras |
| $\Phi$ | Espaço das transformadas |
| **w** | Peso do SVM |
| wo | Representa uma palavra em uma coleção de documentos |
| $W$ | É a função dual da função Lagrangeano $L$. |
| z | Amostra de teste |

# Lista de Figuras

Figura 2.1 *Support vector machine*, ilustração de um classificador binário. As amostras positivas estão representadas por círculos, enquanto que as negativas, por quadrados. Os eixos ordenados representam as dimensões de cada amostra. ..... 9
Figura 2.2. Vetores-suporte demarcando as linhas de separação entre duas classes. Os vetores-suporte são as amostras que ficam na fronteira de separação das classes. Eles estão destacados na figura com um círculo ao seu redor. ..... 10
Figura 2.3. Margem de separação entre hiperplanos nas fronteiras das classes positiva (quadrado preenchido) e negativa (círculo preenchido). No SVM queremos que a margem M seja maximizada. Os eixos ordenados x1 e x2 representam as dimensões da amostras no espaço 2D ..... 12
Figura 2.4. Exemplo de amostras distribuídas de tal forma que um classificador linear não é capaz de separá-las. ..... 14
Figura 2.5. Exemplo de aplicação da função *kernel*. Amostras inicialmente não separáveis linearmente podem ser separadas no espaço das transformadas. Adaptado de (SCHOLKOPF; SMOLA, 2002) ..... 15
Figura 3.1. CPAR - Abrangência das regras de associação. Cada conjunto de regras consegue discriminar bem um conjunto de amostras. O objetivo é ter N regras que abranjam o conjunto todo ao invés de ter uma "super" regra. ..... 33
Figura 3.2. Regras de Associação - Abrangência das regras. Regras muito genéricas comprometem a qualidade do classificador, fornecendo falso positivo. ..... 35
Figura 3.3. SVM e os elementos que serão apagados (amostras na região sombreada). Verificamos nesta figura a separação linear de duas classes. Amostras com fundo preenchido são aquelas já existentes, e aquelas com fundo não preenchido são amostras novas. Extraído de (KATAGIRI; ABE, 2006) ..... 38
Figura 3.4. SVM incremental no qual se trabalha com subconjunto das amostras. Hiperplano ótimo das amostras com fundo preenchido. Amostras com fundo branco são os novos elementos que serão introduzidos na iteração seguinte. Extraído de (KATAGIRI; ABE, 2006) ..... 39
Figura 3.5. SVM, iteração 2. Rotação do Hiperplano ótimo de separação após introdução de novas amostras e remoção das antigas. Extraído de (KATAGIRI; ABE, 2006) ..... 39
Figura 3.6. SVM incremental com hiperesfera de raios $R_1$ e $\rho R_2$ e hiper-cone formado usando-se de um ângulo $\theta$ ajustável. Extraído de (KATAGIRI; ABE, 2006) ..... 40
Figura 3.7. O processo de aprendizado incremental. Extraído de (SYED, et al., 1999). Divisão do conjunto completo em K subconjuntos. Para cada subconjunto aplica-se o SVM, obtendo-se assim um conjunto de vetores-suporte. ..... 41
Figura 4.1. Classificação de uma-classe (são consideradas apenas as amostras de uma classe, por exemplo, as maçãs representadas por *). Extraído de (TAX; DUIN, 1999) ..... 42
Figura 4.2. Ilustração do conceito de classificação de uma-classe. Dadas apenas amostras da classe positiva, uma amostra da classe negativa pode estar localizada na origem. A origem é o único membro da classe negativa. Extraído de (MANEVITZ; YOUSEF, 2001). ..... 46
Figura 4.3. Rede "feed-forward" com gargalo (*bottleneck*). Extraído de (MANEVITZ; YOUSEF, 2006) ..... 47
Figura 6.1. Distribuição das ocorrências de cada palavra da categoria "earn" ..... 75
Figura 7.1 Os quadrados preenchidos ($x_2$) representam as amostras selecionadas na otimização conjunta com $x_1$ durante o treinamento do SMO. Esses dois quadrados, na verdade, serão unificados; o processo de junção (*shrinking*) irá eliminar um deles. As amostras na vizinhança de $x_2$ estão representadas como círculo preenchido. ..... 80

Figura 7.2 Tempo de treinamento para o SMO de Keerthi (triângulo) e o nosso SMO com vizinhança - NSMO (quadrados) à medida que se aumenta o conjunto de treinamento. A base de dados usada foi a *chessboard*. NSMO tem um tempo de treinamento menor e essa diferença ..... 85
Figura 8.1 Arquitetura do SVM em cascata binária. Figura extraída de (HANS et al., 2004). 91
Figura 8.2 Ilustração do conceito de divisão do conjunto de treino em N subconjuntos. À esquerda e ao centro, dois subconjuntos onde as figuras (círculos e quadrados) sólidas representam as amostras do subconjunto. As figuras com molduras representam os vetores-suporte. Em todos os casos, a linha tracejada representa a superfície separadora ideal, e a linha sólida a superfície separadora obtida. Figura extraída de (Hans et al., 2004). ..... 92
Figura 8.3 Base de dados no formato de tabuleiro de xadrez. As amostras destacadas em forma de quadrados e círculos representam uma possível formação de um subconjunto. O "tabuleiro" de xadrez representaria o conjunto completo. ..... 92
Figura 8.4 Arquitetura MTS-SVM proposta. K representa a quantidade de amostras do conjunto completo, e os quadrados representam instâncias de SVM. ..... 94
Figura 8.5 Distribuição aleatória. Entropia de 4.5989 ..... 95
Figura 8.6 Distribuição concentrada. Entropia de 4.3414 ..... 95
Figura 8.7 Distribuição concentrada. Entropia de 3.9991 ..... 96
Figura 8.8 Distribuição não uniforme. Entropia de 4.0351 ..... 96
Figura 8.9 Distribuição exponencial. Entropia de 0.1538 ..... 97
Figura 8.10 Base de dados cuja distribuição das amostras lembra duas bananas, uma classe representada por quadrados e outra classe por círculos. Subdividindo o conjunto em três partes, podemos usar, neste caso exemplo, um *kernel* linear para a classificação das amostras de cada subconjunto. ..... 97
Figura 8.11 Ilustração do procedimento para obtenção do traço de um segmento de amostras. Divide-se o conjunto completo em pequenas unidades segundo uma dimensão como, por exemplo, a dimensão x1. A subdivisão ideal será um múltiplo da unidade fundamental, o traço. ..... 98
Figura A.1. Árvore ilustrando a ciência da otimização. Adaptado de (ARGONNE, 2005)... 118
Figura A.2. Regiões de domínio viável (fonte: notas de aula Prof. Dr. Emílio Silva [SILVA, 2004]). O domínio viável corresponde o interior da região delimitada pelas superfícies $h_1$, $h_2$, $h_3$ e $h_4$. As linhas tracejadas são as curvas de níveis. ..... 119

# Lista de Tabelas

Tabela 5.1 Combinação entre os possíveis resultados numa classificação. VP=Verdadeiro Positivo, VN=Verdadeiro Negativo, FP=Falso Positivo, FN=Falso Negativo. ..... 57
Tabela 5.2. Média e desvio padrão de todos os resultados da classificação usando cada um dos subconjuntos da Reuters-21578, R10,R90 e R115. Extraído de (DEBOLE; SEBATIANI, 2004). ..... 58
Tabela 5.3. Cálculo da divergência de Kullback-Leibler entre classes de documentos. ..... 60
Tabela 5.4. Resultados da classificação de uma-classe usando a base de dados da Reuters.... 61
Tabela 6.1. Categoria "earn" (à esquerda) e “acq” (à direita), obtida da Reuters-21578. Na coluna ocorrência está o total de ocorrências da palavra correspondente. A coluna “ranking” informa a posição da palavra entre aquelas que mais repetem. ..... 73
Tabela 6.2 Categoria "crude" (a esquerda) e “grain” (direita), obtido da Reuters-21578. ..... 74
Tabela 6.3 Categoria "money-fx". ..... 74
Tabela 6.4. Quantidade de documentos divididos em amostras de treino e teste, segundo a divisão originalmente realizada e disponível na base da Reuters-21578. ..... 75
Tabela 6.5 Nomenclatura usada para representação das classes da coleção 20NG ..... 76
Tabela 6.6 Classe 01 – Comparativo entre três representações dos documentos. Média e desvio do valor de F1. ..... 77
Tabela 6.7 Classe 02 – Comparativo entre três representações dos documentos. Média e desvio do valor de F1. ..... 77
Tabela 6.8 Classe 10 – Comparativo entre três representações dos documentos. Média e desvio do valor de F1. ..... 77
Tabela 6.9 Classe 18 – Comparativo entre três representações dos documentos. Média e desvio do valor de F1. ..... 78
Tabela 6.10 Classe-01. Palavra-simples vs PTC. Medidas F1. ..... 78
Tabela 6.11 Classe-10. Palavra-simples vs PTC. Medidas F1. ..... 79
Tabela 6.12 Classe-14. Palavra-simples vs PTC. Medidas F1. ..... 79
Tabela 6.13 Classe-18. Palavra-simples vs PTC. Medidas F1. ..... 79
Tabela 7.1 NSMO ganhos em tempo de treinamento comparando com o SMO de Keerthi. Base de dados de Reuters, a Reuters-21578. Foi realizado um treinamento 2-fold com 5 repetições. ..... 84
Tabela 7.2. À esquerda, resultados da classificação de Joachims 2000, ao centro, Naïve Bayes, Bayes Net, Find Similar, Decision Tree e SVM de Dumais,e a direita os resultados da classificação usando a correlação total ponto-a-ponto com atualização da vizinhança (NSVM). Em itálico e fonte de texto menor, os resultados inferiores à nossa técnica e em negrito em fonte de texto maior, resultados que superam a nossa técnica. ..... 85
Tabela 7.3 NSMO ganhos em tempo de treinamento comparando com o SMO de Keerthi. Base de dados 20-*Newsgroup*. Análise preliminar. ..... 86
Tabela 7.4 Classe 14 vs 07. Comparação de tempos, NSMO vs SMO. ..... 87
Tabela 7.5 Classe 14 vs 02. Comparação de tempos, NSMO vs SMO. ..... 87
Tabela 7.6 Classe 14 vs 01. Comparação de tempos, NSMO vs SMO. ..... 87
Tabela 7.7 Classe 14 vs 18. Comparação entre os valores F1 do NSMO e SMO. ..... 88
Tabela 7.8 Classe 14 vs 10. Comparação entre os valores F1 do NSMO e SMO. ..... 88
Tabela 7.9 Classe 14 vs 07. Comparação entre os valores F1 do NSMO e SMO. ..... 88
Tabela 7.10 Classe 14 vs 02. Comparação entre os valores F1 do NSMO e SMO. ..... 88
Tabela 7.11 Classe 14 vs 01. Comparação entre os valores F1 do NSMO e SMO. ..... 88
Tabela 7.12 $t_{calc}$ dos valores F1 listados nas tabelas 7.7 a 7.11 ..... 88

Tabela 8.1 Base de dados UCI-*adult dataset* dividindo a base em 13 subconjuntos, usando a idade como referência na divisão. Existe uma sobreposição entre os grupos para não limitar a capacidade de generalização do SVM. ..... 100
Tabela 8.2 Base de dados UCI-*adult dataset*. Tempo de treinamento de cada um dos 13 subconjuntos. Foi realizado um treinamento 2-fold com 5 repetições. ..... 100
Tabela 8.3 Treinamento dos 5 subconjuntos do grupo 5 da Tabela 8.2. Ilustração do conceito de divisão de conjuntos. A soma dos tempos de treinamento dos cinco subconjuntos é muito menor do que o tempo do grupo 5 na Tabela 8.2. ..... 101
Tabela 8.4 Classe 1 contra a classe 2 ..... 102
Tabela 8.5 Classe 1 contra a classe 7 ..... 102
Tabela 8.6 Classe 1 contra a classe 10. Comparação entre MTS-SVM e SVM. ..... 102
Tabela 8.7 Classe 1 contra a classe 14. Comparação entre MTS-SVM e SVM. ..... 102
Tabela 8.8 Classe 1 contra a classe 18. Comparação entre MTS-SVM e SVM. ..... 102
Tabela 8.9 Classe 1 contra a classe 2. Capacidade de generalização. ..... 103
Tabela 8.10 Classe 1 contra a classe 7 Capacidade de generalização. ..... 103
Tabela 8.11 Classe 1 contra a classe 10 Capacidade de generalização. ..... 103
Tabela 8.12 Classe 1 contra a classe 14 Capacidade de generalização. ..... 103
Tabela 8.13 Classe 1 contra a classe 18 Capacidade de generalização. ..... 103
Tabela 8.14 Resultados experimentais do MTS-SVM para dados de cogumelos. ..... 104
Tabela E.1 Classe 01, representação dos documentos utilizando a combinação ATN. ..... 130
Tabela E.2 Classe 01, representação dos documentos utilizando a combinação NTC. ..... 131
Tabela E.1 Classe 01, representação dos documentos utilizando a combinação NNN. ..... 132
Tabela E.2 Classe 02, representação dos documentos utilizando a combinação ATN. ..... 133
Tabela E.3 Classe 02, representação dos documentos utilizando a combinação NTC. ..... 134
Tabela E.4 Classe 02, representação dos documentos utilizando a combinação NNN. ..... 135
Tabela E.5 Classe 18, representação dos documentos utilizando a combinação ATN. ..... 136
Tabela E.6 Classe 18, representação dos documentos utilizando a combinação NTC. ..... 137
Tabela E.7 Classe 18, representação dos documentos utilizando a combinação NNN. ..... 138
Tabela E.8 Classe 10, representação dos documentos utilizando a combinação ATN. ..... 139
Tabela E.9 Classe 10, representação dos documentos utilizando a combinação NTC. ..... 140
Tabela E.10 Classe 10, representação dos documentos utilizando a combinação NNN. ..... 141

# 1. Introdução

## 1.1 Resumo

Esta tese discute pesquisas realizadas em classificação de textos ou documentos; novas propostas foram introduzidas otimizando o algoritmo de classificação, analisando a técnica classificação de uma-classe e propondo novas formas de análise de atributos. Em todos os casos, foi utilizada a teoria do aprendizado baseado na Máquina de Vetor Suporte (*Support Vector Machine* - SVM), introduzida, na década de 60, por Vapnik (Vapnik e Chervonenkis, 1964).

A classificação de uma-classe tem recebido muita atenção nos últimos anos (TAX; DUIN, 1999; SCHÖLKOPF et al., 1999) para tratar dos problemas de classificação binária (duas classes) ou das multiclasses em que a quantidade de amostras entre as diferentes classes não é uniforme.

Esta não uniformidade, e em muitos casos a falta de amostras representativas da classe, acarreta a geração de um classificador com baixo índice de generalização. Assim, a classificação de uma-classe tem por objetivo realizar uma análise focada em apenas uma classe; esta análise é apresentada nesta tese utilizando-se as máquinas de vetores-suporte.

O SMO (do termo *Sequential Minimal Optimization*) (PLATT, 1999) foi otimizado e modificado para reduzir o tempo de treinamento do classificador. A otimização do SMO foi feita por meio da atualização de múltiplos Lagrangeanos vizinhos ao Lagrangeano que mais viola a condição de otimização. A segunda proposta para redução no tempo de treinamento do SVM foi conduzida através da subdivisão (permanente) do conjunto de amostras.

Quando aplicamos o SVM na vida real, a quantidade de amostras a serem usadas no treinamento pode ser muito grande. Este fato tem um impacto direto na performance do SVM já que o processo iterativo de aprendizado considera quase todas essas amostras, muitas das quais, redundantes. Para contornar esse problema, um novo conceito de pré-seleção de amostras candidatas ao vetor suporte é apresentado.

Outro fator muito importante na classificação de documentos é a análise dos atributos dos documentos que compõem as amostras de treinamento do SVM. Os atributos são as palavras ou conjunto de palavras. Assim, dado um documento, percebe-se que é gerado um vetor de atributos muito esparso, já que o vetor de atributos completo é representado por todas as palavras que apareceram nos documentos que compõem a classe.

A seguir, serão apresentados os motivos pela escolha do tema, 'classificação de documentos'.

## 1.2 A importância da Classificação de Documentos

Com a disponibilidade de um volume de dados muito grande, precisamos de mecanismos eficientes para extrair as informações que nos interessam. Muitas vezes, não sabemos exatamente o que buscamos, por exemplo, queremos realizar um teste ergométrico para avaliação das condições cardiorrespiratórias, mas desconhecemos o termo (ou palavra) técnico a usar na procura. Buscamos, então, pelo que queremos por meio do uso de outros termos, na esperança de encontrar o que desejamos.

Existem ainda situações em que sabemos exatamente o que queremos e encontramos a informação desejada, mas outras informações interessantes - que nem imaginávamos - estão ao redor. Por exemplo, quando procuramos por um livro numa biblioteca ou livraria podemos ir direto ao livro que queremos, mas, olhando ao redor, podemos ver dezenas de outros títulos ou temas novos que também são de nosso interesse.

A análise exploratória de dados acima apresentada é hoje uma forma comumente usada em mecanismos como Recuperação da Informação (*Information Retrieval*), da qual - dada uma ou mais palavras-chave - queremos obter textos ou documentos que contenham essa palavra-chave. No entanto, podem existir textos sem a palavra-chave utilizada na procura que apresentam conteúdo (semântica) similar aos que retornaram na busca.

Uma forma muito útil de visualizar textos ou documentos com conteúdos parecidos é por meio de um mapa ou mapeamento semântico. O mapa semântico para recuperação da informação tem sido estudado desde a década de 60 por Doyle (DOYLE, 1961). Seu objetivo é facilitar buscas na literatura, fornecendo informações correlatas.

Neste trabalho usaremos o SVM, uma técnica de aprendizado de máquina que se baseia no princípio de aprendizado estatístico. A seguir, uma breve introdução sobre aprendizado de máquina é apresentada.

## 1.3 Aprendizado de Máquina

Aprendizado de Máquina se inspira nos conceitos de diversas áreas de pesquisas como estatística, inteligência artificial, filosofia, teoria da informação, biologia, ciência cognitiva, complexidade computacional e teoria do controle.

Alguns casos de sucesso na aplicação de Aprendizado de Máquina são:

- Reconhecimento de palavras faladas – (WAIBEL, 1989; LEE, 1989);
- Predição da taxa de recuperação de pacientes com pneumonia – (COOPER et al., 1997);
- Detecção de uso fraudulento de cartões de crédito – (POMERLEAU, 1989);
- Possibilidade de jogar *backgammon* – (TESAURO, 1992; TESAURO, 1995);
- Classificação de tumores – (SILVA, 2008);
- Recuperação de imagens médicas – (RAHMAN; BHATTACHARYA; DESAI, 2007; LEHMANN et al., 2005)

O aprendizado (MITCHELL, 1997; SEBASTIANI, 2002) considerado aqui é qualquer programa que apresente condição de melhorar sua performance na execução de uma tarefa, a partir da experiência. Em geral, aprendizado de máquina está relacionado com Estatística, ou seja, com padrões existentes nas amostras que serão utilizadas no treinamento e que representem as características mais comuns. Mas, às vezes, esse padrão pode não existir. Assim, muitas técnicas de aprendizado de máquina se baseiam no reconhecimento de padrão estatístico, veja a seguir uma breve descrição sobre o Reconhecimento de Padrão Estatístico.

## 1.4 Reconhecimento de Padrão Estatístico

Reconhecimento de padrões é algo que o nosso cérebro realiza a todo instante. Quando reconhecemos a face de uma pessoa, o formato de um objeto ou nos lembramos de situações já vividas, estamos reconhecendo algum padrão armazenado em nosso cérebro. Esta tarefa, natural para nós, é bem complicada para ser programada em computador.

Quando se fala em reconhecimento de padrões (BISHOP, 1995), isto significa uma gama muito ampla de aplicações (processamento de informações). Três aplicações estão listadas a seguir:

- Reconhecimento do locutor (*speech recognition*) - (ERIKSSON et al., 2005)
- Classificação de padrões de escrita manual - (LING; LEUNG; LAM, 2006)
- Identificação de falhas em diagnósticos médicos e mecânicos

Muitas dessas atividades são facilmente executadas por nós, mas quando tentamos criar um programa para desempenhar a mesma função, verificamos que a dificuldade é muito grande.

É necessário, então, realizar um estudo minucioso com um bom embasamento teórico a fim de buscar um *framework* que facilite as atividades de Reconhecimento de Padrões. Um desses *frameworks* tem uma grande influência da Estatística - que reconhece a natureza probabilística das informações que processamos - e nas maneiras com que devemos expressar os resultados.

Reconhecimento de Padrões Estatísticos (RPE) é um campo de pesquisa bem antigo. O aprendizado de máquina computacional usa conceitos da RPE - adicionando-se a teoria da computação para ganhar poder de processamento e tornar viáveis muitas das equações e teorias que antigamente eram muito complexas.

A partir das considerações apresentadas nas duas seções anteriores, os objetivos deste trabalho foram desenvolvidos e são apresentados logo a seguir.

## 1.5 Objetivos do Trabalho

A classificação de duas classes ou a classificação binária tem sido amplamente estudada nas últimas décadas. No entanto, o conceito de classificação de uma-classe é algo mais recente. Este trabalho tem como finalidade propor melhorias aplicadas na classificação tradicional (a classificação binária) e ao mesmo tempo analisar a classificação de uma-classe. Um dos principais desafios no treinamento de qualquer Máquina de Aprendizado (MA) é a redução no tempo de treinamento, geralmente associada ao volume de dados muito grande.

Embora a capacidade de processamento da informação tenha aumentado muito, o volume de dados também tem crescido. Assim, é proposto neste trabalho duas formas para reduzir o tempo de treinamento, mas ao mesmo tempo mantendo a capacidade de generalização da Máquina de Aprendizado.

Os objetivos apresentados acima são buscados neste trabalho através das seguintes inovações:

- Uma nova metodologia de análise de atributos. Esta etapa é fundamental em qualquer classificador de textos devido à dimensão das amostras. É utilizada a correlação total ponto-a-ponto como nova proposta para representação dos documentos.
- Melhoria do SMO, otimizando-o no seu processo de atualização dos coeficientes Lagrangeanos. A cada iteração, não só o par de amostras que mais violam a condição KKT será atualizado, mas também seus vizinhos serão atualizados conforme critérios apresentados neste trabalho.
- SVM baseado em subconjunto de amostras. Uma abordagem inovadora com relação às técnicas tradicionais.

Os experimentos serão conduzidos utilizando-se base de dados pública como a Reuters-21578 (LEWIS, 1997) e a 20Newsgroup - também conhecida como 20NG – (LANG, 1995). Ambas as bases são apresentadas no Capítulo 5. A Reuters-21578 é composta por um conjunto de notícias previamente classificadas. É uma base de tamanho médio (menos de 30.000 notícias) com muitas categorias e poucas amostras; mesmo assim ela tem sido usada como referência em trabalhos de classificação de texto. A 20NG é uma outra base de dados pública contendo textos pré-categorizados das listas de discussão na *internet*.

## 1.6 Considerações Finais do Capítulo

Foram apresentados neste capítulo a proposta de trabalho, um breve resumo, os motivos que levaram ao desenvolvimento deste trabalho e alguns conceitos sobre Aprendizado de Máquina e Reconhecimento de Padrões Estatísticos.

O SVM é uma técnica de aprendizado de máquina fortemente embasada nos conceitos de aprendizado de máquina estatística. Sua formulação, conforme veremos nos próximos capítulos, é totalmente matemática (teoria da otimização matemática), mas incorpora a definição de limites superiores na capacidade de generalização.

## 1.7 Organização da Tese

Este trabalho está organizado da seguinte forma:

- No Capítulo 2, apresentamos o SVM, o SMO e aspectos teóricos são abordados. No Apêndice A, um embasamento sobre a teoria de otimização, fundamental para o entendimento do SVM, é apresentado.
- No Capítulo 3, diversos tipos de aprendizado são apresentados. A solução para uma preocupação crescente - o aumento no volume de dados - é aprender apenas os 'conceitos' novos; o aprendizado incremental é apresentado neste capítulo. Nossa proposta difere desses tipos de aprendizado. Maiores detalhes são apresentados nos Capítulos 7 e 8.
- No Capítulo 4, descrição do domínio – Pretexto para classificação de uma-classe – é apresentada.
- No capitulo 5, são tratados aspectos da análise de atributo, além disso, as bases de dados utilizadas nesta tese são apresentadas. Toda informação deve ser pré-processada, gerando como resultado atributos que caracterizam a informação. Exploramos a análise de atributos mostrando técnicas usadas na literatura.
- No Capítulo 6, apresentamos uma nova forma de indexar os atributos de um documento. A classificação de documentos é o alvo de pesquisa discutido neste trabalho.
- No Capítulo 7, apresentamos a atualização dos coeficientes de Lagrange vizinhos como forma de diminuir o tempo de treinamento do SVM. É apresentada neste capítulo a diminuição no tempo de treinamento bem como a manutenção da capacidade de generalização do classificador através do teste de hipóteses (detalhes no Apêndice F)
- No Capítulo 8, uma nova estratégia de aprendizado visando diminuição no tempo de treinamento é apresentada. Subdividindo um conjunto em partes menores, mas ordenadas segundo a dimensão de maior entropia, conseguimos um treinamento simples e não "repetitivo" dos SVM.
- No Capítulo 9 finaliza-se esta tese discutindo os resultados obtidos, as limitações e os desafios ainda existentes e que podem ser explorados como trabalhos futuros.

# 2. *Support Vector Machine* – SVM

Neste capítulo iremos revisar os fundamentos teóricos envolvidos na teoria de aprendizado das máquinas de vetores-suporte. Deste ponto em diante as Máquinas de Vetores de Suporte serão muitas vezes referenciadas pela sua sigla, SVM.

A revisão da teoria do SVM é importante porque serve de pretexto para o algoritmo de aprendizado que foi modificado, o SMO, que significa Otimização Mínima Sequencial, ou *Sequential Minimal Optimization*.

Outro tema que é abordado neste capítulo é o aprendizado de máquina. Ele é útil quando temos problemas que não são facilmente programáveis usando equações matemáticas e regras determinísticas. As redes neurais artificiais – RNAs (*Artificial Neural Network* - ANN) e o SVM apresentam equações matemáticas genéricas, ou seja, apresentam funções que não são programadas especificamente para um único problema.

As RNAs e os SVMs apresentam não apenas formulações matemáticas, mas também um conjunto de coeficientes e pesos que são ajustados por meio de um processo de treinamento; em outras palavras, realizam o aprendizado a partir de exemplos, visando a diminuição do erro associado ao treinamento.

O aprendizado, seja supervisionado ou não, é guiado pelas amostras de treinamento que disponibilizamos a esses sistemas. Assim, para todos os sistemas que codificam o aprendizado de máquina, as seguintes variáveis são definidas:

- Amostras de entrada **x**, geralmente devem ser na forma vetorial. Essas amostras são geradas independente e identicamente distribuídas segundo uma função distribuição de probabilidade desconhecida P(**x**). $\mathbf{x} \in \Re^n$ ;
- Identificação da amostra por y, é a classe à qual **x** pertence;
- $f(\mathbf{x})$ é a função que realiza o mapeamento do domínio das entradas para o domínio das saídas; o conjunto de funções $f(\mathbf{x})$ também é chamado de hipóteses;
- As amostras **x** serão identificadas pelo índice t; quando estivermos referenciando duas amostras **x** em especial, usaremos os índices i e j, ou seja, $i, j \in t = 1,2,...,N$ .

Em geral, empregamos o aprendizado de máquina para classificação de amostras. Classificação binária é quando temos duas classes. A multiclassificação é uma classificação finita de K classes com K>2. No Apêndice A encontram-se informações acerca da teoria fundamental do SVM.

## 2.1 Introdução

Iniciaremos a explicação do SVM partindo da sua forma original, ou seja, classificação de duas classes. Nesse caso, temos o seguinte conjunto de treinamento:

$$S = \left\{ (\mathbf{x_1}, y_1) \cdots (\mathbf{x_N}, y_N); \quad \mathbf{x_t} \in \Re^n, y \in \{-1,+1\} \right\} \qquad \textbf{Eq. 2.1}$$

Na Equação 2.1, acima, S é um conjunto de pares formados por N amostras **x** (n-dimensional) de treinamento e suas respectivas identificações ou classificações, y (*target*). As classes apresentadas são duas, a positiva e a negativa.

SVM é uma MA que discrimina um conjunto de amostras de teste dividindo-o em dois grupos (é um classificador binário em sua formulação original), e se baseia na teoria de aprendizado estatístico e na teoria de otimização. A teoria matemática de otimização tem sido usada em várias áreas da ciência (economia, mecânica, física entre outras); por exemplo, em mecânica (HAFTKA, 1993) há o problema de otimização que visa minimizar a massa de uma treliça sujeita a um carregamento com restrições de tensão mecânica em cada barra.

SVM (VAPNIK, 1995) é um classificador linear que, usando um hiperplano, separa um conjunto de amostras positivas de um conjunto de amostras negativas. Geometricamente, esse método pode ser visto como uma tentativa de busca da melhor superfície $\sigma_i$, no conjunto de todas as superfícies $\sigma_1$, $\sigma_2$ ... $\sigma_n$ no espaço r-dimensional que separa os exemplos de treinamento positivos dos negativos (superfície de decisão). $\sigma_i$ separa os exemplos positivos dos negativos, buscando maximizar a distância entre as superfícies limítrofes positivas e negativas.

O *Support Vector Machine* é um classificador linear cuja função *f* discriminante geral é da forma:

$$f(\mathbf{x}, \mathbf{w}, b) = sign(\mathbf{w} \cdot \mathbf{x} + b) \qquad \textbf{Eq. 2.2}$$

Onde, *sign*(q) =+1 se q ≥ 0 e *sign*(q) =-1 se q < 0; b é o *bias*, esse parâmetro permite que o hiperplano separador se posicione no local correto para separação das amostras.

Usando-se da Equação 2.2, separam-se dois espaços, cada um deles identificado como +1 (círculos na parte superior) ou –1 (quadrados na parte inferior), como na Figura 2.1. Essa figura contém uma ilustração de um SVM linear, que mostra a separação entre classes de elementos com certo grau de ruído (um quadrado invadindo a área dos círculos).

Existem diversas retas (imagine a mesma reta da Figura 2.1 girando levemente em torno do seu centro, veja a Figura 2.2) que poderiam estar no lugar daquela apresentada na Figura 2.1; assim, o que o SVM visa é encontrar a maior margem de separação entre as duas classes, de forma a garantir uma boa separação entre elas.

Os vetores-suporte são as amostras limítrofes da margem e servem como fronteira na sua maximização. Queremos que o SVM tenha uma boa capacidade de generalização, assim teremos que escolher a melhor função discriminadora dentre todas as possíveis.

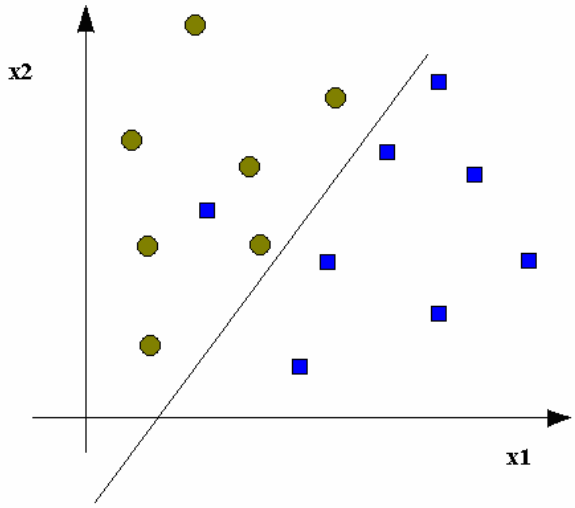



**Figura 2.1 *Support vector machine*, ilustração de um classificador binário. As amostras positivas estão representadas por círculos, enquanto que as negativas, por quadrados. Os eixos ordenados representam as dimensões de cada amostra.**

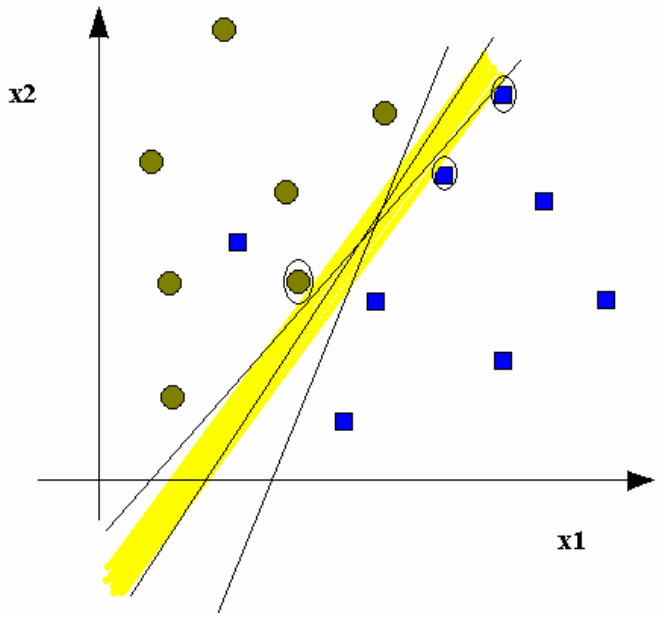



**Figura 2.2. Vetores-suporte demarcando as linhas de separação entre duas classes. Os vetores-suporte são as amostras que ficam na fronteira de separação das classes. Eles estão destacados na figura com um círculo ao seu redor.**

A formulação do SVM resulta em um problema de otimização quadrática e envolve o princípio de minimização do risco estrutural (SRM, do inglês *structural risk minimization*), em vez do princípio de minimização do risco empírico (ERM, do inglês *empirical risk minimization*), muito empregado pelas redes neurais artificiais.

O erro associado ao aprendizado de máquina durante a sua fase de treinamento é um valor que reflete um subconjunto (o conjunto de treinamento) de todo um domínio (conjunto de treino e teste). Dizer que um classificador baseado em ERM atingiu uma taxa de erro de 0,01 diz muito pouco da capacidade da máquina de aprendizado (MA). Assim, o SRM minimiza o erro de generalização, diferentemente do ERM, que visa minimizar o erro associado às amostras de treinamento. Em vez de construir uma função de decisão *f* por meio da minimização do erro de treinamento (ERM), o SRM visa construir *f* que minimize o limite superior do teste de erro dado por:

$$R(z) \leq R^{emp}(z) + \kappa(f, h, \nu) \qquad \textbf{Eq. 2.3}$$

Onde ν pertence ao intervalo [0,1] é um parâmetro de ajuste, $R^{emp}(z)$ é o risco empírico e $\kappa(f,h,\nu)$ é o termo de confiança; h é a dimensão VC (Vapnik e Chervonenkis) (VAPNIK, 1995)

das funções de decisão candidatas; z é a função hipótese associada ao aprendizado supervisionado.

O parâmetro z é dado por: $z = (\mathbf{w}, b) \in \Re^{n+1}$

O parâmetro h é uma medida de noção de capacidade, ou seja, é a habilidade de uma MA para aprender qualquer conjunto de treino sem erro. É uma medida de flexibilidade, a qual controla o risco desejado. Muita capacidade implica no *overfitting* e pouca capacidade implica em muito erro empírico.

A dimensão VC de um conjunto de funções (VAPNIK, 1995) diz respeito à quantidade máxima de amostras (no contexto de um classificador) que um SVM consegue separar (*shatter*). Assim, para uma classe de funções lineares num espaço bidimensional $\Re^2$ a dimensão VC é 3, ou seja, uma reta consegue separar no máximo 3 pontos no plano. A dimensão VC diz respeito à capacidade de classificação de umaclasse de funções.
A capacidade de generalização de uma SVM está ligada diretamente às funções-hipóteses selecionadas. Essas funções-hipóteses podem ser quantificadas pela Função de Risco dada por *R(z)*.

$$R(z) = \int \frac{1}{2} |y - f(\mathbf{x}, z)| \quad dP(\mathbf{x}, y) \qquad \textbf{Eq. 2.4}$$

Onde, $\frac{1}{2}|y - f(\mathbf{x}, z)|$, conhecida como função perda, é a diferença entre a saída esperada e a saída obtida e P(**x**,y) é uma função distribuição desconhecida das amostras **x**.

O que queremos é um $z^*$ tal que minimize o Risco, no entanto, a distribuição P(**x**,y) é desconhecida. Assim, teremos que usar o conjunto S e dados empíricos para estabelecer um limite máximo de Risco, tal como apresentado na Equação 2.4.

Note-se, abaixo, que no Risco Empírico não temos o envolvimento da função distribuição de probabilidades:

$$R_{emp}(z) = \frac{1}{2N}\sum_{t=1}^{N}|y_t - f(\mathbf{x_t}, z)| \qquad \textbf{Eq. 2.5}$$

Uma vez estabelecido o limite superior da função risco, queremos que o classificador linear encontre a melhor reta que separe as duas classes.

Percebe-se, na Figura 2.2, que existem várias possibilidades para separar linearmente duas classes, o objetivo do SVM é, pois, trabalhar com margem de separação entre as duas classes em vez de trabalhar com apenas uma superfície ou hiperplano, ou seja, implementar o princípio de minimização do risco estrutural.

A maximização da margem M de separação entre as superfícies positivas e negativas está ilustrada na Figura 2.3.

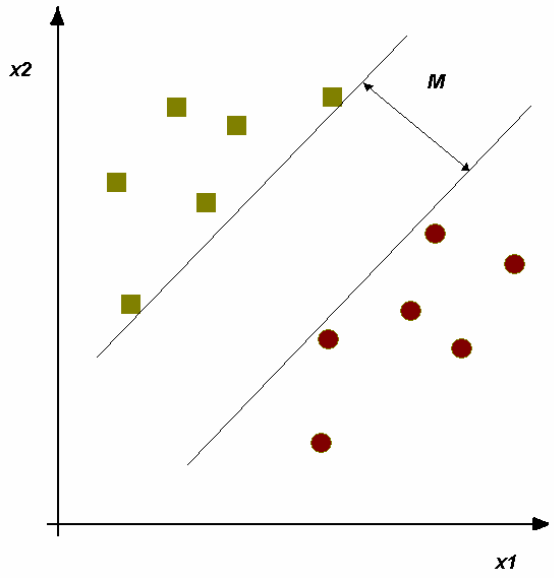



**Figura 2.3. Margem de separação entre hiperplanos nas fronteiras das classes positiva (quadrado preenchido) e negativa (círculo preenchido). No SVM queremos que a margem M seja maximizada. Os eixos ordenados x1 e x2 representam as dimensões da amostras no espaço 2D**

Dado que (no espaço 2D):

- O semi-plano positivo é separado pela reta, $\{\mathbf{xp} : \mathbf{w}^{\mathbf{T}} \cdot \mathbf{xp} + b = +1\}$
- O semi-plano negativo é separado pela reta, $\{\mathbf{xn} : \mathbf{w}^{\mathbf{T}} \cdot \mathbf{xn} + b = -1\}$
- $\mathbf{xp} = \mathbf{xn} + \lambda \mathbf{w}$

Os parâmetros **w** e b são, respectivamente, o peso e o *bias*. Eles serão ajustados durante o processo de treinamento do SVM. **xp** representa as amostras da classe positiva e **xn** representa as amostras da classe negativa.

Pode-se calcular a margem M em função de ***w*** e b, como a seguir.

$$M = |\mathbf{xp} - \mathbf{xn}|$$
$$M = |\lambda \mathbf{w}|$$
$$M = \lambda \sqrt{\mathrm{w}^{\mathrm{T}} \cdot \mathrm{w}} \qquad \textbf{Eq. 2.6}$$

Mas,

$$\mathbf{w} \cdot (\mathbf{xn} + \lambda \mathbf{w}) + b = 1$$
$$\mathbf{w} \cdot \mathbf{xn} + b + \lambda \mathbf{w}^{\mathbf{T}} \cdot \mathbf{w} = 1 \qquad \textbf{Eq. 2.7}$$
$$\lambda \mathbf{w}^{\mathbf{T}} \cdot \mathbf{w} = 2$$
$$\lambda = \frac{2}{\mathbf{w}^{\mathbf{T}} \cdot \mathbf{w}}$$

Dessa forma, tem-se:

$$M = \frac{2\sqrt{\mathrm{w}^{\mathrm{T}} \cdot \mathrm{w}}}{\mathrm{w}^{\mathrm{T}} \cdot \mathrm{w}} = \frac{2}{\sqrt{\mathrm{w}^{\mathrm{T}} \cdot \mathrm{w}}} \qquad \textbf{Eq. 2.8}$$

Resolver um problema usando o SVM resume-se, então, a:

minimizar uma função objetiva

$$f : f = \frac{1}{2}\left\|\mathbf{w}\right\|^2 \qquad \textbf{Eq. 2.9}$$

com **n** restrições da forma

$$\mathrm{w}^{\mathrm{T}} \cdot \mathrm{x} + b \geq +1, \qquad se \quad y = +1$$
$$\mathrm{w}^{\mathrm{T}} \cdot \mathrm{x} + b \leq -1, \qquad se \quad y = -1 \qquad \textbf{Eq. 2.10}$$

Onde **n** representa o espaço amostral do problema, e a função *f* é uma função quadrática.

Problemas como o acima apresentado são desafios de otimização, como visto no começo deste capítulo. Temos em particular um problema de otimização quadrática na Equação 2.9, com restrições lineares (Equação 2.10). Pode-se resolvê-lo por diversas técnicas, tais como:

- Gradiente descente
- *Simulated annealing*
- Inversão de matriz
- *Expectation maximization* (EM)
- Método de Newton
- Otimização mínima sequencial

Existem situações em que as amostras a serem separadas pelo SVM não estão bem "comportadas" como ilustra a Figura 2.4. Existirão situações nas quais um classificador linear como apresentado não encontrará alguma maneira de separar as amostras. Nesse caso é necessário usar um classificador com função de separação não linear. O quadrado entre os círculos representa uma amostra com características diferentes do grupo ao qual elas pertencem; essas amostras são tratadas, dentro do possível, com parâmetros de ajuste no treinamento do SVM. A seção seguinte trata dessas situações.

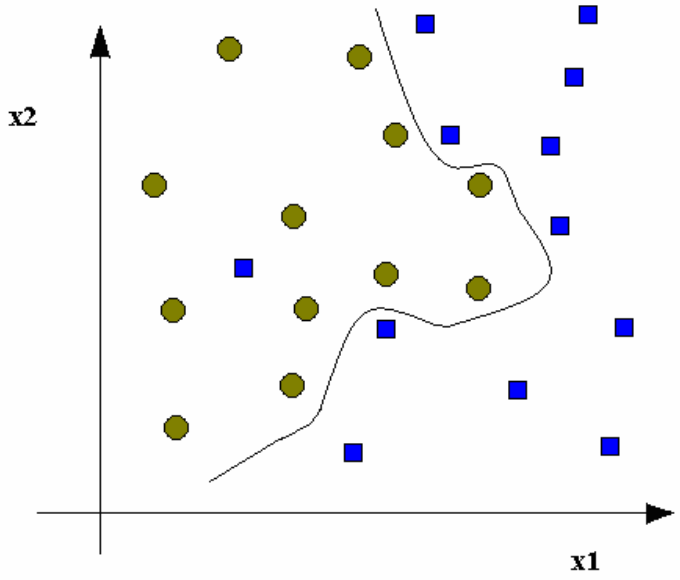



**Figura 2.4. Exemplo de amostras distribuídas de tal forma que um classificador linear não é capaz de separá-las.**

## 2.2 Função kernel

Um SVM linear poderá não ser capaz de separar as amostras de um domínio usando um semi-plano. Nessas situações pode-se trabalhar em um novo domínio para o qual as amostras de

entrada serão mapeadas. Esse novo domínio chamado de espaço das características permite o uso de um classificador linear para separar as amostras.

A Equação 2.9 apresentada na seção anterior está usando uma função *kernel* linear. Uma função *kernel* nada mais é do que um produto escalar entre dois vetores. Por intermédio dela é possível trabalhar no espaço das transformadas onde se consegue realizar uma separação linear entre amostras das duas classes.

A Figura 2.5 ilustra o conceito de mapeamento das amostras do espaço das entradas para o espaço das transformadas. A figura à esquerda na Figura 2.5 apresenta amostras de duas classes não separáveis por uma reta; já essas mesmas amostras no espaço das transformadas são separáveis linearmente por um hiperplano (figura à direita).

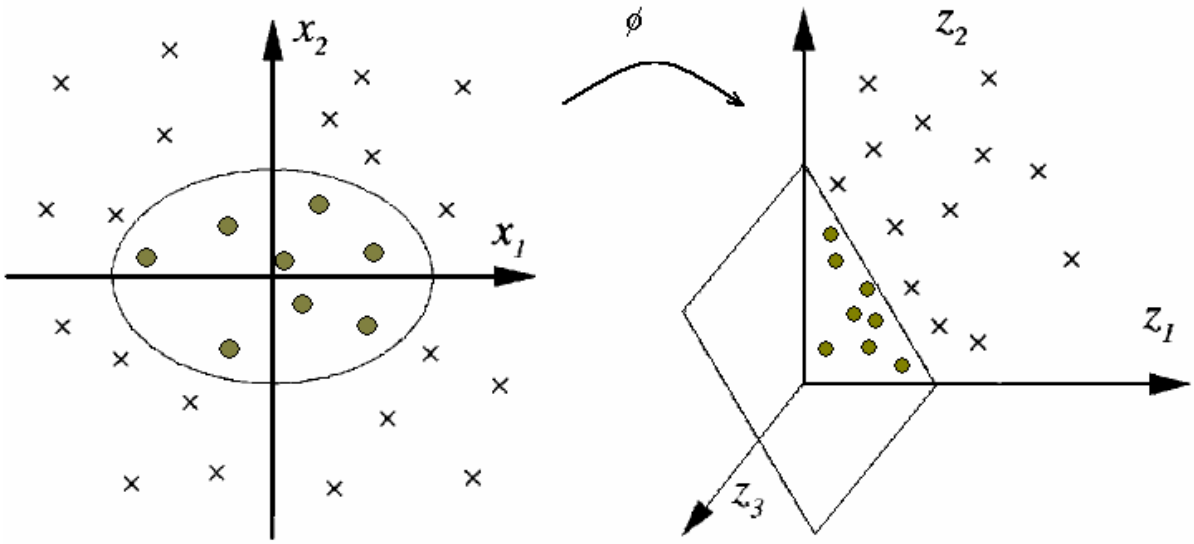



**Figura 2.5. Exemplo de aplicação da função *kernel*. Amostras inicialmente não separáveis linearmente podem ser separadas no espaço das transformadas. Adaptado de (SCHOLKOPF; SMOLA, 2002)**

A seguir listamos algumas funções *kernel* frequentemente utilizadas na literatura (SCHOLKOPF; SMOLA, 2002).

## 2.3 Exemplos de funções *kernel*

As funções *kernel* mais usadas estão apresentadas a seguir. A função *kernel* linear (Equação 2.11) é usada no SVM linear onde as classes são linearmente separáveis. Já a função *kernel*

polinomial e a RBF, respectivamente Equações 2.12 e 2.13, são utilizadas quando não se consegue uma separação linear das amostras.

***Kernel* linear**

$$K(\mathbf{x},\mathbf{x}') = \left(\mathbf{x}^T \cdot \mathbf{x}\right) \qquad \textbf{Eq. 2.11}$$

***Kernel* polinomial**

$$K(\mathbf{x},\mathbf{x}') = \left(1 + \left(\mathbf{x}^T \cdot \mathbf{x}\right)\right)^p \qquad \textbf{Eq. 2.12}$$

Na Equação 2.12, p indica o grau da função polinomial.

***Kernel* de base gaussiana radial, *radial basis function* - RBF**

$$K(\mathbf{x},\mathbf{x}') = \exp\left(-\frac{1}{2\gamma}\left\|\mathbf{x}-\mathbf{x}'\right\|^2\right) \qquad \textbf{Eq. 2.13}$$

Na Equação 2.13, γ (gama) indica a largura da função de base radial gaussiana.

A seguir, será apresentado um algoritmo de otimização que implementa o SVM. Esse algoritmo permite trabalhar com problemas de otimização sem a necessidade de trabalhar com matriz das funções *kernel*, uma das partes mais custosas (complexidade computacional) do SVM.

## 2.4 Otimização sequencial mínima

A otimização sequencial mínima (do inglês, *sequential minimal optimization* - SMO) (PLATT, 1998) é uma forma analítica de resolver o problema de otimização da programação quadrática apresentada anteriormente. A SMO é um caso particular de um método chamado ***chunking***, no qual se analisam, apenas dois multiplicadores Lagrangeanos por vez.

Existem várias versões do SVM, muitas delas trabalham na sua otimização tratando a atualização da inversa da matriz *kernel* de diversas formas. Uma das formas mais eficientes e menos custosa para se trabalhar com a atualização sucessiva da inversa de uma matriz é pelo uso da fórmula Sherman-Morrison-Woodbury (veja Apêndice C). Os autores mostram como atualizar a inversa de uma matriz de forma incremental.

No entanto, esse mesmo método de atualização da inversa de uma matriz sofre o mesmo problema das demais técnicas de atualização da inversa de uma matriz, o seu custo computacional na medida em que a matriz aumenta.

Conforme apresentado na seção anterior, a maximização da margem M que separa os hiperplanos positivo e negativo se dá pela resolução de um problema dual de programação quadrática expresso por (VAPNIK, 1995):

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 \qquad \textbf{Eq. 2.14}$$

Sujeito a:

$$y_t \cdot \left(\mathbf{w^T} \cdot \mathbf{x}_t + b\right) \geq 1, \forall t \qquad \textbf{Eq. 2.15}$$

Assim, o problema da programação quadrática pode ser expresso, usando-se os Lagrangeanos, por:

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\langle \mathbf{w} \cdot \mathbf{w} \rangle - \sum_{t=1}^{N} \alpha_t \left[ y_t \cdot \left( \langle \mathbf{w} \cdot \mathbf{x_t} \rangle + b \right) - 1 \right] \qquad \textbf{Eq. 2.16}$$

Onde $\alpha$ é o coeficiente de Lagrange associado a cada amostra x. O Lagrangeano L deve ser minimizado em função das suas variáveis primárias **w** e b. Mas sabemos que no ponto de cela a derivada com relação às variáveis primárias deve ser zero; assim temos a seguinte equação:

$$\begin{aligned} \mathbf{w} &= \sum_{t=1}^{N} y_t \alpha_t \mathbf{x}_t \\ \sum_{t=1}^{N} y_t \alpha_t &= 0 \end{aligned} \qquad \textbf{Eq. 2.17}$$

Das Equações 2.16 e 2.17 temos a seguinte equação:

$$\begin{aligned} L(\mathbf{w}, b, \alpha) &= \frac{1}{2}\langle \mathbf{w} \cdot \mathbf{w} \rangle - \sum_{i=1}^{N} \alpha_i \left[ y_i \cdot \left( \langle \mathbf{w} \cdot \mathbf{x_i} \rangle + b \right) - 1 \right] \\ &= \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \langle \mathbf{x_i} \cdot \mathbf{x_j} \rangle - \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j + \sum_{i=1}^{N} \alpha_i \\ &= \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \langle \mathbf{x_i} \cdot \mathbf{x_j} \rangle = W(\alpha) \end{aligned} \qquad \textbf{Eq. 2.18}$$

A Equação 2.18 está em função de α apenas. O problema pode, então, ser resolvido no dual conforme mostra a Equação 2.19.

$$\max_{\alpha} W(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} y_i y_j < \mathbf{x_i} \cdot \mathbf{x_j} > \alpha_i \alpha_j \qquad \textbf{Eq. 2.19}$$

A função *W* é a função dual da função Lagrangeano *L*.

Sujeito a uma restrição de desigualdade,

$$\alpha_t \geq 0, \qquad \forall t$$

e a uma restrição de igualdade,

$$\sum_{t=1}^{N} y_t \alpha_t = 0 \qquad \textbf{Eq. 2.20}$$

$$0 \leq \alpha_t \leq C, \forall \quad 1 \leq t \leq N \qquad \textbf{Eq. 2.21}$$
(restrição do quadrado)

Onde α são os multiplicadores de Lagrange e *W*, a função objetiva.

Existe uma relação direta entre os multiplicadores de Lagrange e as restrições. Assim, uma vez determinados os multiplicadores de Lagrange pode-se determinar o vetor **w** e o coeficiente (também chamado de limiar) b.

$$\mathbf{w} = \sum_{t=1}^{N} y_t \alpha_t \mathbf{x}_t$$

$$b = -\mathbf{w}^{\mathbf{T}} \cdot \mathbf{x}_t + y_t \quad \text{Para algum } \alpha_t > 0 \qquad \textbf{Eq. 2.22}$$

### 2.4.1 O parâmetro de regularização

Nem sempre se consegue uma separação perfeita (veja na Figura 2.1 e 2.2 a presença de um elemento quadrado na classe dos círculos) dos dois planos. Para tratar essas situações "ruidosas", foi proposta em 1995, por Cortes e Vapnik (CORTES; VAPNIK, 1995) uma modificação da Equação de otimização, adicionando-se uma variável de correção $\xi_t$ e uma

constante C de ajuste para obter uma margem grande, mas, ao mesmo tempo, com baixo erro de classificação.

Dessa maneira, a nova função objetiva será:

$$\min_{\mathbf{w},b,\xi} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{t=1}^{N}\xi_t^k \qquad \textbf{Eq. 2.23}$$

Onde k=1 para 1-norm SVM e k=2 para 2-norm SVM.

Sujeito a:

$$y_t(\mathbf{w}^{\mathbf{T}} \cdot \mathbf{x}_t + b) \geq 1 - \xi_t, \forall_t \qquad \textbf{Eq. 2.24}$$
$$\xi_t \geq 0, \forall t$$

O Lagrangeano da função objetiva será:

$$W(\alpha) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{t=1}^{N}\xi_t - \sum_{t=1}^{N}\alpha_t(y_t \cdot (\mathbf{w}^{\mathbf{T}} \cdot \mathbf{x}_{\mathbf{t}} + b) - 1 + \xi_t) \qquad \textbf{Eq. 2.25}$$

A restrição de desigualdade para a nova função objetiva será:

$$0 \leq \alpha_t \leq C, \forall t$$

Assim como no caso das SVM de Vapnik, o SVM de Platt (chamado de SMO) pode utilizar funções *kernel* (que obedecem às condições de Mercer (BURGES, 1998), resultando em um problema QP positivo definido) para uma classificação não-linear. Para tal, será necessário substituir o produto vetorial por uma função *kernel* $K(\mathbf{x}_i, \mathbf{x}_j)$.

A condição Karush-Kuhn-Tucker (KKT) para o problema QP acima é uma condição necessária e suficiente (devido à convexidade da função objetiva) e é dada por:

$$\alpha_t\left(y_t \cdot \left(\mathbf{w}^{\mathbf{T}} \cdot \mathbf{x}_t + b\right) - 1 + \xi_t\right) = 0 \qquad \textbf{Eq. 2.26}$$

$$(C - \alpha_t)\xi_t = 0 \qquad \textbf{Eq. 2.27}$$

que podem ser reescritos da seguinte forma:

$$\alpha_t(y_t u_t - 1 + \xi_t) = 0 \qquad \textbf{Eq. 2.28}$$

$$(C - \alpha_t)\xi_t = 0 \qquad \textbf{Eq. 2.29}$$

Portanto, tem-se a seguinte propriedade:

$$\begin{aligned} &\alpha_t = 0 \Rightarrow y_t u_t \geq 1 \\ &0 < \alpha_t < C \Rightarrow \xi_t = 0 \Rightarrow y_t u_t = 1 \\ &\alpha_t = C \Rightarrow y_t u_t \leq 1 \end{aligned} \qquad \textbf{Eq. 2.30}$$

onde $u_t$ é a saída do SVM para o t-ésimo elemento de treino.

SMO trabalha com dois α ($\alpha_1$ e $\alpha_2$) em cada passo de otimização: dessa maneira, pode-se tratar o problema de forma algébrica ao invés de usar cálculos numéricos (PLATT, 1998).

A segunda derivada de *W* com relação ao $\alpha_2$ é dada por:

$$\frac{\partial^2 W}{\partial \alpha_2^2} = 2k_{12} - k_{11} - k_{22} = \eta \qquad \textbf{Eq. 2.31}$$

onde $k_{i,j}$ representa a função *kernel* $K(x_i,x_j)$ e η é o passo de ajuste descrito acima.
Define-se também $E_i$ como o erro na i-ésima amostra de treino, ou seja,

$$E_i = u_i - y_i \qquad \textbf{Eq. 2.32}$$

O multiplicador Lagrangeano pode, por consequência, ser expresso por:

$$\alpha_2^{novo} = \alpha_2^{antigo} - \frac{y_2 (E_1 - E_2)}{\eta} \qquad \textbf{Eq. 2.33}$$

A escolha dos dois multiplicadores de Lagrange é feita selecionando-se uma amostra do conjunto de treinamento que viole a condição KKT. Uma vez encontrada tal amostra, uma segunda é selecionada heuristicamente para então os dois multiplicadores de Lagrange serem otimizados em conjunto.

Ao final da otimização, o SVM é atualizado e o processo se repete para uma nova amostra que viole a condição KKT para que, ao final, toda amostra de treino seja avaliada, ou seja, todas as amostras devem satisfazer à condição KKT.

A heurística usada para a escolha do segundo α é a maximização do passo de otimização, dada pela Equação 2.31.

## 2.5 Extensão do SMO de Platt

O algoritmo SMO é um algoritmo de simples codificação, mas também apresenta boas propriedades como velocidade de execução. Uma das características do SMO original de Platt é a atualização de um único limiar (parâmetro b nas Equações 2.24 e 2.35). Muitos autores (KEERTHI et al., 2001) apontam esse fato como um ponto negativo do SMO.

Em (KEERTHI et al., 2001) os autores descrevem o problema do SMO da seguinte forma, o parâmetro β (que é o mesmo de b na condição ótima) não precisa ser atualizado durante as iterações; no entanto, como ele é usado nos "estágios" intermediários, precisamos calcular o β ao final de cada atualização dos Lagrangeanos.

Contudo, esse valor de limiar pode não ser o ideal já que ainda não atingimos a condição ótima; assim, a cada iteração podemos estar introduzindo um erro de aproximação. Keerthi et al. (KEERTHI et al., 2001), então, buscaram uma forma desse parâmetro não ser dependente das atualizações de dois Lagrangeanos por iteração. A proposta desses autores é usar a relação entre o problema de otimização primal e dual, ou seja, o *gap* da dualidade.

Assim, (KEERTHI et al., 2001) introduziram um sub-problema a resolver, o Lagrangeano do dual da função objetiva é descrito por:

$$L = \frac{1}{2}\mathbf{w}(\alpha)\cdot\mathbf{w}(\alpha) - \sum_t \alpha_t - \sum_t \delta_t\alpha_t + \sum_t \mu_t(\alpha_t - C) - \beta\sum_t \alpha_t y_t \qquad \textbf{Eq. 2.34}$$

$t = 1,2,3,\cdots,N$ e $N \in \Re$

Onde se define:

$$F_i = \langle \mathbf{w}\cdot\mathbf{x}_i \rangle - y_i = \sum_j \alpha_j y_j k(\mathbf{x}_i, \mathbf{x}_j) - y_i \qquad \textbf{Eq. 2.35}$$

$i, j \in t$

As condições KKT para o problema dual são:

$$\frac{\partial L}{\partial \alpha} = (F_t - \beta)y_t - \delta_t + \mu_t = 0,\ \delta_t \geq 0,\ \delta_t\alpha_t = 0,\ \mu_t \geq 0,\ \mu_t(\alpha_t - C) = 0,\ \forall t \qquad \textbf{Eq. 2.36}$$

As condições KKT a serem satisfeitas podem ser descritas da seguinte forma:

**Se** $\alpha_t$=0,

$$\delta_t \geq 0 \text{ e } \mu_t = 0 \Rightarrow y_t(F_t - \beta) \geq 0 \qquad \textbf{Eq. 2.37}$$

**Se** $0 < \alpha_t < C$,

$$\delta_t = 0 \text{ e } \mu_t = 0 \Rightarrow y_t(F_t - \beta) \approx 0 \qquad \textbf{Eq. 2.38}$$

**Se** $\alpha_t = C$,

$$\delta_t = 0 \text{ e } \mu_t \geq 0 \Rightarrow y_t(F_t - \beta) \leq 0 \qquad \textbf{Eq. 2.39}$$

Assim, podemos juntar as equações acima da seguinte forma:

$$\begin{aligned} t \in I_0 \cup I_1 \cup I_2 &\Rightarrow F_t \geq \beta \\ t \in I_0 \cup I_3 \cup I_4 &\Rightarrow F_t \leq \beta \end{aligned} \qquad \textbf{Eq. 2.40}$$

Onde:

$$\begin{aligned} I_0 &\equiv \{t : 0 < \alpha_t < C\} \\ I_1 &\equiv \{t : y_t = +1, \alpha_t = 0\} \\ I_2 &\equiv \{t : y_t = -1, \alpha_t = C\} \\ I_3 &\equiv \{t : y_t = +1, \alpha_t = C\} \\ I_4 &\equiv \{t : y_t = -1, \alpha_t = 0\} \end{aligned} \qquad \textbf{Eq. 2.41}$$

Os subconjuntos acima podem ser agrupados da seguinte forma:

$$\begin{aligned} I_{up} &= \{t \mid \alpha_t < C, \quad y_t = 1 \quad ou \quad \alpha_t > 0, \quad y_t = -1\} \\ I_{low} &= \{t \mid \alpha_t < C, \quad y_t = -1 \quad ou \quad \alpha_t > 0, \quad y_t = 1\} \end{aligned} \qquad \textbf{Eq. 2.42}$$

$I_{up}$ contém os índices dos $\alpha_t$ que podem ser incrementados sem a violação imediata das restrições quadradas (*box constraints*), veja a Equação 2.21. E, de forma análoga, os coeficientes cujos índices estão em $I_{low}$ podem ter seus valores decrescidos.

A restrição de igualdade apresentada na Equação 2.20 garante que o incremento de um coeficiente e o decremento de outro, mantêm viável a solução do problema. Assim é

fundamental a escolha de um elemento de cada conjunto a cada iteração para garantir que a solução do sistema ainda seja viável.

E, para validar as condições KKT, devemos observar a seguinte equação:

$$b_{up} = \min\{F_t : t \in I_0 \cup I_1 \cup I_2\} = F_{t_up}$$
$$b_{low} = \max\{F_t : t \in I_0 \cup I_3 \cup I_4\} = F_{t_low}$$

Assim, as condições KT implicam em $b_{up} \geq b_{low}$, ou seja, $\forall t \in I_0 \cup I_1 \cup I_2, F_t \geq b_{low}$ e $\forall t \in I_0 \cup I_3 \cup I_4, F_t \leq b_{up}$. Note-se que nesta comparação o limiar b não foi usado.

Ou seja, o algoritmo tem como condição de parada quando $b_{up} \leq b_{low} - \varepsilon$, onde ε é um parâmetro de tolerância ajustado empiricamente.

Abaixo apresentamos o pseudocódigo resumido do algoritmo de Keerthi.

**Pseudocódigo da seleção de dois pares de coeficientes com a heurística de Keerthi (KEERTHI, 2001)**

1. O conjunto de exemplos é dividido em dois grupos $I_{up}$ e $I_{low}$
   $I_{up} = \{t \mid \alpha_t < C, \quad y_t = 1 \quad or \quad \alpha_t > 0, \quad y_t = -1\}$
   $I_{low} = \{t \mid \alpha_t < C, \quad y_t = -1 \quad or \quad \alpha_t > 0, \quad y_t = 1\}$
2. O conjunto de trabalho ($i_{11}$,$i_{12}$) é selecionado.
   $i_{11} = \arg \max\{F_i\}, i \in I_{up}$
   $i_{12} = \arg \min\{F_i\}, i \in I_{low}$
3. Os coeficientes $b_{up}$ e $b_{low}$ (limiares) são determinados.
   $b_{up}=F_{11}$
   $b_{low}=F_{12}$
4. A condição de parada do SMO é testada.
   IF $b_{up} > b_{low} - \varepsilon$
   THEN return ($\alpha_{11}$, $\alpha_{12}$)
   ELSE END SMO
   END

Até agora temos apresentado o SVM binário, na seção seguinte apresentaremos o SMO de uma-classe, uma variação do SMO binário para identificação de novidades.

## 2.6 SMO – Classificação de uma-classe

$$\min_{w\in F,\xi\in R^l,\rho\in R} \frac{1}{2}\|\mathbf{w}\|^2 + \frac{1}{\nu\cdot N}\sum_{t=1}^{N}\xi_t - \rho \qquad \textbf{Eq. 2.43}$$

cujas restrições são:

$$(\mathbf{w}\cdot\Phi(\mathbf{x}_t)) \geq \rho - \xi_t, \qquad \xi_t \geq 0 \qquad \textbf{Eq. 2.44}$$

Onde, $0<\nu<1$ é um parâmetro de ajuste que visa maximizar a distância da origem e ao mesmo tempo maximizar as amostras contidas pela região criada pelo hiperplano. Esse parâmetro corresponde à taxa de anormalidade esperada para o conjunto de amostras.

$\xi_t$ representam as variáveis de ajuste (*slack variable*) de cada uma das amostras que ultrapassam o hiperplano separador. Esta variável de ajuste permite que algumas amostras estejam além da fronteira definida pelo hiperplano que retém as amostra da classe.

$\Phi$ representa a função de mapeamento do espaço das entradas para o espaço dos atributos.

Após a resolução do problema de otimização, a função de decisão para cada ponto é dada por:

$$f(\mathbf{x}) = \operatorname{sgn}((\mathbf{w}\cdot\Phi(\mathbf{x}))-\rho) \qquad \textbf{Eq. 2.45}$$

O Lagrangeano da função f(x) pode ser escrito em função dos multiplicadores Lagrangeano α e β por:

$$L(\mathbf{w},\xi,\rho,\alpha,\beta) = \frac{1}{2}\|\mathbf{w}\|^2 + \frac{1}{\nu\cdot N}\sum_{t=1}^{N}\xi_t - \rho - \sum_{t=1}^{N}\alpha_t\cdot((\mathbf{w}\cdot\Phi(\mathbf{x}_t))-\rho+\xi_t) - \sum_{t=1}^{N}\beta_t\xi_t \qquad \textbf{Eq. 2.46}$$

Onde $\alpha, \beta \geq 0$

Agora, derivando o Lagrangeano com relação às variáveis primárias, $\mathbf{w}, \xi, \rho$, e igualando a zero, temos (PLATT, 1998):

$$\frac{dL(\mathbf{w}, \xi, \rho, \alpha, \beta)}{d\mathbf{w}} = \mathbf{w} - \sum_{t=1}^{N} \alpha_t \cdot \Phi(\mathbf{x}_t) = 0 \qquad \textbf{Eq. 2.47}$$

$$\frac{dL(\mathbf{w}, \xi, \rho, \alpha, \beta)}{d\xi_t} = \frac{1}{\nu \cdot N} - \alpha_t - \beta_t = 0 \qquad \textbf{Eq. 2.48}$$

$$\frac{dL(\mathbf{w}, \xi, \rho, \alpha, \beta)}{d\rho} = -1 + \sum_{t=1}^{N} \alpha_t = 0 \qquad \textbf{Eq. 2.49}$$

Dado que a função *kernel* é expressa por:

$$k(\mathbf{x_1}, \mathbf{x_2}) = (\Phi(\mathbf{x}_1) \cdot \Phi(\mathbf{x}_2)) \qquad \textbf{Eq. 2.50}$$

Através das Equações 2.47, 2.48, 2.49 e 2.50 obtém-se a formulação dual a seguir:

$$\frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j k(\mathbf{x_i}, \mathbf{x_j}) \qquad \textbf{Eq. 2.51}$$

Com as seguintes restrições:

$$0 \leq \alpha_t \leq \frac{1}{\nu \cdot N} \qquad \textbf{Eq. 2.52}$$

$$\sum_{t=1}^{N} \alpha_t = 1 \qquad \textbf{Eq. 2.53}$$

Podemos reescrever a Equação 2.45 usando a Equação 2.50 da seguinte forma:

$$f(\mathbf{x}) = \text{sgn}\left( \sum_{t=1}^{N} \alpha_t \cdot k(\mathbf{x}_i, \mathbf{x}) - \rho \right) \qquad \textbf{Eq. 2.54}$$

No ponto ótimo, a equação de desigualdade (Equação 2.44) pode ser escrita como igualdade, e assim podemos extrair o valor de ρ

$$\rho = (\mathbf{w} \cdot \Phi(\mathbf{x_i})) = \sum_{j=1}^{N} \alpha_j \cdot k(\mathbf{x_j}, \mathbf{x_i}) \qquad \textbf{Eq. 2.55}$$

SMO quebra um problema de otimização quadrática grande em uma série de problemas de otimizações de tamanho 2. Assim, o objetivo do SMO é expressar de forma analítica o mínimo da Equação 2.51 em função de dois parâmetros.

Suponhamos que esses dois parâmetros tenham índices a e b, assim $\alpha_a$ e $\alpha_b$ são esses dois parâmetros. A função objetiva da Equação 2.51 pode ser reescrita como:

$$L(\alpha_a, \alpha_b) = \frac{1}{2}\alpha_a^2 k_{aa} + \frac{1}{2}\alpha_b^2 k_{bb} + \alpha_a \alpha_b k_{ab} + \sum_{i=a,b} \alpha_i \sum_{j=1.\, j\neq a,b}^{N} \alpha_j k_{ij} + L' \qquad \textbf{Eq. 2.56}$$

Onde L' é uma constante com relação aos termos $\alpha_a$ e $\alpha_b$.

Dado que $s^{antigo} = \alpha_a^{antigo} + \alpha_b^{antigo} = \alpha_a + \alpha_b$, temos que :

$$\alpha_a = -\alpha_b + s^{antigo} \qquad \textbf{Eq. 2.57}$$

Pelas Equações 2.56 e 2.57 temos o Lagrangeano em função de $\alpha_b$. Derivando esse Lagrangeano com relação a $\alpha_b$, ou seja, $\frac{\partial L(\alpha_b)}{\partial \alpha_b} = 0$, obtemos o seu mínimo.

Obtêm-se, desta forma, as seguintes equações:

$$\eta = \frac{1}{k_{aa} + k_{bb} - 2k_{ab}} \qquad \textbf{Eq. 2.58}$$

$$\alpha_b = \alpha_b^* + \eta\left[\bar{f}^*(a) - \bar{f}^*(b)\right] \qquad \textbf{Eq. 2.59}$$

Onde,

$$\bar{f}(\mathbf{x}) = \sum_{t=1}^{N} \alpha_t k(\mathbf{x_t}, \mathbf{x}) \qquad \textbf{Eq. 2.60}$$

Da mesma forma que um SMO tradicional de Platt, devemos estabelecer um limite máximo e mínimo das restrições da caixa (*box constraint*).

Assim, podemos definir as seguintes restrições:

$$L = \max\left( s^{antigo} - \frac{1}{\nu \cdot N}, 0 \right)$$
$$H = \min\left( \frac{1}{\nu \cdot N}, s^{antigo} \right) \qquad \textbf{Eq. 2.61}$$

As condições KKT a serem satisfeitas são:

Caso 1:

$\alpha_i = 0 \Rightarrow \bar{f}(\mathbf{x_i}) > 0$

Caso 2:

$$0 < \alpha_i < \frac{1}{\nu \cdot N} \Rightarrow \bar{f}(\mathbf{x_i}) = 0$$

Caso 3:

$$\alpha_i = \frac{1}{\nu \cdot N} \Rightarrow \bar{f}(\mathbf{x_i}) < 0$$

## 2.7 Exemplos de Aplicação do SVM

Nesta seção, alguns exemplos de aplicação do SVM para classificação serão apresentados.

### 2.7.1 Recuperação de Imagem

Recuperação de imagem baseada em conteúdo (*Content Based Image Retrieval* - CBIR) tem sido desenvolvida com sistemas de classificação binária (ZHANG; LIN; BO ZHANG, 2001) e auto-organizáveis (LAAKSONEN; KOSKELA; OJA, 2002).

Em sistemas auto-organizáveis as imagens são agrupadas segundo sua similaridade. No cálculo da similaridade são usados os atributos extraídos da imagem. Como exemplo de um sistema auto-organizável pode-se citar a rede de Kohonen (KOHONEN, 2001).

Em (LAAKSONEN; KOSKELA; OJA, 2002) foi utilizada uma rede neural artificial do tipo auto-organizável (Self-Organizing Map - SOM) para a recuperação de imagem. Neste método, as imagens são mapeadas do domínio das imagens para o domínio da rede SOM de dimensão 2. Ou seja, as imagens são mapeadas para um reticulado 2D preservando a topologia original.

A rede SOM é uma rede não supervisionada, e a PicSOM apresenta duas características importantes para um CBIR, são elas:
- consulta por imagens usando imagens (*query by pictorial example* - QBPE)
- retroalimentação de relevância (*relevance feedback* – RF)

Em QBPE espera-se que uma pessoa indique uma imagem e o sistema (CBIR) retorne um conjunto de imagens similares. A consulta por uma imagem não é feita usando-se palavras, mas sim, imagens.

Já no caso do RF queremos que o sistema aprenda as nossas preferências após sucessivas consultas e refinos ou seleções. A retroalimentação de relevância destaca os atributos mais importantes atribuindo pesos por meio do processo de aprendizado iterativo.

PicSOM é composto por um conjunto de redes SOM em paralelo para recuperação de imagens relevantes de uma base de imagens. Esse conjunto de SOM foi treinado com diferentes atributos extraídos de uma mesma imagem utilizando-se técnicas diversas; assim, cada SOM torna-se específica para um tipo de atributo.

A similaridade ou diferença entre imagens é algo relativo. Cada pessoa tem seu ponto de vista, que pode ser diferente em relação a outros. Assim, o RF tem a função de mapear esse critério subjetivo intrínseco na percepção de cada um. A ideia básica do PicSOM nesse contexto é que cada SOM represente uma função de similaridade; se uma pessoa indica a semelhança entre duas imagens, a SOM que apresentar esta semelhança será escolhida no processo de personalização da recuperação de imagem.

Já em (ZHANG; LIN; BO ZHANG, 2001) foi utilizada uma SVM para treinar RF. Desta vez foi realizada uma associação entre consultas e imagens. Para destacar características intrínsecas de similaridade tal como realizado no trabalho mencionado anteriormente, associações entre consultas e imagens selecionadas foram utilizadas.

SVM foi, então, treinado para distinguir duas classes de imagens as relevantes e as irrelevantes. Por exemplo, dado que a consulta é “águia”, as imagens que representam uma águia são marcadas como figuras relevantes e aquelas que não são águia recebem uma indicação de irrelevância. SVM, então, é treinado para classificar figuras de uma águia.

Em (CHEN; ZHOU; HUANG, 2001) foi utilizado OCSVM para recuperação de imagem. Foram usadas cinco classes (avião, carro, cavalo, águia e vitrais) de imagens, cada qual contendo 100 amostras.

Apenas 10 amostras de cada classe foram selecionadas para o treinamento. E então, todas as 500 imagens foram utilizadas no teste de classificação. Os resultados obtidos não foram bons; no entanto, um pré-processamento mais eficaz deveria ter sido realizado.

### 2.7.2 Diagnóstico Médico Auxiliado por Computador

Wei e colaboradores do Instituto de Tecnologia de Illinois - Estados Unidos - (WEI; YANG; NISHIKAWA, 2009) utilizaram o SVM para classificação das microcalcificações. Essa classificação e muito importante para diagnosticar câncer de mama, dando, desta forma, aos radiologistas uma segunda opinião nos diagnósticos.

Os agrupamentos das microcalcificações podem ser indicações (antecipada) importantes de câncer de mama. Ela é de difícil analise mesmo para um especialista humano.

O SVM utilizado pelos pesquisadores usou a função *kernel* do tipo RBF com sigma ajustado para o valor de 2.5, e C=100 para as 600 amostras de treino, 300 para cada uma das classes, a positiva e a negativa.

A base de dados utilizada consistia de imagens de mamografia obtidas pelo departamento de radiologia da universidade de Chicago. Com um total de 200 imagens diferentes de 104 casos (46 tumores malignos e 58 tumores benignos), cada imagem apresentava microcalcificações.

## 2.8 Considerações Finais

Neste capítulo vimos a teoria do SVM, uma das implementações mais populares do SVM, a SMO, melhorias do SMO e a versão de uma classe do SMO. O SMO foi usado neste trabalho como o algoritmo a ser otimizado e modificado. Uma das etapas mais custosa no treinamento do SMO é a seleção do conjunto de trabalho.

Nesta tese é proposta uma nova forma de gerenciar os pares de coeficientes que mais violam as condições KKT e que portanto, mais contribuem para a solução final. No Capítulo 7 é apresentado o NSMO. Nele, as amostras vizinhas ao coeficiente selecionado para atualização numa iteração do programa serão candidatos para atualização desde que também violem as condições KKT.

Note que no SVM as amostras vizinhas fazem parte da fronteira. Assim, elas são selecionadas para agilizar o processo de treinamento.

No próximo capítulo serão apresentados conceitos de Aprendizado no âmbito do SVM. Aprendizado baseado em regras é uma técnica tradicional que se opõe às máquinas de aprendizado como SVM e RNA. Idéias interessantes deste tipo de aprendizado foram aproveitadas serviram de motivação para o classificador apresentado nos Capítulos 7 e 8.

# 3. Aprendizado com SVM

Existem diversos tipos de aprendizado no contexto do Aprendizado de Máquina. Em geral, podemos dividir em duas as formas de conduzir uma aprendizagem, são elas:

- Aprendizado Supervisionado
- Aprendizado Não-Supervisionado

Quando se fala em aprendizado supervisionado, temos amostras de treinamento já previamente identificadas ou classificadas (estamos tratando aqui de classificação). Assim, o objetivo é treinar o sistema de tal forma que ele possa identificar padrões.

Muitos modelos de redes neurais artificiais - como a rede de múltiplas camadas, perceptron e a rede neural de base radial (RBF) - são usados no aprendizado supervisionado.

Já no caso do aprendizado não supervisionado, o objetivo é obter alguma informação estruturada das amostras de treinamento que não estão previamente identificadas ou classificadas. Temos como objetivo entender padrões que não são facilmente observáveis e, portanto, não identificáveis.

É comum o uso da análise de agrupamento em aprendizados não supervisionados. Como alguns exemplos de algoritmos, podemos citar a rede de Kohonen - *Self-Organizing Map* - (KOHONEN, 2000, 2001) e as análises K-médias.

Nesta tese, trabalharemos com o tipo de aprendizado supervisionado, no qual há uma identificação prévia das amostras utilizadas no treinamento e no teste. Ainda sobre o aprendizado, existe outro tipo de aprendizado que não é baseado em aprendizado de máquina. Neste caso, o aprendizado se realiza por meio da determinação de um conjunto de regras *booleanas* ou regras.

A Seção 3.1 apresenta o aprendizado baseado em regras, muito popular por mostrar claramente a formulação da função $f$ que classifica as amostras, algo que não é visível nos pesos ajustados após o treinamento de uma rede neural artificial ou de uma SVM.

Um problema enfrentado por qualquer método de aprendizado é o volume de dados a tratar, tal como foi discutido no Capítulo 1. O aprendizado incremental tem por finalidade trabalhar com um pequeno subconjunto de amostras de treinamento, que se renova a cada iteração ou ciclo de aprendizado do sistema, até que todas as amostras tenham sido utilizadas. A Seção 3.2 apresenta o aprendizado incremental.

## 3.1 Aprendizado baseado em regras

Aprendizado baseado em regras de associação tem sido empregado como alternativa ao aprendizado de máquina. Neste modelo de aprendizado, muito utilizado na classificação, fornecemos sucessivamente conjuntos de regras (são literais na forma normal conjuntiva) para manter apenas aquelas que apresentarem melhores resultados de classificação.

Esta indução de regras por métodos de abrangência sequencial (ou sucessiva) tem sido utilizada em algoritmos como Foil (QUINLAN; CAMERON-JONES, 1993), CPAR (YIN; HAN, 2003) e CMAR (LI; HAN; PEI, 2001). O objetivo é encontrar regras que classifiquem conjuntos de amostras de uma classe e não classifiquem conjuntos de amostras da outra classe.

Como vantagem para a classificação baseada em regras, os autores destacam:

- Apropriada para atributos categóricos, ou seja, em vez de usarmos +1 ou -1, o objetivo é identificar um conjunto de classes, por exemplo, textos da literatura portuguesa, textos de medicina e textos de política;
- É mais eficiente (no sentido de que a acurácia é melhor) do que técnicas como Redes Neurais Artificiais, SVM, Redes Bayesianas entre outras técnicas;
- É de fácil compreensão para um ser humano. Tanto nas RNAs quanto nos SVMs e até mesmo nas Redes Bayesianas não se tem uma fácil compreensão da regra ou função hipótese que realiza a discriminação das classes.

Mostraremos mais adiante que as afirmações acima não são sempre verdadeiras. No entanto, métodos como CPAR (*Classification based on Predictive Association Rules*) deixam muito mais evidentes as vantagens dos métodos indutivos como as Redes Neurais Artificiais e SVM.

Em linhas gerais, CPAR funciona da seguinte forma:

- As regras são aprendidas uma por vez;
- A cada vez que uma regra é aprendida, as amostras abrangidas[1] por essa regra são removidas;
- Repete-se todo o processo até que não restem mais amostras a serem classificadas ou até que se atinja um limiar[2].

O termo 'aprendizado' é um pouco forçado neste contexto, mas ele será aqui empregado para seguir a nomenclatura usada na literatura.

Assim, o algoritmo de abrangência sequencial pode ser resumido da seguinte forma:

```
Enquanto (existem amostras não classificadas)
        Gere uma regra
        Remova as amostras positivas[3] que satisfazem à regra
Fim
```

A Figura 3.1 ilustra o conceito das abrangências das regras.

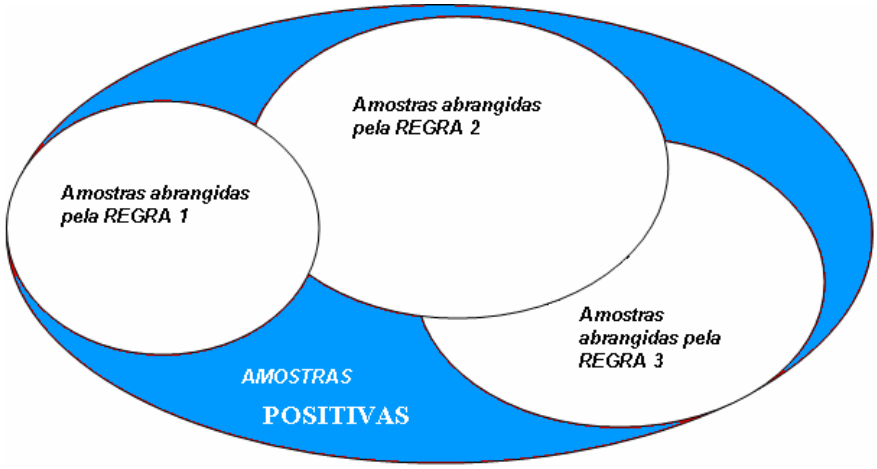



**Figura 3.1. CPAR - Abrangência das regras de associação. Cada conjunto de regras consegue discriminar bem um conjunto de amostras. O objetivo é ter N regras que abranjam o conjunto todo ao invés de ter uma "super" regra.**

[1] Por abrangidas entendam-se as amostras classificadas pela regra.
[2] Por limiar entenda-se: existe um conjunto de regras que classificam com certa precisão um conjunto significante das amostras de uma classe desejada e não classificam amostras de outras classes. Existe então uma margem de acerto e falha.
[3] As amostras positivas são aquelas nas quais as regras de indução estão sendo geradas com o propósito de classificação.

Um exemplo de regra CNF (veja Apêndice B) é: ($A_1$=1) E ($A_2$=3) E ($A_5$=2) $\Rightarrow$ classe positiva

As regras são formadas seguindo os seguintes procedimentos.

- Inicialização das regras: condições = { }
- Geração das regras objetivando alta qualidade/confiabilidade e baixa redundância
- Uso do critério *greedy* (YIN; HAN, 2003) para seleção da literal de melhor qualidade/confiabilidade[4] e adição desta à regra

[4] A qualidade de uma regra leva em consideração abrangência e acurácia. Ou seja, quanto mais amostras da classe positiva forem abrangidas pela regra e quanto menos amostras de outras classes forem selecionadas, melhor será uma regra.

Em (QUINLAN; CAMERON-JONES, 1993) foi realizada uma medição do ganho (ganho de Foil) de informação devido à adição de um novo literal. Assim, o objetivo é sempre buscar pelas literais para maximizar o ganho.

O ganho de Foil pode ser descrito da seguinte forma: antes da adição de uma literal, tínhamos |P| (quantidade) amostras positivas (ou amostras da classe positiva) e |N| amostras negativas. Após a adição da literal teremos, |P'| amostras positivas e |N'| amostras negativas.

$$GanhoFoil = |P'| \cdot \left[ \log\left( \frac{|P|}{|P|+|N|} \right) - \log\left( \frac{|P'|}{|P'|+|N'|} \right) \right]$$

Assim, podemos reescrever o algoritmo da seguinte forma.

```
Enquanto (verdadeiro)
        Encontre o melhor predicado p
        Se o ganho_foil(p)>limiar então
                Adicione p à regra corrente
        Fim
Fim
```

A Figura 3.2 ilustra esse novo conceito. A cada adição de uma literal, podemos abranger mais amostras da classe positiva, mas também corremos o risco de abranger amostras da classe negativa. Temos ao centro, a regra mais específica, e por ser específica esta regra consegue classificar corretamente um conjunto muito pequeno de amostras. À medida que a regra torna-se mais abrangente, mais amostras são classificadas, correta ou incorreta.

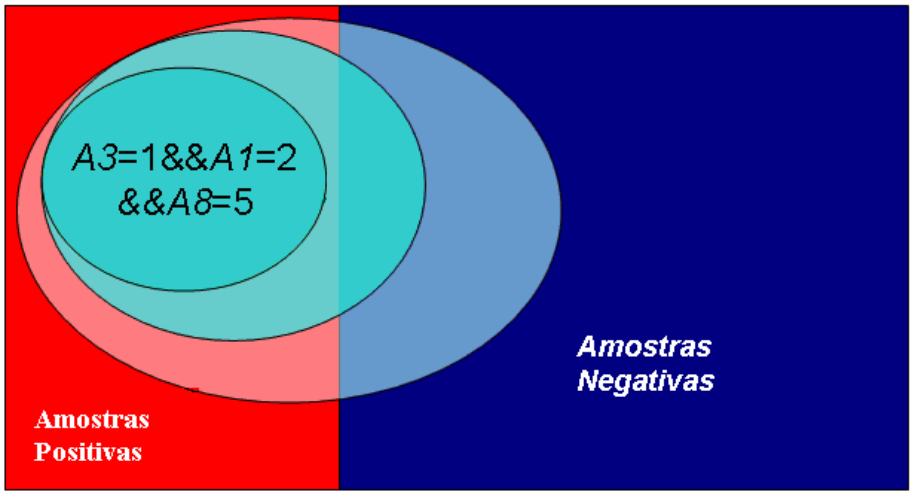



**Figura 3.2. Regras de Associação - Abrangência das regras. Regras muito genéricas comprometem a qualidade do classificador, fornecendo falso positivo.**

As regras de associação apresentam como vantagem sua fácil interpretação e compreensão. No entanto, há um enorme custo computacional na formação das regras, para que funcionem. No contexto da classificação de textos, cada literal é representada pelos atributos que são as palavras usadas no conjunto de textos da classe.

Assim, há inúmeras combinações que funcionam e outras que não funcionam. Encontrar um conjunto de literais seria um processo bem demorado.

## 3.2 Aprendizado Incremental

Quando se fala em aprendizado incremental, existem dois fatores importantes a considerar: o conceito que está sendo assimilado (aprendizagem) muda com o tempo ou é fixo? No primeiro caso damos o nome de "**mudança de conceito**" (*concept drift*) e no segundo caso damos o nome de "**aprendizado incremental**".

Quando temos uma mudança de conceito, as amostras mais antigas podem ser enganosas, já que os novos conceitos são diferentes dos antigos. Já no caso do aprendizado incremental onde não há mudança de conceito, podemos nos concentrar apenas na performance da MA comparando os resultados do treinamento em lote (batch) com o aprendizado incremental. Pode-se comparar a eficiência computacional e também a taxa de acerto ou erro do conceito que está sendo assimilado.

As máquinas de vetores-suporte (*Support Vector Machine* - SVM) ao final do seu processo de treinamento identificam um conjunto de amostras (chamados de vetores-suporte) que representam uma pequena parte do total de amostras que compõem o conjunto de treinamento. Esses vetores-suporte servem de margem para separação de duas classes. Esta característica intrínseca do SVM (seleção de amostras que compõem os vetores-suporte) nos permite investir na abordagem incremental do processo de treinamento, ou seja, podemos trabalhar com uma quantidade menor de dados sem perda de performance.

A princípio, todos os algoritmos SVM que utilizam os métodos dos conjuntos de trabalho (*working set methods*), especialmente os *shrinking* (OSUNA; FREUND; GIROSI, 1997; Platt, 1999) podem ser considerados algoritmos que implementam aprendizado incremental. No entanto, como todas as amostras são reutilizadas no processo iterativo do algoritmo, não há ganho em processamento e consumo de memória, ou seja a complexidade computacional não foi reduzida, assim esses métodos não podem ser considerados aprendizado incremental no sentido atribuído por este trabalho, ou seja, não usar todas as amostras.

Outros autores trabalham com a questão do aprendizado incremental eliminando-se algumas amostras. O aprendizado incremental de (SYED et al., 1999) funciona da seguinte forma, dado um subconjunto inicial A, treina-se o SVM encontrando os vetores-suporte. Esses vetores-suporte são acrescidos a outro subconjunto B para ser treinado novamente pelo SVM para encontrar novos vetores-suporte e assim por diante sucessivamente. O problema nesta abordagem é que o autor está considerando que os subconjuntos selecionados são amostras representativas do total em que a distribuição acompanha também o universo completo.

A condição acima mencionada não é verdadeira para todas as situações, ainda mais quando as amostras são obtidas de forma instantânea sem se poder realizar uma análise prévia. No processo iterativo acima descrito, obtêm-se vetores-suporte a cada nova iteração; esses

elementos estão em número muito pequeno quando comparados ao total de amostras do subconjunto, assim, a sua influência para o estabelecimento das novas margens de separação também serão pequenas, caso a distribuição seja diferente.

Katagiri e Abe, (KATAGIRI; ABE, 2006) tentam resolver o problema da eliminação dos vetores-suporte de um conjunto prévio de treinamento quando se adicionam novas amostras em cuja distribuição há uma rotação dos hiperplanos separadores. Para tanto, uma hiperesfera é gerada para cada um dos lados da classificação. Essa hiperesfera é complementada com um cone de tal forma a selecionar ou eliminar amostras de treinamento para o aprendizado incremental.

Para reduzir o consumo de memória, autores (DOMENICONI; GUNOPULOS, 2001) propuseram um simples método de aprendizado incremental que remove as amostras que não são os vetores-suportes a cada passo do treinamento da SVM, ver a Figura 3.3. Essa abordagem tem como problema remover amostras que ao adicionarmos novos elementos nas iterações sucessivas, se tornariam nos novos vetores-suporte. Desta forma, o aprendizado incremental deve-se preocupar também com os futuros vetores-suporte.

Na Figura 3.3 temos amostras separadas linearmente. O processo incremental trabalha com um subconjunto de amostras (na Figura 3.3, são aquelas com fundo preenchido) para obter os vetores-suporte. Uma vez treinado, novas amostras devem ser acrescidas (na Figura 3.3 são aquelas com fundo não preenchido). O hiperplano separador é indicado pela função f(x) e a margem entre as classes é $2*\delta$. É no parâmetro $\beta$ de ajuste (manual) que se definem as amostras - antigas e novas - a serem excluídas no treinamento do novo subconjunto formado pelos vetores-suporte.

A abordagem de Cauwenberghs e Poggio (CAUWENBERGHS; POGGIO, 2001) estabelece uma região para remoção das amostras baseada na distância do hiperplano separador antes do treinamento incremental (ver a Figura 3.3). O problema nesta técnica está na sua ineficiência quanto à rotação das margens (Figuras 3.4 e 3.5) na medida em que a inclusão de novos elementos no processo iterativo pode alterar a distribuição destes elementos.

Katagiri e Abe (2006) tentam resolver o problema da rotação do hiperplano separador considerando que os vetores-suporte candidatos sempre estarão próximos ao hiperplano separador e também à superfície da região que inclui os vetores-suporte de uma iteração.

Uma hiperesfera (Figura 3.6) de menor volume é inicialmente gerada para cada uma das duas classes de forma a conter todas as amostras. A seguir uma hiperesfera concêntrica de raio menor é gerada. Finalmente o hiper-cone é gerado cujo *vertex* está localizado no centro da hiper-esfera e que abre em direção oposta à margem de separação. Todas as amostras interiores à hiperesfera de menor raio e todas as amostras interiores ao hipercone são apagadas. O restante das amostras são candidatos a futuros vetores-suporte, que serão usados nas iterações seguintes.

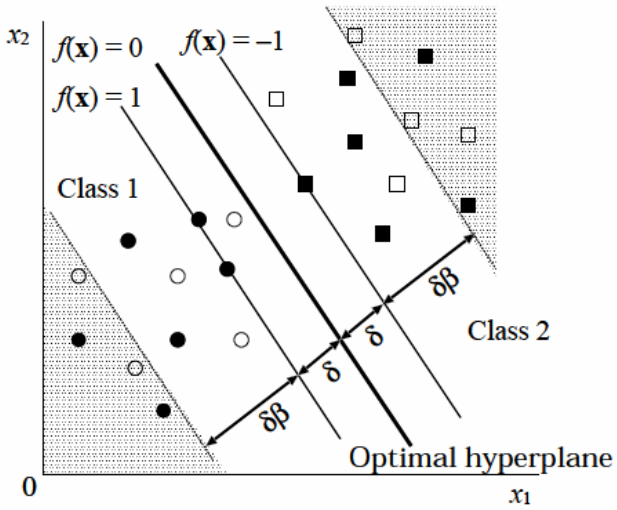



**Figura 3.3. SVM e os elementos que serão apagados (amostras na região sombreada). Verificamos nesta figura a separação linear de duas classes. Amostras com fundo preenchido são aquelas já existentes, e aquelas com fundo não preenchido são amostras novas. Extraído de (KATAGIRI; ABE, 2006)**

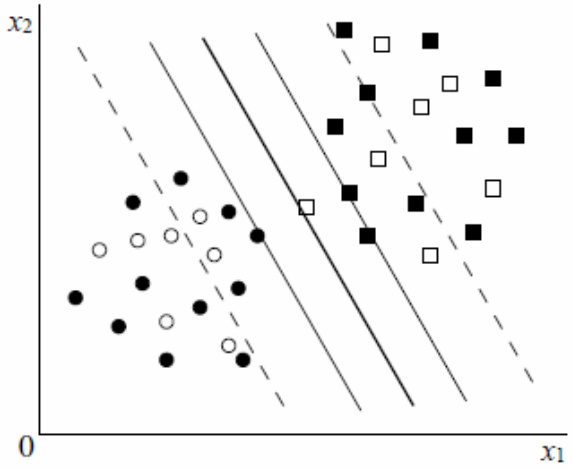



**Figura 3.4. SVM incremental no qual se trabalha com subconjunto das amostras. Hiperplano ótimo das amostras com fundo preenchido. Amostras com fundo branco são os novos elementos que serão introduzidos na iteração seguinte. Extraído de (KATAGIRI; ABE, 2006)**

Na Figura 3.4, as linhas tracejadas indicam uma configuração possível para separação das amostras. Dependendo da escolha das amostras, ou seja, dependendo da formação dos subconjuntos, obtemos diferentes hiperplanos separadores.

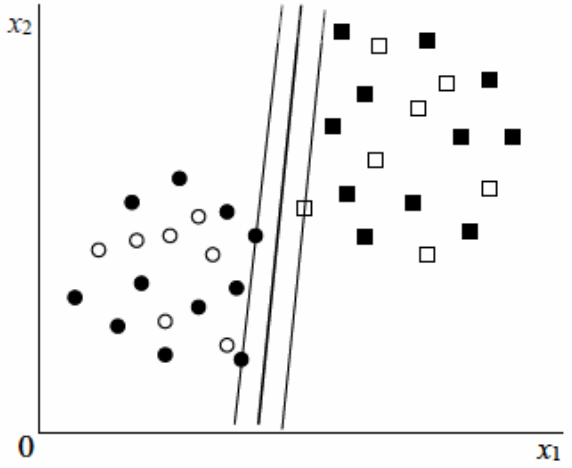



**Figura 3.5. SVM, iteração 2. Rotação do Hiperplano ótimo de separação após introdução de novas amostras e remoção das antigas. Extraído de (KATAGIRI; ABE, 2006)**

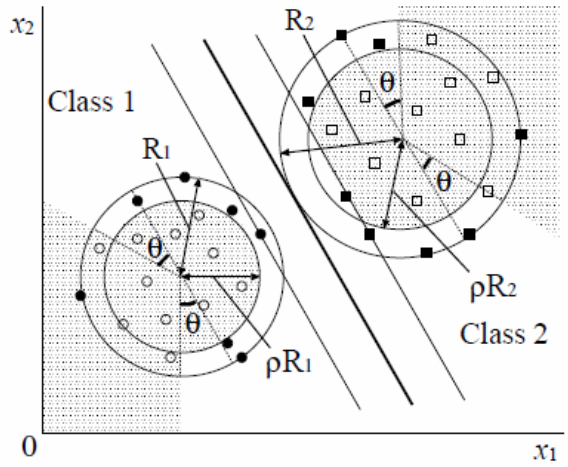



**Figura 3.6. SVM incremental com hiperesfera de raios $R_1$ e $\rho R_2$ e hiper-cone formado usando-se de um ângulo θ ajustável. Extraído de (KATAGIRI; ABE, 2006)**

Em (SYED, et al., 1999), os autores demonstram uma preocupação quanto ao volume de dados a serem processados. Cada vez mais tratamos quantidade de informação volumosa a ponto de atingirmos os limites computacionais, por exemplo, a quantidade de memória disponível num computador. Desta forma, os autores (SYED, et al., 1999) propõem um *Framework* para o aprendizado incremental (Figura 3.7), uma forma de tratar as informações aos blocos, combinando os resultados obtidos sucessivamente. As idéias apresentadas no trabalho de Syed serviram de motivação inicial e referência para a arquitetura proposta neste trabalho, a MTS-SVM apresentada no Capítulo 8.

As máquinas de vetores-suporte (*Support Vector Machine* – SVM) apresentam uma característica (VAPNIK, 1992) que favorece o aprendizado incremental. Dado um conjunto contendo N amostras, os elementos que compõem os vetores-suporte são selecionados. Esses elementos estão em geral na borda ou fronteira entre os elementos que serão classificados. Assim, é possível dividir o conjunto de amostras ou elementos de treino do SVM em pequenos blocos para eleger o vetor suporte a cada treinamento e incluí-lo no treino sucessivo.

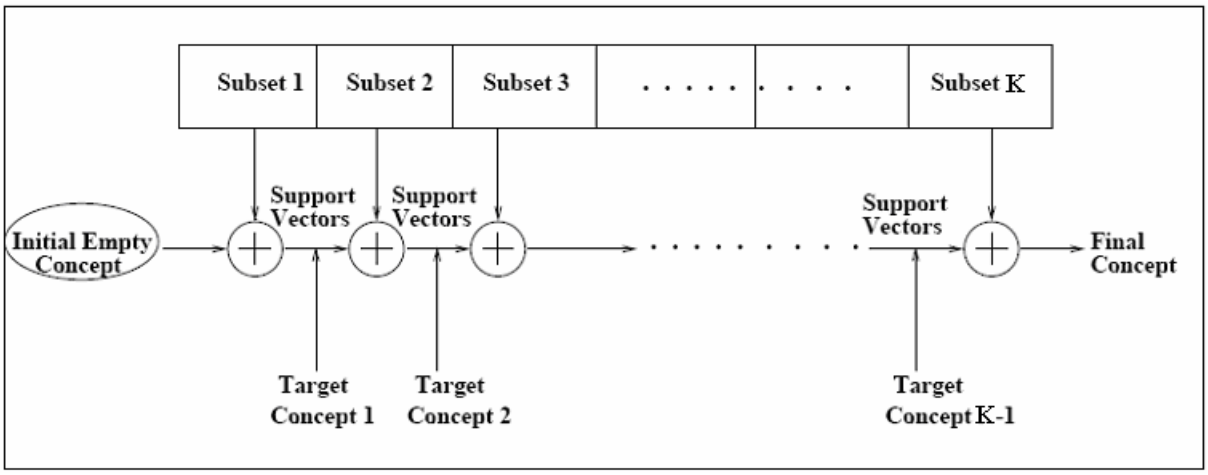



**Figura 3.7. O processo de aprendizado incremental. Extraído de (SYED, et al., 1999). Divisão do conjunto completo em K subconjuntos. Para cada subconjunto aplica-se o SVM, obtendo-se assim um conjunto de vetores-suporte.**

Os resultados obtidos pelos autores mostraram que o aprendizado incremental consegue uma acurácia tão boa quanto o processo *batch* tradicional no qual se consideram todas as amostras de uma única vez.

A melhor forma de minimizar o custo computacional é reduzir a quantidade de amostras. Para não eliminar as amostras importantes propomos uma etapa de filtragem das amostras semelhantes. As amostras filtradas não serão usadas no processo de treinamento. Detalhes deste processo estarão no capítulo de proposta de trabalho mais adiante.

O conceito de subdividir o conjunto de treinamento não é novo; Vapnik em seus trabalhos, já fala a respeito da possibilidade de trabalhar com subconjuntos. Na seção anterior, aprendizado baseado em regras, comenta-se sobre a seleção de atributos formando um subconjunto no qual se aplicará o classificador. Nossa motivação para trabalhar com subconjunto é reforçada pelos resultados de (SYED et al., 1999) e (VAPNIK, 1992).

# 4. Descrição do Domínio de Dados usando Vetores-suporte - SVDD

Existem diversas metodologias e técnicas para obter informações acerca de um conjunto de amostras. Pode-se trabalhar com classificação das amostras segregando aquelas que apresentem algum padrão em comum, pode-se realizar uma análise de regressão descobrindo, por exemplo, o tipo de série temporal, ou pode-se realizar uma análise de agrupamento das amostras juntando todas aquelas que possuam alguma característica em comum.

Apresentaremos neste capítulo o classificador de uma classe, também conhecido como descritor de dados ou identificador de *outliers* ou novidades.

A Figura 4.1 mostra um classificador binário padrão separando a classe das maçãs e a classe das peras por uma linha sólida. Já a linha tracejada separa a classe maçã das não maçãs. Desta forma, a maçã em destaque fora da fronteira seria considerada uma não maçã no classificador de uma classe enquanto que no classificador binário padrão seria considerada uma pêra.

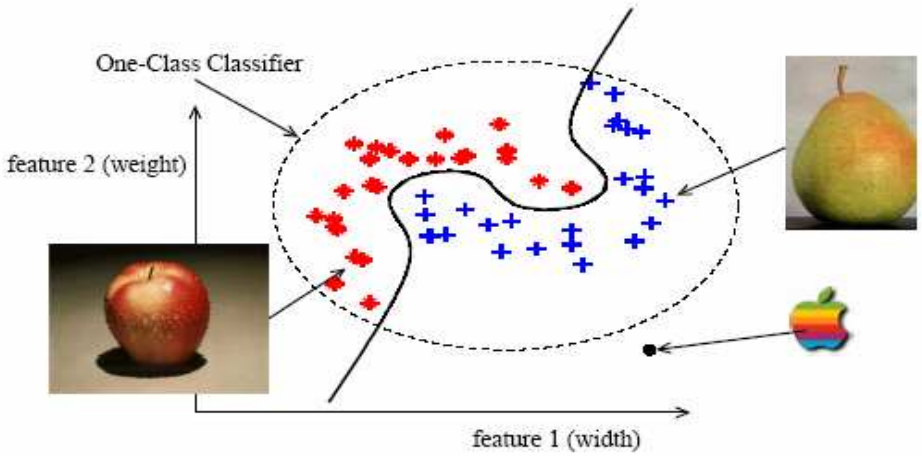



**Figura 4.1. Classificação de uma-classe (são consideradas apenas as amostras de uma classe, por exemplo, as maçãs representadas por *). Extraído de (TAX; DUIN, 1999)**

No SVDD (TAX; DUIN, 1999), por meio de uma função densidade de probabilidade intrínseca das amostras usadas no treinamento da MA, conseguimos definir uma fronteira dentro da qual elas se concentram. Sempre poderão existir os *outliers* que deverão ser tratados de forma a encontrar o menor volume (ou área) que englobe a maioria das amostras e ao mesmo tempo minimize a chance de colocar um *outlier* dentro da fronteira. SVDD usa o SVM para trabalhar com esse *tradeoff*.

Descrever um conjunto de amostras ou descrever o domínio das amostras diz respeito a caracterizar semelhanças entre elas; o resultado prático disso é definir uma fronteira dentro da qual essas amostras se encontram.

O método descrito por Duin baseia-se no SVM; a grande diferença está na definição de uma hiperesfera com volume mínimo (ou menor raio) que contenha todas as amostras alvos (*target*). O trabalho de Duin é uma extensão do trabalho de Schölkpf.

Dado um conjunto de N amostras, $\{x_t, t = 1,..., N\}$, a esfera de centro "a" e raio R pode ser descrita pela equação abaixo:

$$(\mathbf{x_t} - a)(\mathbf{x_t} - a)^T \le R^2 + \xi_t \qquad \textbf{Eq. 4.1}$$
$$\xi_t \ge 0$$

Onde $\xi$ é o parâmetro de ajuste (*slack variable*). O objetivo é minimizar o valor do R e o valor do parâmetro de ajuste.

$$f(R, a, \xi_t) = R^2 + C\sum_t^N \xi_t \qquad \textbf{Eq. 4.2}$$

C é uma constante usada no *tradeoff* entre o volume da esfera e o número de amostras alvas rejeitadas.

Construindo o Lagrangeano para incorporar as restrições, obtemos:

$$L(R, a, \alpha_t, \xi_t) = R^2 + C\sum_t^N \xi_t - \sum_t^N \alpha_t \{R^2 + \xi_t - (x_t^2 - 2ax_t + a^2)\} - \sum_t^N \gamma_t \xi_t \qquad \textbf{Eq. 4.3}$$

Com $\alpha_t \ge 0$ e $\gamma_t \ge 0$

Colocando as derivadas parciais iguais a zero, obtemos:

$$\sum_t^N \alpha_t = 1 \qquad \textbf{Eq. 4.4}$$

$$a = \frac{\sum_{t}^{N} \alpha_t \mathbf{x_t}}{\sum_{t}^{N} \alpha_t} = \sum_{t}^{N} \alpha_t \mathbf{x_t}$$

$$0 \leq \alpha_t \leq C$$

Re-escrevendo o Lagrangeano, considerando as equações acima, temos:

$$L = \sum_{i}^{N} \alpha_i (\mathbf{x_i} \cdot \mathbf{x_i}) - \sum_{i,j}^{N} \alpha_i \alpha_j (\mathbf{x_i} \cdot \mathbf{x_j}) \qquad \textbf{Eq. 4.5}$$

Índices i e j utilizados no lugar de t na Equação 4.5 e 4.6 porque duas variáveis estão em evidência.

Assim, uma amostra de teste **z** estará contida na fronteira se satisfazer a equação abaixo:

$$(z-a)(z-a)^T = (z \cdot z) - 2\sum_{i}^{N} \alpha_i (z \cdot \mathbf{x_i}) + \sum_{i,j}^{N} \alpha_i \alpha_j (\mathbf{x_i} \cdot \mathbf{x_j}) \leq R^2 \qquad \textbf{Eq. 4.6}$$

Da mesma forma como no SVM, existirão situações em que será necessário transportar as amostras do seu espaço original para uma dimensão maior de forma que se consiga realizar a separação entre os elementos positivos e negativos (em uma classificação de duas classes). Só que no caso do SVDD precisamos ajustar uma esfera (fronteira) que englobe as amostras; muitas vezes isso não será possível no espaço original. Empregamos então as funções *kernel*, que são a representação do produto interno no espaço dos atributos.

Seja a função *K* uma função *kernel*, temos então o Lagrangeano abaixo:

$$L = \sum_{i}^{N} \alpha_i K(\mathbf{x_i}, \mathbf{x_i}) - \sum_{i,j}^{N} \alpha_i \alpha_j K(\mathbf{x_i}, \mathbf{x_j}) \qquad \textbf{Eq. 4.7}$$

O teste de aceitação da amostra de teste z deverá atender à equação abaixo:

$$K(z \cdot z) - 2\sum_{i}^{N} \alpha_i (z \cdot \mathbf{x_i}) + \sum_{i,j}^{N} \alpha_i \alpha_j (\mathbf{x_i} \cdot \mathbf{x_j}) \leq R^2 \qquad \textbf{Eq. 4.8}$$

A função *kernel* mais adequada para o SVDD é a gaussiana, descrita a seguir:

$$K(\mathbf{x_i}, \mathbf{x_j}) = \exp\left(-(\mathbf{x_i} - \mathbf{x_j})^2 / s^2\right) \qquad \textbf{Eq. 4.9}$$

Onde s é um parâmetro livre que poderá ser ajustado. No caso de uma função gaussiana, a seguinte identidade é valida:

$$K(\mathbf{x_i}, \mathbf{x_i}) \equiv 1 \qquad \textbf{Eq. 4.10}$$

Assim, podemos reescrever o Lagrangeano da seguinte forma:

$$L = 1 - \sum_{i}^{N} \alpha_i^2 - \sum_{i \neq j}^{N} \alpha_i \alpha_j K(\mathbf{x_i}, \mathbf{x_j}) \qquad \textbf{Eq. 4.11}$$

## 4.1 Classificação de Uma-Classe segundo Schölkopf

A classificação de uma-classe é realizada selecionando-se apenas as amostras positivas para identificar os *outliers*. Assim a classificação de uma-classe pode ser descrita por:

Suponha que um conjunto de dados ou amostras apresente uma distribuição de probabilidade P no espaço dos atributos. Encontre um subconjunto S no espaço dos atributos, de tal forma que a probabilidade do ponto de teste de P fique fora de S é limitada por um valor $v \in (0,1)$ definido a priori (MANEVITZ; YOUSEF, 2001, tradução nossa).

A solução para este problema é obtida estimando uma função f que seja positiva em S e negativa em seu complemento, ver a Equação 4.12.

$$f(\mathbf{x}) = \begin{cases} +1 & se \quad x \in S \\ -1 & se \quad x \in \overline{S} \end{cases} \qquad \textbf{Eq. 4.12}$$

Suponha que $x_1, x_2, \ldots, x_N$ sejam as amostras de treino pertencentes a uma classe X, onde X é um conjunto compacto do $R^{\aleph}$ (um espaço de n dimensões). Seja $\Phi : X \rightarrow H$ um mapeamento do espaço original para o espaço dos atributos. O objetivo do algoritmo é separar as amostras X mapeadas no espaço dos atributos H, separando-as da origem com a maior margem possível, veja a Figura 4.2.

Assim, para separar as amostras com relação à origem (cruzamento dos eixos coordenados que representam as dimensões do espaço das amostras) devemos resolver a seguinte equação quadrática com restrições lineares.

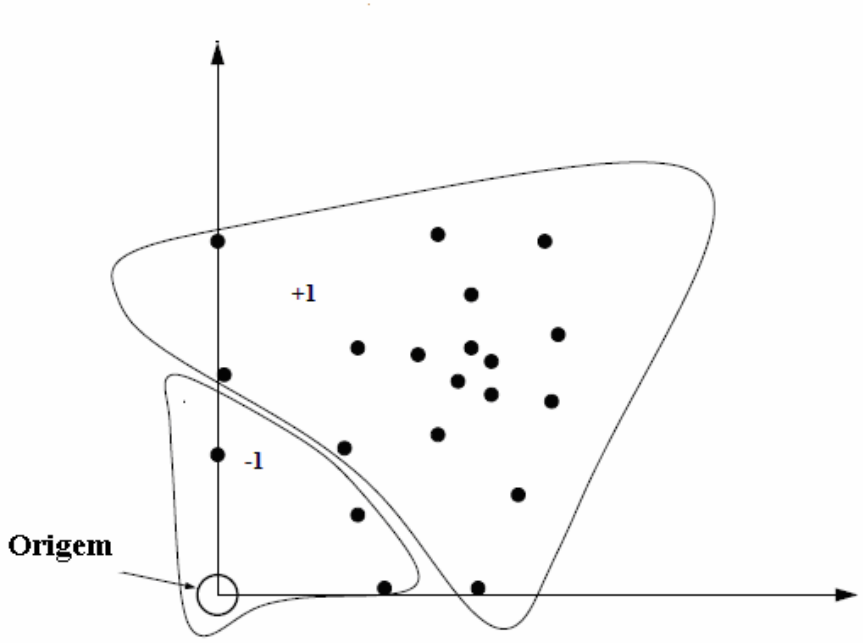



**Figura 4.2. Ilustração do conceito de classificação de uma-classe. Dadas apenas amostras da classe positiva, uma amostra da classe negativa pode estar localizada na origem. A origem é o único membro da classe negativa. Extraído de (MANEVITZ; YOUSEF, 2001).**

$$\min \frac{1}{2}\left\|\mathbf{w}^2\right\| + \frac{1}{vN}\sum_{t=1}^{N}\xi_t - \rho \qquad \textbf{Eq. 4.13}$$

Sujeito a

$$\begin{gathered}\left(\mathbf{w}\cdot\Phi(\mathbf{x_t})\right) \geq \rho - \xi_t \\ \text{t=1,2,... N} \\ \xi_t \geq 0\end{gathered} \qquad \textbf{Eq. 4.14}$$

Se **w** e ρ resolvem o problema acima, então a função de decisão será:

$$f(\mathbf{x}) = sign\left(\left(\mathbf{w}\cdot\Phi(\mathbf{x})\right) - \rho\right) \qquad \textbf{Eq. 4.15}$$

Ela será positiva para a maioria das amostras $\mathbf{x}_i$ contidas no conjunto de treino.

A classificação e a recuperação de documentos são de grande importância nos sistemas de informação. Quando temos à disposição apenas elementos de uma classe, temos o que é chamado classificação de uma-classe.

Manevitz e Yousef (MANEVITZ; YOUSEF, 2006) tratam a classificação de uma-classe usando as redes neurais artificiais do tipo *feed-forward*. O propósito do trabalho desses autores é filtrar um conjunto de documentos selecionando apenas aqueles de interesse.

Os documentos de interesse são as amostras positivas a serem usadas no treinamento da rede. A topologia da rede está apresentada na Figura 4.3. Temos uma rede de três camadas, na qual a entrada e a saída são compostas de m neurônios e a camada intermediária apresenta k neurônios em que k<m. A rede é treinada para aprender a função identidade sobre as amostras.

As amostras apresentam dimensão m; assim, essa rede realiza uma compressão das amostras de entrada. Um nome popularmente utilizado para essa rede é "*autoencoder*" inicialmente utilizado por (JAPKOWICZ; MYERS; GLUCK, 1995).

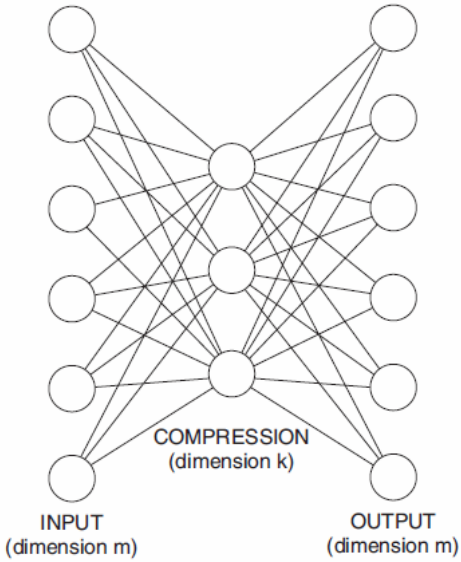



**Figura 4.3. Rede "feed-forward" com gargalo (*bottleneck*). Extraído de (MANEVITZ; YOUSEF, 2006)**

A idéia básica da rede *auto encoder* está no aprendizado de uma função identidade de dimensão k menor que a dimensão m, ou seja, a rede trabalha com um subespaço do espaço original. Após o treinamento da rede, dada uma amostra de entrada, se a saída for ela mesma (identidade), a amostra é considerada "interessante".

Manevitz e Yousef trabalharam com cerca de seis neurônios (ajustados empiricamente) na camada oculta. Os documentos usados são de uma base pública, a Reuters-21578 (LEWIS,

1997). Para cada uma das categorias dos documentos disponíveis em Reuters-21578, 25% das amostras (positivas) foram utilizadas para treinamento. Os testes foram realizados com o restante das amostras positivas e também foram usadas as amostras negativas.

A seguir, um exemplo de aplicação da classificação de uma-classe será apresentada.

## 4.2 Exemplo de Classificação de Uma-Classe – Detecção de Intrusão

Sistemas para detecção de intrusão (*intrusion detection system* - IDS) são fundamentais para a proteção de uma rede de computadores. Classificação de uma-classe tem sido empregada (PERDISCI; GU; LEE, 2006) para identificar tráfegos maliciosos, obtendo-se bons resultados.

Existem duas abordagens (MCHUGH; CHRISTIE; ALLEN, 2000) para detecção de intrusão, são elas:

- *Misuse*: quando buscamos por ataques conhecidos chamados de assinaturas ou padrões de ataque. São atividades ou comportamentos maliciosos já conhecidos.
- *Anomaly*: neste caso existem modelos de normalidade. Variações muito grandes em relação ao modelo normal (por exemplo, usando a distância de Mahalanobis) são consideradas ataque.

O problema no uso da técnica *Misuse* está no fato de se necessitar de uma constante atualização da assinatura ou padrão de ataque. Já a detecção de anomalia enfrenta o problema da alta taxa de falso positivo. A vantagem da detecção de anomalia está na possibilidade de identificar ataques novos, os chamados ataques de zero – dias; já no caso da técnica *misuse* isto não é possível.

A motivação pelo uso das técnicas de aprendizado não supervisionado vem das hipóteses de que padrões de tráfego normal extraído dos registros reais de tráfego de rede são muito maiores do que os padrões anormais. Além disso, os padrões de assinatura também podem ser utilizados em conjunto para destacar mais os padrões anormais em um conjunto de tráfego.

Um *payload* anômalo é aquele cuja distância de Mahalanobis entre o *payload* de teste e o *payload* de tráfego normal excede um limiar pré-estabelecido (MCHUGH; CHRISTIE; ALLEN, 2000).

A detecção de anomalia não é algo simples. Padrões de comportamento normal podem ser imitados (alterando-se alguns bytes) de forma que esse padrão esteja dentro da superfície separadora que engloba os padrões normais de tráfego, ou seja, temos um falso negativo. Assim torna-se necessário um maior estudo sobre os padrões normais (amostras positivas) de tráfego.

McHugh utiliza um conjunto de classificadores para atacar o problema de imitação de tráfego normal. Além disso, utiliza o modelo 2-grama (WANG; STOLFO, 2004) modificado, o $2_{\nu}$-grama, onde $\nu$ representa um deslocamento em relação a posição do byte corrente. Ou seja, é uma análise 2-grama na qual os bytes estão separados de $\nu$ bytes.

A função *kernel* utilizada foi a gaussiana. E o gama utilizado foi 0.5. O parâmetro nu foi variado para encontrar o melhor valor.

# 5. Análise dos Atributos

Neste capítulo apresentaremos as bases de dados utilizadas nas pesquisas realizadas. Abordaremos também a extração e análise dos atributos.

A análise dos atributos constitui uma das etapas mais importantes em qualquer técnica de aprendizado de máquina. Uma análise mal conduzida poderá levar o sistema especialista ao fracasso. São os atributos que melhor caracterizam as amostras de um conjunto em estudo e, assim sendo, um capítulo inteiro lhe será dedicado.

Todo experimento foi conduzido usando um computador comum, com a seguinte configuração: Pentium 1.8GHz CPU e 1GB de memória RAM.

## 5.1 Base de dados

As bases de dados utilizadas neste trabalho são apresentadas nesta seção. Duas bases de dados contendo elementos textuais foram selecionadas, a Reuters-21578 e a 20 Newsgroup; essas bases são frequentemente citadas em artigos que abordam classificação de textos ou documentos. Além delas, a "UCI-Adult dataset" e a "UCI-Mushroom dataset" foram também usadas para mostrar que as estratégias e os algoritmos apresentados nesta pesquisa são extensíveis a outros domínios.

### 5.1.1 Reuters-21578 – base de dados textual

Esta base é uma base de dados pública da agência de notícias Reuters (www.reuters.com), em que os documentos já estão previamente classificados e identificados para serem utilizados no treinamento ou teste. Ela é utilizada como *benchmark* pelos pesquisadores (SEBATIANI, 2002; JOACHIMS, 1999) no campo da Recuperação da Informação, assim, diferentes pesquisadores podem compartilhar seus resultados.

A coleção contém um conjunto de 21.578 notícias do ano de 1987, e tais notícias são consideradas documentos ou textos. Os documentos dessa coleção estão categorizados em 135 categorias, havendo alguns poucos documentos que aparecem em mais de uma classe. Em geral as categorias abordam temas da economia e negócios. A distribuição dos documentos em cada uma dessas classes não é uniforme, algumas apresentam mais elementos, enquanto que outras, menos.

Há ainda uma relação intrínseca entre as classes, ou seja, não existe uma amarração ou ligação explícita entre as classes, mas elas mantêm uma relação de semelhança, por exemplo, as classes TRIGO e GRÃO apresentam muitas características comuns a ambas.

Muitos pesquisadores têm trabalhado com apenas um subconjunto das classes disponíveis nessa coleção, e muitas vezes, a escolha desses subconjuntos impede uma comparação de resultados mesmo usando-se a mesma base de dados.

A partição no modo Apté (*ModApté split*) tem sido a forma mais comum de seleção das amostras e das classes a serem utilizadas no treinamento e teste de máquinas de aprendizado. Ela consiste em:

- O conjunto das 10 categorias com a maior quantidade de amostras positivas para treinamento é utilizado. Este conjunto recebe o nome de R10
- O conjunto das 90 categorias com pelo menos uma amostra positiva para treinamento e pelo menos uma amostra positiva para teste é utilizado. Este conjunto recebe o nome de R90
- O conjunto das 115 categorias com pelo menos uma amostra de treinamento é utilizado. Este conjunto recebe nome de R115

No modo Apté, o que tem sido mais frequente é a seleção de 9603 documentos para treinamento e 3299 documentos para teste.

Foi ela montada e disponibilizada aos pesquisadores das diversas áreas e que utilizam de técnicas de aprendizado de máquinas diferentes para que possam comparar seus resultados.

No entanto, a Reuters-21578 apresenta diversos parâmetros de refino e assim, cada pesquisador pode adotar um critério diferente em cima da mesma base. Debole e Sebatiani (2004) mostram essa preocupação quando evidenciam a possibilidade da escolha de subconjuntos diferentes pelos pesquisadores.

A base de dados está disponível em:
http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html

### 5.1.2 20 Newsgroup – base de dados textual

A coleção de dados 20 Newsgroup ou 20NG (LANG, 1995) contem aproximadamente 20.000 mensagens enviadas nas listas de discussão da *internet*. Algumas classes apresentam similaridade, enquanto que outras não apresentam nenhuma relação.

Esta base de dados está disponível em:
http://people.csail.mit.edu/people/jrennie/20Newsgroups

Há um total de 20 categorias, todas com quantidade de amostras bem próxima. Cada categoria está dividida entre amostras de treino e teste. As classes disponíveis nessa base são:

- Comp
  - comp.graphics
  - comp.os.ms-windows.misc
  - comp.sys.ibm.pc.hardware
  - comp.sys.mac.hardware
  - comp.windows.x
- Misc
  - misc.forsale
- Talk
  - talk.politics.misc
  - talk.politics.guns
  - talk.politics.mideast
- Rec
  - rec.autos
  - rec.motorcycles
  - rec.sport.baseball
  - rec.sport.hockey
- Sci
  - sci.crypt
  - sci.electronics
  - sci.med
  - sci.space
- Others
  - talk.religion.misc
  - alt.atheism
  - soc.religion.christian

A 20NG foi processada da seguinte forma:

- Cabeçalhos dos emails foram removidos
- Os mesmos stopwords gerados no tratamento da base da Reuters-21578 foram usados
- Os mesmos processos de radicalização empregados na Reuters-21578 foram aplicados

Esta base foi utilizada para testes do PTC e NSMO.

### 5.1.3 UCI – Adult dataset

A base de dados "adult dataset" (KEERTHI; LIN, 2003) contém dados do censo americano de 1994 para predição (ou classificação) da renda anual de uma pessoa. Com 14 atributos e 48.842 amostras a base apresenta amostras já identificadas com o valor da renda, maior que U$ 50 mil, ou então, menor ou igual a U$ 50 mil.

A base de dados está disponível em:
http://archive.ics.uci.edu/ml/datasets/Adult

Não foi realizado nenhum tipo de processamento sob as amostras da base. Esta base foi utilizada para testes do Subset SVM.

### 5.1.4 UCI – Mushroom dataset

A base de dados de cogumelos apresenta atributos que descrevem as características físicas do cogumelo. A finalidade é classificar os cogumelos entre os comestíveis e os venenosos. Com 22 atributos e 8124 amostras, essa base é de 1981.

Não foi realizado nenhum tipo de processamento sob as amostras da base.

A base de dados está disponível em:
http://archive.ics.uci.edu/ml/datasets/Mushroom

Esta base de dados foi utilizada para testes do Subset SVM.

## 5.2 Representação dos Documentos

Os documentos ou textos são compostos por palavras. Eles precisam ser representados em alguma forma numérica para que uma MA consiga treinar um classificador. Todas as palavras de todos os documentos que compõem as amostras de treino foram extraídas e processadas (normalização ou *stemming*). Essas palavras são colocadas em uma representação vetorial onde cada elemento do vetor corresponde a uma única palavra distinta, temos desta forma uma representação vetorial do documento. Cada palavra do espaço vetorial representa uma dimensão das amostras (o documento ou texto).

A quantidade de palavras existentes em um idioma é grande, as palavras processadas precisam passar por algum critério de seleção para redução no número de atributos.

Uma das formas para redução da dimensão é a seguinte: as palavras foram ordenadas segundo a sua frequência de ocorrência, ou seja, ordenação decrescente segundo a quantidade de ocorrências totais de cada palavra no conjunto (total) de documentos. E então, a partir de um processo empírico apenas as m melhores (segundo ordenação decrescente de ocorrência) palavras são utilizadas. Esta lista de m palavras recebe o nome de *keywords*.

Existem outras abordagens para redução da dimensionalidade; a informação mútua (MI, *mutual information*), taxa de ganho (GR, *gain ratio*) e estatística Chi-quadrada (DEBOLE; SEBASTIANI, 2003; YANG; PEDERSEN, 1997) são algumas das mais usadas.

A representação dos documentos utilizada pode ser dividida em:

**1.** representação binária: neste caso, sempre que uma palavra aparecer em um documento sua posição no vetor de palavras (*keywords*) recebe 1, caso contrário recebe 0. Desta forma, para cada documento temos uma representação vetorial (binária) de dimensão m.

**2.** representação em frequência: neste caso, temos uma representação vetorial (de números reais) na qual cada posição da palavra recebe o valor da frequência (normalizada) de ocorrência da palavra em cada documento.

**3.** Tf-idf *representation*: "term frequency inverse document frequency". Neste caso, uma variação da representação por frequência tem a fórmula abaixo:

$$tf-idf\left(palavra_chave\right)=frequencia\left(palavra_chave\right)\cdot\left[\log\frac{n}{K\left(palavra_chave\right)}+1\right] \quad \textbf{Eq. 5.1}$$

Onde n representa o número de documentos do conjunto de treino, e K representa uma função que retorna a quantidade de documentos nos quais a palavra (*keyword*) aparece.

**4.** Hadamard *representation*: ou representação do produto de Hadamard, cada índice do vetor m dimensional recebe o produto da frequência de ocorrência da palavra dentro do documento

pelo valor correspondente à frequência de ocorrência desta mesma palavra em todos os documentos.

Existe uma série de variações da representação TF-IDF, as representações SMART (SALTON, 1971), o peso de cada atributo é representado pela tripla {contribuição do TF}{contribuição do IDF}{normalização}, veja abaixo mais detalhes:

- Contribuição do TF:
  - **n**: usar o tf (frequência do termo) normalmente
  - **l**: Log(1) + ln(tf)
  - **a**: aumentado, ou seja, é normalizado entre 0.5 e 1.0 da seguinte forma

    $$0.5 + \frac{tf}{2 \cdot MaxTF}$$
- Contribuição do IDF:
  - **n**: nenhuma, ou seja, sempre 1
  - **t**: tradicional, ou seja, é expresso por $\log\left(\frac{n+1}{K}\right)$, onde n é o número de documentos em uma coleção e *K* uma função que retorna o número de documentos onde há ocorrência de uma palavra
- Contribuição da normalização:
  - **n**: nenhuma contribuição
  - **c**: normalização co-seno

| Representação SMART | Pesos |
|---|---|
| NNN | $w_{ij}=tf_{ij}$ |
| NTC | $w_{ij} = \frac{idf_j \cdot tf_{ij}}{\sqrt{\sum_{k=1}^{n} (idf_k \cdot tf_{ik})^2}}$ |
| ATN | $w_{ij} = idf_j + \left(0.5 + \frac{tf_{ij}}{2 \cdot MaxTF_i}\right)$ |

Segundo (BUCKLEY, 1993) as evidências mostram que métodos de atribuição de pesos realizados de maneira adequada é muitas vezes mais importante do que a seleção dos atributos. Isso não quer dizer que a seleção dos atributos não é importante, os dois mecanismos são fundamentais para a obtenção de bons resultados na classificação.

Em (SADAT; YOSHIKAWA; SHUNSUKE, 2003) a representação ATN e NTC mostraram melhores resultados, e em (LEE; PARK; CHOI, 2001) a representação ATN e NTC também apresentaram bons resultados. Diante desses fatos, resolve-se testar as representações NNN, NTC e ATN sobre a base de dados 20Newsgroup.

As médias F1 Micro e Macro (ver a Tabela 5.1) serão utilizadas para apresentação dos resultados de classificação. A Tabela 5.2 (DEBOLE; SEBATIANI, 2004) abaixo mostra os resultados comparativos da classificação dos documentos utilizando as subdivisões da base de dados da Reuters-21578.

As médias F1 Micro e Macro são formas de verificar o desempenho da classificação levando-se em consideração os valores da precisão (*precision*) e revocação (*recall*). A precisão (Equação 5.2) é a razão do número de documentos classificados corretamente como de classe positiva pelo número total de documentos classificados como de classe positiva.

Já a revocação (Equação 5.3) leva em consideração o total de amostras classificadas como positivo (como no caso da precisão) em relação ao total real (atribuído por um especialista).

A combinação dos casos possíveis da precisão e revocação é usada da seguinte forma:

- *Micro-averaging*: relação entre a quantidade de amostras consideradas positivas e negativas, ambas em acordo entre o classificador e o especialista, em relação a todas as outras possibilidades. Ver a Equação 5.4.
- *Macro-averaging*: relação entre a quantidade de amostras em desacordo entre o classificador e o especialista, são amostras consideradas da classe negativa pelo classificador, mas em desacordo com o especialista, ou vice-versa, em relação a todas as outras possibilidades. Ver a Equação 5.5.

Abaixo estão as equações de cada caso, precisão, revocação, micro e macro médias. Na Tabela 5.1 estão as combinações possíveis.

$$\text{Precisão} = \text{P} = \frac{VP}{VP + FP} \qquad \textbf{Eq. 5.2}$$

$$\text{Revocação} = \text{R} = \frac{VP}{VP + FN} \qquad \textbf{Eq. 5.3}$$

$$\text{Precisão - Média micro} = \text{p}_{\text{ma}} = \frac{\sum_{i=1}^{m} VP_i}{\sum_{i=1}^{m} VP_i + \sum_{i=1}^{m} FP_i}$$

$$\text{Revocação - Média micro} = \text{r}_{\text{ma}} = \frac{\sum_{i=1}^{m} VP_i}{\sum_{i=1}^{m} VP_i + \sum_{i=1}^{m} FN_i} \qquad \textbf{Eq. 5.4}$$

$$\text{F1 - Média micro} = \frac{2 \times p_{ma} \times r_{ma}}{p_{ma} + r_{ma}}$$

$$\text{Precisão - Média macro} = \text{p}_{\text{mac}} = \frac{\sum_{i=1}^{m} P_i}{m}$$

$$\text{Revocação - Média macro} = \text{r}_{\text{mac}} = \frac{\sum_{i=1}^{m} R_i}{m} \qquad \textbf{Eq. 5.5}$$

$$\text{F1 - Média Macro} = \frac{2 \times p_{mac} \times r_{mac}}{p_{mac} + r_{mac}}$$

Na equação 5.5, $P_i$ e $R_i$, referem-se a, respectivamente, precisão e revocação de cada um dos pares de classes treinados.

**Tabela 5.1 Combinação entre os possíveis resultados numa classificação. VP=Verdadeiro Positivo, VN=Verdadeiro Negativo, FP=Falso Positivo, FN=Falso Negativo.**

| | | Especialista | |
|---|---|---|---|
| **Categoria** | | **POSITIVO** | **NEGATIVO** |
| **Classificador** | **POSITIVO** | VP | FP |
| | **NEGATIVO** | FN | VN |

Pode-se verificar a variação dos resultados para o mesmo algoritmo de classificação e para o mesmo procedimento de extração de atributos. Quanto menor o subconjunto, melhor é o resultado. Este resultado está relacionado com a quantidade de amostras disponíveis para cada classe.

Os índices de acerto (medida F1) (DEBOLE; SEBATIANI, 2004) na classificação foram comparados com outras técnicas como Redes Neurais Artificiais, Naive Bayes e Vizinhança

mais próxima. SVM apresentou o melhor resultado, mas as outras técnicas apresentaram resultados bem próximos ao SVM.

**Tabela 5.2. Média e desvio padrão de todos os resultados da classificação usando cada um dos subconjuntos da Reuters-21578, R10,R90 e R115. Extraído de (DEBOLE; SEBATIANI, 2004).**

| | Média Micro F1 | | Média Macro F1 | |
|---|---|---|---|---|
| | **Média** | **Desvio Padrão** | **Média** | **Desvio Padrão** |
| **R(10)** | 0.852 | 0.048 | 0.715 | 0.097 |
| **R(90)** | 0.787 | 0.059 | 0.468 | 0.068 |
| **R(115)** | 0.784 | 0.062 | 0.494 | 0.118 |

## 5.3 Análise dos Atributos – Seleção dos Atributos

A análise de atributos (*feature analysis*) é muito dependente dos tipos de amostras que estejam em estudo. Por exemplo, para análise de imagens temos uma vasta metodologia para extração das características que melhor realçam essas imagens. E mesmo dentro do mesmo contexto de análise de imagens, para cada tipo de imagem haverá diversos tipos de extração de atributos.

As imagens médicas (geralmente em tons de cinza quando se trata de ressonância magnética) são diferentes das fotos tiradas de uma paisagem utilizando-se uma câmera fotográfica de uso comum. Ainda sobre as imagens, podem-se pretender finalidades diferentes, podemos querer extrair contornos ou formas geométricas.

Este trabalho aborda a aplicação das máquinas de aprendizado para a classificação de documentos ou textos. Assim, os atributos a serem extraídos também serão bem específicos para esta aplicação.

No processo de classificação de documentos as amostras são representadas pelos documentos e os atributos são representados pelas palavras que compõem o texto. Cada texto é formado por um conjunto de palavras que podem se repetir. Mas nem todo texto apresenta todas as palavras de referência.

As palavras de referência compõem o chamado modelo vetorial (*vector space model)*; ele nada mais é do que uma relação distinta de todas as palavras que aparecerem nos textos que

compõem as amostras de treinamento. No limite teremos todas as palavras da língua segundo a qual o texto foi escrito, mas na prática uma quantia muito menor é utilizada, cerca de 5000 podendo variar de acordo com a área em estudo.

Assim, temos N amostras no conjunto de treino, cada qual com uma quantia distinta de palavras, ou seja, as amostras de treinamento apresentam atributos bem esparsos.

Como outro exemplo, podemos citar a classificação dos mamíferos; nem todos os mamíferos possuem pelos. Ou seja, atributos esparsos ocorrem com muita frequência em diversas áreas de estudo. Como primeira tentativa para "compactar" a dimensão do espaço dos atributos, foi realizado o seguinte experimento (voltando agora para a classificação dos textos).

Dado N textos, obtêm-se seus atributos (ou palavras). Em seguida é realizada uma ordenação das amostras que compõem o conjunto de treino segundo a frequência dos atributos. Atributos mais frequentes (que aparecem em um conjunto distinto de textos) são selecionados e os respectivos textos também.

Após a identificação dos atributos com maior frequência de ocorrência, **A** atributos foram selecionados segundo sua ordem decrescente. Obteve-se assim uma representação vetorial para os documentos. É natural alguns documentos serem excluídos, provisoriamente, do conjunto de treinamento por não terem nenhum atributo dentre aqueles com maior ocorrência, ou seja, conjunto A de atributos.

Atributos com maior frequência de ocorrência abrangem mais de 80% dos textos pertencentes a uma mesma classe ou conjunto, seguindo a distribuição esperada das palavras de qualquer idioma (lei de *Zipf*). Essa frequência, no entanto, não serve de critério decisivo para classificação de documentos. Podem existir outras classes que apresentarão palavras que se repetem na mesma frequência.

Como feito normalmente, as palavras conhecidas como *stop words* foram removidas. Foram definidas cerca de 400 palavras como *stop words* neste trabalho.

Todas as palavras foram radicalizadas (*stemmed*) (PORTER, 1980) usando o algoritmo de remoção de sufixo de Porter, ou seja, as palavras foram reduzidas a sua forma-base ou raiz.

As palavras como correndo, conversando e cantando são reduzidas a correr, conversar e cantar. Dessa forma, as palavras que outrora seriam consideradas diferentes podem ser contabilizadas como uma mesma palavra, aumentando assim a frequência de ocorrência.

O algoritmo de Porter foi concebido para o idioma inglês, mas já existem variações do algoritmo de radicalização para o nosso idioma português. Em (SOARES; PRATI; MONARD, 2009) os autores apresentam suas propostas.

Nas Tabelas 5.3 e 5.4 estão alguns valores da medida de semelhança entre as classes de textos em estudo (Reuters-21578, descrição da base de dados na próxima seção). Como medida de semelhança ou diferença foi utilizada a divergência de Kullback-Leibler (KL) para comparar as distribuições de probabilidades entre duas classes. A divergência KL faz parte da chamada f-divergência, uma classe de divergências que são medidas invariantes [MORENO; HO; VASCONCELOS, 2004].

Percebe-se que a divergência é baixa entre as mesmas classes (neste caso, utilizou-se a classe de teste como comparativo em relação à classe de treino) e maior entre classes diferentes. A Tabela 5.3 não é simétrica porque para cada classe, um espaço vetorial que representa a classe é gerada, assim, a representação dos documentos muda de acordo com a classe referência.

As classes pré-classificadas do conjunto de documentos da Reuters-21578 usadas são as "Earn", "Acq", "Crude", "Money-fx" e "Grain". Essas divisões referem-se à classificação realizada pela própria Reuters, podendo ter documentos pertencentes a mais de uma classe. Cada classe possui um conjunto de documentos divididos para treinamento e teste.

**Tabela 5.3. Cálculo da divergência de Kullback-Leibler entre classes de documentos.**

| Classe Referência | Earn | Acq | Crude | Money | Grain |
|---|---|---|---|---|---|
| **Earn** | 0 | 0,301 | 0,203 | 0,299 | 0,318 |
| **Acq** | 0,323 | 0 | 0,144 | 0,279 | 0,160 |
| **Crude** | 0,443 | 0,332 | 0 | 0,102 | 0,005 |
| **Money** | 0,516 | 0,419 | 0,137 | 0 | 0,146 |
| **Grain** | 0,543 | 0,386 | 0,040 | 0,016 | 0 |

Esses dados nos permitem dizer que a distribuição das palavras caracteriza um documento pertencente a uma classe. No entanto, ela não é condição suficiente para obtermos um bom

classificador. Na tabela 5.4 temos alguns resultados que mostram o índice de acerto numa classificação de uma-classe (detalhes nos próximos capítulos). Os valores vão de zero a um, sendo um o melhor resultado possível; nesta tabela o classificador treinado é testado contra outras classes além das amostras da mesma classe não utilizadas no processo de treinamento. 3-Fold significa que as amostras de treinamento foram divididas em dois grupos, um usado no treinamento e outro (Train2) usado para teste.

O treinamento do classificador foi realizado tendo como base uma classe (ela está indicada na primeira coluna na tabela 5.4). Após o treinamento, amostras de teste e de treinamento (amostras marcadas como de treinamento na base de dados original da Reuters-21578) não usadas no treinamento além de amostras de outras classes foram utilizadas para testar o classificador.

Percebe-se que há alto índice de acerto, mas também em alguns casos, alto índice de falso positivo. O treinamento deste classificador foi realizado unicamente levando-se em consideração as palavras de maior ocorrência na classe de treino.

**Tabela 5.4. Resultados da classificação de uma-classe usando a base de dados da Reuters.**

| | **Earn** | **Acq** | **Money-Fx** | **Grain** | **Crude** | **Crude Test** | **Crude Train2** |
|---|---|---|---|---|---|---|---|
| **Crude 3-Fold** | 0,358 | 0,863 | 0,7534 | 0,625 | 0,9111 | 0,9115 | 0,9407 |
| | **Earn** | **Earn Test** | **Earn Train2** | **Acq** | **Money-fx** | **Grain** | **Crude** |
| **Earn 3-Fold** | 0,931 | 0,779 | 0,701 | 0,1333 | 0 | 0 | 0,1579 |
| | **Earn** | **Acq** | **Money-fx** | **Money-fx Test** | **Money-fx Train2** | **Grain** | **Crude** |
| **Money-fx 3-Fold** | 0,069 | 0,243 | 0,9438 | 0,8696 | 0,9101 | 0,48 | 0,5 |
| | **Earn** | **Acq** | **Acq Test** | **Acq Train2** | **Money-fx** | **Grain** | **Crude** |
| **Acquisition 3-Fold** | 0,419 | 0,917 | 0,9026 | 0,9136 | 0,2791 | 0,0313 | 0,4444 |
| | **Earn** | **Acq** | **Money-fx** | **Grain** | **Grain Test** | **Grain Train2** | **Crude** |
| **Grain 3-Fold** | 0,194 | 0,333 | 0,2308 | 0,9329 | 0,8571 | 0,92 | 0,82 |

A partir dos dados das Tabelas 5.3 e 5.4 podemos verificar que os documentos de uma mesma classe apresentam características que as diferem das outras, mas que uma simples classificação baseada nas palavras de maior ocorrência não é suficiente.

Algumas classes apresentaram resultados bons. Por exemplo, quando o SVM foi treinado tendo como a classe referência a classe "Earn" (segunda linha na Tabela 5.4), os testes realizados com amostras da mesma classe tiveram índices de acerto elevado (Earn-Test: 77,9%, Earn-Train2: 70,1%). Já as classes "Money-fx" e "Grain" praticamente não tiveram amostras classificadas como sendo da classe "Earn", o que é esperado para um classificador de uma-classe. Finalmente, as duas outras classes, "Acq" e "Crude" tiveram respectivamente, 13,33% e 15,79% das amostras classificadas como sendo da classe "Earn".

Por outro lado, os resultados da classe "Crude" foram ruins (primeira linha na Tabela 5.4). Mais de 86% das amostras da classe "Acq" foram classificadas como sendo da classe "Crude".

Analisando-se duas classes que apresentaram resultados ruins, essas classes são identificadas nessa base pública como "***crude***" e "***grain***", pode-se chegar à seguinte conclusão. A técnica utilizada nesta proposta criava um modelo vetorial ($VS_{crude}$) que representa um sub-conjunto do VS completo.

$VS_{crude}$ contém as palavras que apareceram nas amostras de treinamento da classe Crude. Essa classe foi utilizada como a classe referencia para o treinamento de uma instância do SVC, ou seja, treinamos o $SVC_{crude}$ para identificar os documentos que pertencem à classe Crude. Após o treinamento confrontamos esse classificador de uma-classe com as amostras das outras classes, conseguindo assim obter uma taxa de acerto e erro.

A restrição das palavras, ou melhor, a seleção das palavras da classe Crude para compor o $VS_{crude}$, foi realizada para limitar o tamanho da dimensão dos atributos, em vez de $|VS|$, trabalhamos com $|VS_{crude}|$, em que

$$|VS| >> |VS_{crude}|$$

Usando-se $VS_{crude}$, muitos textos das outras classes que não apresentavam nenhuma palavra de $VS_{crude}$ foram eliminados logo no começo. O restante dos textos apresentava característica

comum aos textos da classe *Crude*. Assim esses textos remanescentes que não pertencem à classe *Crude* foram identificados como pertencentes a ela.

A conclusão que se chegou foi que não basta utilizar as palavras mais freqüentes numa classe, as palavras precisam ser discriminantes. Essas palavras, ao contrário das outras selecionadas acima pelo critério de maior ocorrência entre os documentos da mesma classe, no caso, o "Crude", podem ocorrer com muito pouca frequência entre textos distintos de uma mesma classe, no entanto, elas ocorrem em frequência muito menor nas outras classes. Ou seja, uma palavra discriminante não precisa aparecer várias vezes, basta ter a capacidade de discriminar documentos da classe e documentos que não pertencem à classe.

Assim, o $VS_{crude}$ ideal é formado pelas palavras de maior ocorrência dentro da mesma classe e pelas palavras que mais caracterizam a classe.

No entanto, resta ainda um problema. Podemos ter criado um novo subconjunto de VS, o $VS_{crude}$ tão esparso quanto o conjunto original (levando-se em consideração a proporção no número de palavras entre os dois conjuntos). Então é necessário mais um refinamento. Esse refinamento será realizado usando-se a Informação Mútua e o Ganho de Informação (PEDERSEN, 1997; CAROPRESO; MATWIN; SEBASTIANI, 2001).

## 5.4 Agrupamento de Textos e Atributos

Bekkerman et al. (BEKKERMAN et al., 2003) realizam um extenso estudo na classificação de documento usando SVM e agrupamento distribucional das palavras. Essa representação de agrupamento de palavra é calculada usando-se um método conhecido como "gargalo" da informação (*information bottleneck*) que gera uma representação compacta e concisa do documento.

Os dados experimentais de Bekkerman são comparados à tradicional abordagem *bag-of-words* (BOW). As bases de dados utilizadas são três:

- 20 Newsgroups
- Reuters-21578
- WebKB

Diversos trabalhos (DUMAIS et al., 1998; WEISS et al., 1999; JOACHIMS, 1997) em classificação de texto têm sido realizados utilizando a representação BOW dos documentos, com bons resultados. No entanto, a classificação baseada em agrupamento de palavras tem apresentado resultados superiores ao BOW (PEREIRA; TISHBY; LEE, 1993) dependendo do tipo de conjunto de dados.

Agrupamento de palavras apresentou resultados superiores ao BOW para o conjunto de dados "20 Newsgroups", no qual a classificação dos documentos é mais complexa, ou seja, os termos de baixa ocorrência têm um peso importante para a classificação. O mesmo não ocorre com o conjunto de dados da Reuters e WebKB, no qual a estrutura é mais simples, ou seja, poucas palavras são suficientes para atingir um grau de classificação elevado.

O esquema de agrupamento distribucional dos seguintes autores citados (PEREIRA; TISHBY; LEE, 1993) consiste em agrupar palavras que representam uma distribuição sobre as categorias dos documentos no qual elas aparecem. Dado um conjunto de categorias $C = \{c_i\}_{i=1}^{m}$, uma distribuição de palavra wo sobre as categorias pode ser descrita por:

$$\{P(c_i \mid wo)\}_{i=1}^{m} \qquad \textbf{Eq. 5.6}$$

As palavras são então agrupadas usando-se o método de agrupamento aglomerativo. O método de agrupamento distribucional de Pereira (PEREIRA; TISHBY; LEE, 1993) é um caso particular de uma forma mais genérica de outros estudos (TISHBY; PEREIRA; BIALEK, 1999).

Hiroyama e Matsumoto (HIROYAMA; MATSUMOTO, 2002) realizam uma análise de agrupamento 2D para classificação de documentos. É realizada uma análise de agrupamento no nível dos documentos e, então, uma análise de agrupamento das palavras que aparecem nesses documentos. O objetivo é escolher as palavras mais adequadas para fins de classificação de documento.

Seus experimentos empíricos mostram que, mesmo entre os documentos que pertencem a uma mesma classe (foi utilizada a base pública da Reuters, a 21578) não existe uma distribuição uniforme das palavras. Não existe um conjunto de palavras que possa caracterizar todos os documentos da classe. Esses dados foram obtidos por meio do cálculo da divergência (a

divergência de Kullback-Leibler) da distribuição das palavras que ocorrem em cada uma das classes analisadas.

O experimento foi realizado da seguinte forma: dada uma classe A, ela foi subdividia em 4 grupos (aleatoriamente formada); assim, temos $A_1$, $A_2$, $A_3$, $A_4$,. Foi então realizada uma comparação da divergência KL entre $A_1$ e as outras classes de A e também com mais outras classes que não são A.

O que se esperava era um baixo valor da divergência entre $A_1$ e $A_{2,3,4}$ e um valor elevado entre $A_1$ e as outras classes. Isto não aconteceu, valores elevados da divergência KL foram observados entre algumas das classes de A. Assim, a condição de distribuição uniforme ou idêntica não pode ser utilizada; em outras palavras, não se pode escolher um conjunto de palavras de maior frequência esperando que seja o suficiente para discriminação entre as classes.

Em (HIROYAMA; MATSUMOTO, 2002) os autores trabalham com a co-ocorrência entre uma palavra wo e um documento d. A probabilidade de co-ocorrência entre eles pode ser expressa pela seguinte equação:

$$P(wo,d) = P(C_{wo},C_d)\cdot P(w \mid C_{wo})\cdot P(d \mid C_d) \qquad \textbf{Eq. 5.7}$$
$$wo \in C_{wo}, \quad d \in C_d$$

Onde $C_{wo}$ e $C_d$ representam o agrupamento a que wo e d pertencem.

Os parâmetros na equação acima são estimados pela estimativa de máxima verossimilhança:

$$P(C_{wo},C_d) = \frac{N(C_{wo},C_d)}{|S|}$$

$$P(wo \mid C_{wo}) = \frac{N(wo)}{N(C_{wo})} \qquad \textbf{Eq. 5.8}$$

$$P(d \mid C_d) = \frac{N(d)}{N(C_d)}$$

Onde N(x) representa a frequência de x, e |S| a quantidade total de pares (palavras wo, e documento d) de amostras.

## 5.5 Recursive Feature Elimination - RFE

O problema da quantidade de atributos a considerar no treinamento das máquinas de aprendizado, bem como a quantidade dos atributos que efetivamente contribuem na solução final do problema, têm sido tratados de diversas maneiras. Isabelle Guyon (GUYON et al, 2002; GUYON, ELISSEEFF, 2003; FURLANELLO et al., 2003; GENTILE, 2004; BEDO; SANDERSON; KOWALCZYK, 2006) aborda esse problema no contexto das SVMs.

Um SVM ao final do seu processo de otimização define um conjunto de amostras que compõem o chamado conjunto de vetor-suporte. Muitas vezes a quantidade dos vetores-suporte é em número muito elevado, lembrando que no caso da SVM essa quantidade influencia diretamente o tempo de resposta no processo de classificação. Soma-se a isso a dimensão de cada amostra, ou seja, a quantidade de atributos utilizados. Dependendo do problema, essa combinação prejudica o uso do SVM.

Para Guyon e colaboradores (GUYON et al, 2002), os atributos que menos contribuem para a solução final foram removidos sucessivamente ou recursivamente; então, este método recebe o nome de eliminação recursiva de atributos (*Recursive Feature Elimination* – RFE)

Dado um SVM cujo vetor peso é expresso por: $\mathbf{w}^* = \sum_{i \in SV} y_i \cdot \alpha_i^* \cdot \mathbf{x}_i$

Definindo uma função custo por: $J(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2$

E a sua variação por: $\delta J(i) = \frac{1}{2}\frac{\partial^2 J(\mathbf{w})}{\partial w_i^2}(\delta w_i)^2 = \frac{1}{2}(w_i)^2$

RFE visa eliminar atributos que causem a menor variação na função custo. Seu algoritmo pode ser descrito da seguinte forma.

***# lista de atributos ordenados***
F={};

***# complemento de F***
R={1, 2, 3, .. , n};

01. Treinar SVM normalmente usando o conjunto de amostras com R de atributos
02. Calcula-se a variação na função custo para cada atributo, $\delta J(i) \qquad \forall i = 1,...,n$
03. Remove-se atributo com menor variação e o coloca no conjunto F
04. Repetem-se as etapas 01, 02 e 03 até que R={}

No algoritmo acima, ao final do processo, F contém uma relação ordenada dos atributos que contribuem mais ou menos para a solução final do SVM. Usando-se RFE para o problema da classificação e seleção de gene, foi possível reduzir os atributos de cerca de 1000 para algumas dezenas (8 ou 20, dependendo dos tipos de amostra), provando ser esta uma técnica eficiente.

RFE apresenta um problema - a necessidade de iterações sucessivas. Dependendo do tamanho do conjunto de treinamento e da dimensão das amostras o processo todo pode ser bem demorado. Assim, RFE foi aperfeiçoado (FURLANELLO et al., 2003) para remover mais de 1 atributo por vez baseado na entropia associada aos pesos das amostras.

A idéia básica é eliminar os pesos de menor valor. Utilizando-se o cálculo da entropia como medida de concentração, pode-se eliminar mais de um atributo. RFE foi originalmente aplicado a problemas em bio-informática

# 6 Indexação das palavras usando correlação total ponto a ponto

A primeira contribuição deste trabalho está descrita nesta seção. A começar pela análise dos atributos a utilizar no contexto da classificação de documentos em que o vetor de atributos costuma ser esparso e de alta dimensão. Como já mencionado nos capítulos anteriores, uma boa análise de atributos é fundamental para criar um sistema robusto capaz de classificar documentos.

No Capítulo 5 foi apresentada a maneira como um documento pode ser representado em sua forma vetorial. A representação do documento por meio de um vetor de índices, em que cada índice representa uma palavra, é geralmente gerada pelas palavras com mais alto valor no resultado do produto entre TF e IDF. Ao contrário da abordagem tradicional, neste trabalho, as palavras que compõem o vetor de índices foram selecionadas segundo o valor da correlação total ponto a ponto – PTC (*pointwise total correlation*).

Foram calculadas as correlações totais ponto a ponto de um conjunto de palavras com relação à categoria à qual os documentos de onde as palavras foram extraídas pertenciam. Assim, estamos empregando uma nova forma de seleção de atributos em substituição ao tradicionalmente utilizado TF (*term frequency*) e TFIDF (IDF, *inverse document frequency*).

Note também que a nossa abordagem também se utiliza de palavras compostas ao invés das palavras simples. Queremos mostrar que existem padrões de palavras intrínsecas a uma categoria. Essa composição de palavras será apresentada nos próximos parágrafos.

Uma vez definidos os atributos (as palavras ou composição das palavras), precisamos selecionar aqueles que são mais relevantes para o processo de classificação, já que não pretendemos usar todas as palavras que compõem o vetor de índices; quanto maiores, mais custosos são os cálculos envolvidos nas etapas sucessivas do treinamento, teste e uso do classificador SVM.

À redução do vetor de índices (*feature selection*) dá-se o nome de redução de dimensionalidade. Redução de dimensionalidade é realizada por meio do mecanismo de seleção de atributos como, por exemplo, informação mútua (MI, *mutual information*), taxa de

ganho (GR, *gain ratio*) e estatística Chi-quadrada (DEBOLE; SEBASTIANI, 2003; YANG; PEDERSEN, 1997). Utilizamos neste trabalho a informação mútua e o ganho de informação (PEDERSEN, 1997; CAROPRESO; MATWIN; SEBASTIANI, 2001) porque elas têm apresentado bons resultados. O ganho da informação mede a redução da entropia (incerteza) de uma classe de teste baseada na observação de uma variável. O ganho mede o quanto um atributo é capaz de separar um conjunto de amostras em classes. A informação mútua mede a correlação existente entre variáveis e classes. Tanto o ganho da informação quanto a informação mútua são dependentes das distribuições de probabilidades que constituem a base do processo de comunicação.

Como base de dados foi utilizada a Reuters-21578 e a 20NG; são duas bases de dados pública, utilizada por pesquisadores na área de recuperação da informação e aprendizado de máquina.

## 6.1 Correlação total ponto-a-ponto - PTC

A escolha pela correlação total ponto a ponto na indexação dos termos foi conduzida pela idéia de que documentos que pertencem a uma mesma classe podem apresentar algum padrão em comum. Este padrão não é necessariamente um conjunto de palavras na sequência, mas podem existir palavras correlatas numa frequência de ocorrência que caracterizam os documentos pertencentes a uma mesma classe ou categoria, quando comparadas aos documentos de outras classes.

Na composição do vetor de índice, o modelo vetorial, geralmente usam-se palavras simples (e não palavras compostas como, por exemplo, ‘quebra-cabeça’). Acreditamos que cada categoria tenha alguns padrões de palavras intrínsecas a ela; por meio do cálculo da correlação total ponto a ponto pretendemos encontrar um conjunto de palavras não sequenciais que sejam representativas da categoria; em outras palavras, que sirvam para discriminar os documentos da categoria.

O cálculo da correlação total ponto a ponto foi realizado da seguinte forma: selecionou-se uma categoria para o estudo, lembrando que queremos realizar uma classificação de uma-classe. A seguir, o tamanho da janela de um contexto foi definido e, finalmente, a correlação total de cada composição de palavras foi calculada.

Um contexto é uma frase curta com N palavras. Em nosso trabalho, usamos cinco palavras; esse tamanho foi definido experimentalmente, levando-se em consideração a capacidade computacional em uso e o valor da correlação obtida. Quanto maior o valor de N, mais combinações de palavras existirão, e a chance de se obter uma repetição dessas composições também diminui, ou seja, quanto maior o tamanho de N mais cálculos serão feitos para obter muitas composições de palavras com baixo índice de correlação.

A extração de atributos baseada no PTC difere dos atributos (*feature*) de duas palavras (*bigrams*) e de três palavras (*trigrams*), tradicionalmente usados no sentido de que as palavras compostas obtidas no PTC não estão necessariamente na sequência em que aparecem no documento.

O uso somente das palavras simples não é apropriado para a classificação de texto; veja nos resultados da Tabela 5.4 o alto índice de falso positivo. Diversas palavras geralmente aparecem em vários documentos tornando as muito comuns em processo discriminatório. Por outro lado, palavras raras podem também não ser úteis por aparecerem em um conjunto muito reduzido de documentos da classe, e também por aparecerem com muita raridade em algumas outras classes.

Neste trabalho, combinamos um conjunto de palavras mantendo a sua ordem relativa, ou seja, se dentro de um contexto a palavra A veio antes de B, essa ordem relativa é mantida. Mas note que B não necessariamente vem logo depois de A.

Por exemplo, podemos ter: “A casa pato torta B” - o contexto é composto por cinco palavras, sendo que “A B” é uma das combinações que podemos formar.

A correlação total ponto a ponto usada para medir a co-ocorrência entre a palavra $wo_i$ dentro de um contexto $c_j$ pode ser expressa pela Equação 6.1. PTC é uma generalização da informação mútua ponto a ponto.

$$PTC(x, y, z \mid c) = \log\left( \frac{P(x, y, z \mid c)}{P(x \mid c) \cdot P(y \mid c) \cdot P(z \mid c)} \right) \qquad \textbf{Eq. 6.1}$$

onde $PTC(x, y, z \mid c_j)$ é a probabilidade de a palavra composta $wo_i$ (combinação das palavras x, y e z) co-ocorrer dentro do contexto c. Em outras palavras, ela representa o grau de co-ocorrência.

Com isso, temos um conjunto de combinações de palavras ordenadas (decrescente) segundo o seu grau de co-ocorrência. Agora precisamos descobrir quais dessas composições são mais relevantes para o processo de classificação de texto e, para tanto, nos utilizamos da informação mútua.

A informação mútua (MI) é zero se as variáveis são independentes entre si. Na Equação 6.2 nós mostramos como calcular a informação mútua entre uma palavra wo e uma categoria c. A equação da informação mútua é uma razão entre a probabilidade conjunta de wo e c com o produto das probabilidades de wo e c.

$$MI(wo, c) = \log_2\left(\frac{P(wo \wedge c)}{P(wo) \cdot P(c)}\right) \qquad \textbf{Eq. 6.2}$$

Seja A a quantidade de ocorrência da palavra wo e c, B a quantidade de ocorrência da palavra wo mas não c, C a quantidade de ocorrência da classe c mas não wo, e N o número total de documentos no conjunto de treinamento. Então, a informação mútua entre wo e c pode ser expressa pela Equação 6.3:

$$MI(w, c) = \log_2\left(\frac{A \times N}{(A + C) \times (A + B)}\right) \qquad \textbf{Eq. 6.3}$$

Nossa proposta visa a destacar as características mais comuns da classe e, ao mesmo tempo, também às que melhor diferenciam os documentos de uma classe com relação aos demais. A seguir, na próxima seção, a nossa abordagem uma-classe da classificação binária será apresentada. Usamos a informação mútua e a correlação total ponto a ponto apresentada anteriormente.

## 6.2 A abordagem uma-classe

Conforme já comentado no Capítulo 2, o SVM foi inicialmente criado para resolver problemas de classificação binária ou duas classes (uma positiva e outra negativa). Já a classificação de uma-classe tem por finalidade identificar amostras que não pertençam ao

conjunto, também conhecidas como ‘novidade’. Estas constituem amostras novas, não vistas até então durante o treinamento do SVM (SCHÖLKOPF et al., 1999; ZHUANG; DAI, 2006).

Na classificação uma-classe, deve-se selecionar os atributos da mesma forma como se realiza numa classificação binária normal. Imaginemos que em vez de classificar documentos tenhamos que classificar cobras, separando-as em venenosas e não venenosas.

Suponhamos também que a cor seja o fator decisivo, quanto mais quente for a cor (vermelho, amarelo, preto) mais venenosa será. O critério baseado no atributo cor é verdadeiro enquanto não se descubra nenhuma outra cobra de cor vermelha amarela não venenosa; em outras palavras, a classificação estará restrita ao domínio conhecido.

Essa restrição ao desconhecimento do universo completo é um fator negativo da classificação uma-classe. Ou seja, não há garantias de que o classificador treinado funcione bem até que comecemos a obter muitos falsos positivos devido à escolha de atributos não discriminatórios.

Na classificação uma-classe de documentos, o objetivo será definir superfícies de separação ou contorno que informem se um documento pertence ou não à classe. Isso será feito tendo como ponto de partida os atributos selecionados que servirão para caracterizar os documentos da classe.

A Tabela 5.4 do Capítulo 5 mostra os resultados da classificação uma-classe. Podem-se notar falsos positivos elevados em muitas classes. Esses resultados levaram a pensar numa nova abordagem para a classificação uma-classe, usamos um classificador binário com um vetor de índices localizado.

Em nossa abordagem, nós não criamos um único modelo vetorial (modelo vetorial global), como feito geralmente, mas sim um modelo vetorial para cada categoria; dessa forma, conseguimos um vetor de índices que representa melhor cada categoria.

Outra vantagem em usar um modelo vetorial específico para uma-classe está no fato de criarmos um vetor de dimensão muito menor quando comparado ao vetor completo. Assim, muitos documentos que não pertencem à classe são eliminados muito antes de serem testados.

## 6.3 Experimentos – classificação de uma-classe – Reuters-21578

Nessa classificação uma-classe, usamos a abordagem de Schölkopf et al. (SCHÖLKOPF et al., 1999), que sugeriram um método de classificação uma classe, adaptando o SVM tradicional.

Dado um conjunto de documentos pertencentes a uma mesma categoria ou classe, as palavras de maior ocorrência foram identificadas; dessas, as de maior ocorrência foram selecionadas em ordem decrescente até completar 20% do total de palavras que aparecem no conjunto todo.

Esse critério da escolha dos 20% das palavras de maior ocorrência foi estabelecido de forma empírica, mas tendo como ajuste o resultado final do índice de acerto e erro do classificador.

Nas tabelas 6.1, 6.2 e 6.3, apresentamos algumas palavras dos conjuntos que foram utilizados no treinamento.

Perceba nas tabelas 6.1, 6.2 e 6.3 que poucas palavras da classe representam a maioria das palavras na classe. Como já dito nos capítulos anteriores, essa informação não é condição suficiente para diferenciar as classes.

**Tabela 6.1. Categoria "earn" (à esquerda) e "acq" (à direita), obtida da Reuters-21578. Na coluna ocorrência está o total de ocorrências da palavra correspondente. A coluna "ranking" informa a posição da palavra entre aquelas que mais repetem.**

| Ocorrência | Palavra | Ranking |
|---|---|---|
| 5.630 | mln | 1 |
| 4.143 | ct | 2 |
| 3.543 | shr | 3 |
| 3.255 | dlr | 4 |
| 2.071 | loss | 5 |
| 1.778 | rev | 6 |
| 1.616 | net | 7 |
| 1.430 | profit | 8 |
| 1.369 | oper | 9 |
| 1.322 | year | 10 |
| 1.128 | shr ct | 11 |
| 1.074 | share | 12 |
| 1.014 | mln mln | 13 |
| 992 | sale | 14 |
| 986 | shr ct ct | 15 |
| 973 | billion | 16 |
| 891 | compani | 17 |
| 746 | mln dlr | 18 |
| 736 | rev mln mln | 19 |
| 604 | pct | 20 |

| Ocorrência | Palavra | Ranking |
|---|---|---|
| 1.624 | share | 1 |
| 1.523 | dlr | 2 |
| 1.389 | compani | 3 |
| 1.122 | mln | 4 |
| 1.012 | inc | 5 |
| 913 | pct | 6 |
| 755 | offer | 7 |
| 736 | corp | 8 |
| 624 | stock | 9 |
| 524 | which | 10 |
| 517 | group | 11 |
| 482 | mln dlr | 12 |
| 453 | acquir | 13 |
| 369 | sale | 14 |
| 364 | acquisit | 15 |
| 364 | bank | 16 |
| 351 | common | 17 |
| 350 | sharehold | 18 |
| 336 | year | 19 |
| 327 | agreem | 20 |

**Tabela 6.2 Categoria "crude" (a esquerda) e "grain" (direita), obtido da Reuters-21578.**

| Ocorrência | Palavra | Ranking |
|---|---|---|
| 984 | oil | 1 |
| 482 | mln | 2 |
| 441 | price | 3 |
| 416 | dlr | 4 |
| 344 | barrel | 5 |
| 301 | opec | 6 |
| 299 | pct | 7 |
| 272 | crude | 8 |
| 260 | year | 9 |
| 233 | bpd | 10 |
| 229 | product | 11 |
| 222 | compani | 12 |
| 161 | market | 13 |
| 158 | which | 14 |
| 151 | ga | 15 |
| 144 | billion | 16 |
| 134 | energi | 17 |
| 130 | offici | 18 |
| 122 | dai | 19 |
| 122 | been | 20 |

| Ocorrência | Palavra | Ranking |
|---|---|---|
| 639 | Tonn | 1 |
| 500 | Mln | 2 |
| 406 | Wheat | 3 |
| 297 | year | 4 |
| 294 | grain | 5 |
| 261 | export | 6 |
| 250 | corn | 7 |
| 242 | agricultur | 8 |
| 228 | pct | 9 |
| 223 | price | 10 |
| 192 | dlr | 11 |
| 170 | crop | 12 |
| 163 | soviet | 13 |
| 162 | nil | 14 |
| 156 | import | 15 |
| 153 | product | 16 |
| 152 | mln tonn | 17 |
| 152 | market | 18 |
| 152 | trade | 19 |
| 138 | depart | 20 |

**Tabela 6.3 Categoria "money-fx".**

| Ocorrência | Palavra | Ranking |
|---|---|---|
| 887 | bank | 1 |
| 609 | dollar | 2 |
| 549 | rate | 3 |
| 505 | market | 4 |
| 394 | pct | 5 |
| 328 | curren | 6 |
| 315 | trade | 7 |
| 300 | exchang | 8 |
| 252 | billion | 9 |
| 252 | mln | 10 |
| 241 | stg | 11 |
| 237 | yen | 12 |
| 207 | dealer | 13 |
| 204 | central | 14 |
| 201 | japan | 15 |
| 177 | foreign | 16 |
| 172 | year | 17 |
| 165 | todai | 18 |
| 165 | polici | 19 |
| 164 | around | 20 |

A Figura 6.1 mostra a distribuição das ocorrências das palavras da classe "earn". Perceba que a curva tem o formato esperado de frequência de ocorrência das palavras em um idioma. Este

fato permite a escolha de poucas palavras que se repetem com maior frequência; caso contrário, temos um total de mais de 60.000 palavras (simples e compostas) compondo o vetor de atributos.

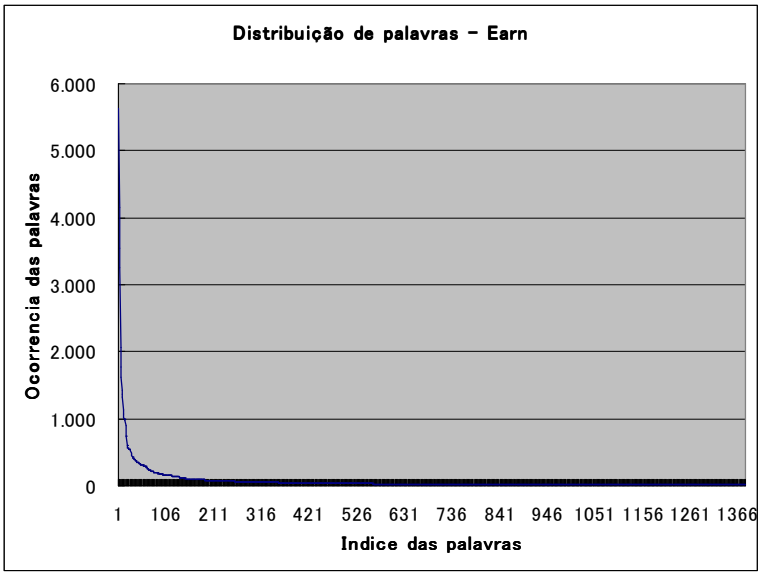



**Figura 6.1. Distribuição das ocorrências de cada palavra da categoria "earn"**

A tabela 6.4 mostra a divisão das amostras em treinamento e teste, além de mostrar uma não uniformidade na quantidade de documentos disponíveis para cada classe. Essa divisão em treinamento e teste já veio na base de dados da Reuters-21578, rotulada.

**Tabela 6.4. Quantidade de documentos divididos em amostras de treino e teste, segundo a divisão originalmente realizada e disponível na base da Reuters-21578.**

| Categoria | Treino | Teste |
|---|---|---|
| Earn | 2.709 | 1.044 |
| Acq | 1.488 | 643 |
| Money-fx | 460 | 141 |
| Grain | 394 | 134 |
| Crude | 349 | 161 |

Uma vez selecionadas as palavras de maior ocorrência e já observando a necessidade em "olhar" as outras classes, definimos um índice para escolher os melhores atributos.

A classificação uma-classe geralmente trabalha apenas com as amostras positivas. Numa classificação binária temos duas classes, a positiva e a negativa.

Imagine que queremos diferenciar laranja e limão. Escolhendo o atributo raio da fruta, podemos ter muitos falsos positivos e negativos, afinal existirão laranjas pequenas e limões grandes. Então, esse atributo não serve para caracterizar cada fruta. A escolha dos atributos tem o mesmo papel. Precisamos escolher aquela que melhor caracterize cada classe.

Para tanto, dadas as palavras que aparecem em uma categoria, foi realizado o cálculo da ocorrência nas outras classes em estudo. Seja i o índice para uma palavra *p* qualquer, seja $F_i$ a quantidade de ocorrência dessa palavra em classes distintas.

Por exemplo, sendo A categoria no qual queremos aplicar a classificação de uma classe, $p_i$ aparece nas classes B e C, mas não em D e F, temos então:

$$indice = \frac{2}{4} = 0{,}5$$

Quanto mais perto de 1, pior. A equação abaixo generaliza esse índice, a frequência inversa de categoria. Onde C representa todas as classes em estudo, menos a classe referência.

$$ICF(p_t) = \frac{F_t}{C} \qquad \textbf{Eq. 6.4}$$

Um argumento contra a técnica acima é de que no limite, ou seja, quanto mais classes houver, a chance de encontrarmos a palavra será maior, chegando a um ponto em que o índice torna-se constante para qualquer palavra.

## 6.4 Experimentos – classificação de uma-classe – 20 Newsgroup

Deste ponto em diante a seguinte representação será usada quando dados da base 20Newsgroup forem mencionados.

**Tabela 6.5 Nomenclatura usada para representação das classes da coleção 20NG**

| Nome da categoria | Representação/Nomenclatura |
|---|---|
| alt.atheism | Classe ou categoria 01 |
| comp.graphics | Classe ou categoria 02 |
| misc.forsale | Classe ou categoria 07 |
| rec.sport.baseball | Classe ou categoria 10 |
| sci.med | Classe ou categoria 14 |
| talk.politics.mideast | Classe ou categoria 18 |

Os experimentos com a base de dados 20Newsgroup foram conduzidos utilizando-se a representação SMART descrito no capítulo anterior. Essa representação é uma variação da composição TFxIDF que atribui um peso a um atributo.

A tabela 6.6 mostra resultados (valores médios de F1) dos três melhores resultados de cada classe. No Apêndice E podem-se ver os dados completos consolidados aqui. A tabela 6.6 mostra que a representação ATN é a que fornece maior acurácia em termos da métrica F1.

**Tabela 6.6 Classe 01 – Comparativo entre três representações dos documentos. Média e desvio do valor de F1.**

| | 2 | 7 | 10 | 14 | 18 |
|---|---|---|---|---|---|
| **ATN** | 0,9230 ± 0,0033 | 0,9025 ± 0,0128 | 0,8849 ± 0,0048 | 0,8920 ± 0,0022 | 0,8722 ± 0,0025 |
| **NTC** | 0,9162 ± 0,0045 | 0,878 ± 0,0043 | 0,8725 ± 0,0054 | 0,8584 ± 0,0096 | 0,8337 ± 0,0170 |
| **NNN** | 0,8749 ± 0,0099 | 0,8811 ± 0,0043 | 0,8549 ± 0,0011 | 0,8613 ± 0,0021 | 0,7932 0,0017 |

Os dados da tabela 6.7 mostram que as representações ATN e NTC superam a representação NNN. Fica difícil decidir entre as representações ATN e NTC.

**Tabela 6.7 Classe 02 – Comparativo entre três representações dos documentos. Média e desvio do valor de F1.**

| | 1 | 7 | 10 | 14 | 18 |
|---|---|---|---|---|---|
| **ATN** | 0,8512 ± 0,0026 | 0,8385 ± 0,0083 | 0,9141 ± 0,0046 | 0,8909 ± 0,0043 | 0,8888 ± 0,0070 |
| **NTC** | 0,8794 ± 0,0032 | 0,8398 ± 0,0083 | 0,8791 ± 0,0032 | 0,8648 ± 0,0077 | 0,8648 ± 0,0058 |
| **NNN** | 0,8526 ± 0,0191 | 0,8340 ± 0,0060 | 0,8397 ± 0,0107 | 0,8262 ± 0,0200 | 0,8594 ± 0,0155 |

A tabela 6.8 apresenta uma pequena vantagem para a representação ATN sobre a representação NTC.

**Tabela 6.8 Classe 10 – Comparativo entre três representações dos documentos. Média e desvio do valor de F1.**

| | 1 | 2 | 7 | 14 | 18 |
|---|---|---|---|---|---|
| **ATN** | 0,8591 ± 0,0024 | 0,9232 ± 0,0043 | 0,8916 ± 0,0133 | 0,8890 ± 0,0027 | 0,8951 ± 0,0138 |
| **NTC** | 0,8537 ± 0,0069 | 0,9140 ± 0,0044 | 0,8233 ± 0,0212 | 0,8396 ± 0,0090 | 0,8612 ± 0,0125 |
| **NNN** | 0,8227 ± 0,0075 | 0,8562 ± 0,0039 | 0,8113 ± 0,0072 | 0,8146 ± 0,0028 | 0,8147 ± 0,0223 |

A tabela 6.9 mostra uma difícil decisão quanto a escolha da representação ATN contra a NTC.

**Tabela 6.9 Classe 18 – Comparativo entre três representações dos documentos. Média e desvio do valor de F1.**

| | 1 | 2 | 7 | 10 | 14 |
|---|---|---|---|---|---|
| **ATN** | 0,8279 ± 0,0152 | 0,9226 ± 0,0023 | 0,8558 ± 0,0107 | 0,8812 ± 0,0108 | 0,9016 ± 0,0021 |
| **NTC** | 0,7915 ± 0,0014 | 0,9204 ± 0,0103 | 0,8493 ± 0,0162 | 0,8777 ± 0,0009 | 0,8915 ± 0,0046 |
| **NNN** | 0,7424 ± 0,0153 | 0,8993 ± 0,0030 | 0,8207 ± 0,0046 | 0,8322 ± 0,0135 | 0,8855 ± 0,0072 |

Analisando as tabelas 6.6 a 6.9, decidiu-se pela escolha da representação ATN. Tanto ATN quanto NTC mostram bons resultados para representar um documento. A representação ATN não possui normalização em seu cálculo, isso também contribuiu por sua escolha.

### 6.4.1 Resultados - 20Newsgroup – Pointwise-total-correlation

Nesta seção os resultados do Pointwise-total-correlation serão apresentados. A base de dados utilizada foi a 20Newsgroup e os documentos foram representados através da notação ATN.

As tabelas 6.10 a 6.13 mostram as medidas F1 obtidas para as classes selecionadas. A notação **SW** significa que os atributos são compostos por palavras simples, enquanto que a notação **PTC** significa que os atributos são compostos por composição de palavras.

Dados da tabela 6.10 mostram resultados do treinamento da classe 01 contra as classes 18, 14, 10, 07 e 02. Veja na Seção 6.4 o significado de cada uma dessas classes. Tanto a representação através das palavras simples, quanto a representação através das palavras compostas mostraram resultados equivalentes dentro da margem de erro.

**Tabela 6.10 Classe-01. Palavra-simples vs PTC. Medidas F1.**

| Classe (-) | SW | PTC |
|---|---|---|
| **18** | 0,8875 ± 0,013 | 0,8750 ± 0,015 |
| **14** | 0,9082 ±0,012 | 0,8938 ± 0,016 |
| **10** | 0,9264 ±0,015 | 0,8904 ± 0,014 |
| **7** | 0,9261 ±0,012 | 0,9149 ± 0,015 |
| **2** | 0,9450 ±0,011 | 0,9260 ± 0,013 |

A tabela 6.11 mostra uma pequena vantagem para a representação simples. Mesmo assim, no geral as duas representações são equivalentes.

**Tabela 6.11 Classe-10. Palavra-simples vs PTC. Medidas F1.**

| Classe (-) | SW | PTC |
|---|---|---|
| **18** | 0,9268 ± 0,011 | 0,9110 ± 0,012 |
| **14** | 0,9291 ± 0,013 | 0,8920 ± 0,013 |
| **7** | 0,9406 ± 0,012 | 0,9065 ± 0,012 |
| **2** | 0,9451 ± 0,015 | 0,9281 ± 0,015 |
| **1** | 0,8883 ± 0,016 | 0,8698 ± 0,014 |

Os dados das tabelas 6.12 e 6.13 mostram também uma equivalência para as duas representações das palavras. Em alguns casos, a representação PTC é superior, mas dentro da margem de erro.

**Tabela 6.12 Classe-14. Palavra-simples vs PTC. Medidas F1.**

| Classe (-) | SW | PTC |
|---|---|---|
| **18** | 0,9513 ± 0,015 | 0,8862 ± 0,016 |
| **10** | 0,9373 ± 0,016 | 0,9167 ±0,013 |
| **7** | 0,9057 ± 0,014 | 0,8987 ±0,014 |
| **2** | 0,9142 ± 0,013 | 0,9119 ±0,015 |
| **1** | 0,8898 ± 0,012 | 0,9038 ±0,011 |

**Tabela 6.13 Classe-18. Palavra-simples vs PTC. Medidas F1.**

| Classe (-) | SW | PTC |
|---|---|---|
| **14** | 0,9088 ±0,011 | 0,8980 ±0,015 |
| **10** | 0,9141 ±0,012 | 0,8901 ±0,012 |
| **7** | 0,8521 ±0,011 | 0,8673 ±0,014 |
| **2** | 0,9262 ±0,013 | 0,9246 ±0,013 |
| **1** | 0,7917 ±0,011 | 0,8452 ±0,014 |

A representação das palavras compostas não mostrou ser superior à representação das palavras simples. No entanto, as palavras compostas trazem consigo uma nova interpretação da classe. As palavras relacionadas umas com as outras ficam evidentes no PTC, esse relacionamento é muito importante para uma análise semântica dos textos.

A seguir, no próximo capítulo, iremos apresentar a segunda proposta deste trabalho, o SVM com alteração em seu algoritmo de atualização dos coeficientes de Lagrange durante as iterações no processo de treinamento.

# 7 SVM baseado na atualização dos coeficientes de Lagrange vizinhos - NSVM

A segunda contribuição deste trabalho está na otimização no processo de aprendizado do SVM; introduzimos o conceito de atualização dos vetores-suporte vizinho para diminuir o processo iterativo durante o treinamento do SVM. Introduziremos as alterações propostas no algoritmo SMO modificado por Keerthi; veja a Figura 7.1.

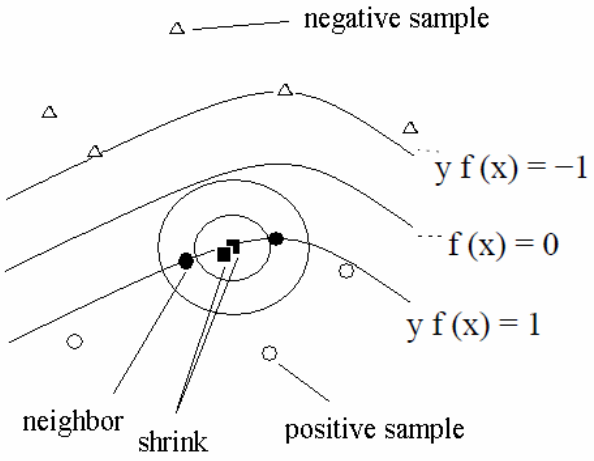



**Figura 7.1 Os quadrados preenchidos ($x_2$) representam as amostras selecionadas na otimização conjunta com $x_1$ durante o treinamento do SMO. Esses dois quadrados, na verdade, serão unificados; o processo de junção (*shrinking*) irá eliminar um deles. As amostras na vizinhança de $x_2$ estão representadas como círculo preenchido.**

Consideramos um subconjunto de vetores candidatos na vizinhança da amostra $x_j$ selecionada na iteração interna (passo 2 do algoritmo 1, veja também a Figura 7.1). Essa vizinhança é definida usando alguma métrica de distância calculada uma única vez, no início do processo de treinamento. O objetivo do cálculo das distâncias não é encontrar agrupamentos (como realizado na rede neural *Self-Organizing Map* - SOM de Teuvo Kohonen) ou determinar uma quantidade finita de agrupamentos (como no caso de uma *K-Nearest-Neighbor* - KNN); a ideia principal nessa etapa do processo de treinamento é unir amostras similares, ou seja, amostras que são vizinhas, e também "fundir" algumas amostras quando a distância Euclidiana entre elas for bem pequena.

Reduzindo o número de amostras de treinamento, o tempo de treinamento do SVM se reduz não apenas porque haverá menos amostras, mas principalmente porque haverá menos cálculo da função *kernel*. Reduzindo as amostras redundantes, estaremos evitando a sua seleção,

evitando assim cálculos desnecessários já que elas (amostras bem próximas uma das outras) não trariam ganhos como vetores-suporte.

Medidas de distância como Coseno, Jaccard e Tanimoto foram consideradas, mas, como os resultados foram muito ruins, logo no início as descartamos e utilizamos a distância Euclidiana.

A escolha da distância Euclidiana se deve ao fato de permitir tratar de forma eficiente vetores esparsos, que são utilizados neste trabalho. Representamos os documentos como vetor de palavras; esse vetor de palavras (modelo vetorial, *vector-space*) contém todas as possíveis palavras presentes nos documentos que compõem o conjunto de treinamento, menos as chamadas *stop words*.

A estratégia de vizinhança foi usada no processo de atualização dos coeficientes de Lagrange de um SMO. Uma das etapas que consomem mais tempo no processo de treinamento de uma SMO está na escolha das amostras candidatas que compõem o conjunto de trabalho (*working-set*). Assim, podemos definir o seguinte teorema:

**Teorema 1**: Amostras que são vizinhas marginais dos vetores-suporte são também candidatas a vetores-suporte.

**Prova**: Os vetores-suporte são elementos ou amostras que estão localizados na fronteira do seu conjunto (por exemplo, conjunto das amostras positivas, veja a Figura 7.1). Eles estão separados de uma distância (margem) do conjunto negativo. Uma amostra próxima ao vetor-suporte que não esteja no interior do conjunto, mas próxima à fronteira deste conjunto, também é candidata a vetor-suporte.

A seleção das amostras baseada no par que mais viola a condição KKT (MVP, *maximal violating pair*) (CHEN; FAN; LIN, 2006) diz o seguinte:
Selecione,

$$\begin{aligned} i &\in \arg\max_{t \in I_{up}(\alpha^k)} \; -y_t \nabla f(\alpha^k)_t \,, \\ j &\in \arg\max_{t \in I_{low}(\alpha^k)} \; -y_t \nabla f(\alpha^k)_t \end{aligned} \qquad \textbf{Eq. 7.1}$$

Onde:

$$I_{up}(\alpha) \equiv \{t \mid \alpha_t < C, \quad y_t = +1 \quad or \quad \alpha_t > 0, \quad y_t = -1\}$$

e **Eq. 7.2**

$$I_{low}(\alpha) \equiv \{t \mid \alpha_t < C, \quad y_t = -1 \quad or \quad \alpha_t > 0, \quad y_t = +1\}$$

Retorne B={i,j}

*f(α)* é a função objetiva *primal* do problema de otimização quadrática do SVM, e y é a identificação (alvo) da classe à qual pertence cada amostra, $y \in [-1,+1]$

Seja $x_k$ uma amostra vizinha de qualquer elemento em B. Se $x_k$ for vizinha de B então $y_{tk}=y_{tB}$ porque em nosso algoritmo calculamos apenas a vizinhança das amostras da mesma classe. Se $x_k$ não for uma vizinha de borda ou marginal, então $\alpha_k<C$. Assim, se $x_k$ for vizinha de $x_B$ e $x_k$ for uma vizinha marginal, então elas possuem o mesmo alvo e $\alpha_t>0$.

Se $x_1$ (amostra selecionada da iteração externa) for $x_i$ e $x_2$ for $x_j$ e $x_k$ também pertencer a $I_{low}$. Nós ainda teremos um novo *working set* B={i,k} que é um MVP.

Assim, durante o processo de atualização dos coeficientes das amostras do SMO, teremos um ganho no processo de treinamento como um todo se atualizarmos os coeficientes próximos àquele selecionada. A probabilidade de que essas amostras sejam vetores-suporte é muito maior do que a probabilidade de qualquer outra amostra ainda não vista.

A função de vizinhança não irá mudar no tempo, e ela é expressa por:

$$n(A,B) = e^{\frac{-d(A,B)}{2\sigma^2}} \qquad \textbf{Eq. 7.3}$$

Onde d(A,B) representa a distância Euclidiana e σ é um parâmetro variável que pode ser ajustado no início do treinamento, uma única vez. A Equação 7.3 proposta neste trabalho foi inspirada na função de vizinhança da rede de Kohonen, a SOM (Self-Organizing Map).

Note que, no entanto, todas as amostras candidatas aos vetores-suporte devem satisfazer à condição KKT. A Figura 7.1 mostra algumas amostras que são candidatas aos vetores-suporte (círculos preenchidos) e algumas outras que não são candidatas (círculo sem preenchimento). Na Figura 7.1, a amostra cujo coeficiente está sendo atualizado está representada por um quadrado.

O algoritmo proposto não irá melhorar o treinamento para sempre, haverá um momento durante o processo de treinamento em que não precisaremos mais atualizar os coeficientes da vizinhança. Imagine, por exemplo, que algum coeficiente da amostra vizinha já tenha sido zerado (inativo) ou tenha superado o limite superior (*bounded*). Não ganharemos nada tentando atualizá-lo porque sabemos que as condições KKT não serão satisfeitas.

Assim, precisamos identificar um momento a partir do qual devemos parar a atualização dos coeficientes das amostras vizinhas. O algoritmo 2 mostra esse processo.

| **Algoritmo 2**: Algoritmo de atualização de vizinhança | |
|---|---|
| 1 | **Inicialização** |
| | Encontre o vizinho de cada amostra |
| 2 | **Processo de atualização SMO** |
| 2.1 | Atualize $\alpha_1$, $\alpha_2$ |
| 2.2 | For each $\alpha_2$'s neighbor |
| 2.3 | $\alpha_{2j} \leftarrow$ coeff. of jth neighbor of sample $x_2$ |
| 2.4 | $\alpha'_{2j} \leftarrow$ coeff. $\alpha_{2j}$ after update * |
| 2.5 | $\Delta\alpha = \frac{\left\|\alpha'_{2j} - \alpha_{2j}\right\|}{\left\|\alpha_{2j}\right\|}$ |
| 2.6 | If $\Delta\alpha < \varepsilon_n$ |
| | Remove $\alpha_{2j}$ as the neighbor of $\alpha_2$ |

No algoritmo 2, nós verificamos o processo de atualização dos coeficientes das amostras de treinamento. No passo 2.4 não calculamos nenhum valor novo de $\alpha_2$ se não for necessário, ou seja, se não houver violação das condições KKT. Foi também estabelecido um valor limiar $\varepsilon_n$, ele é geralmente bem pequeno (0.001).

Outro aperfeiçoamento feito no algoritmo padrão do SMO de Keerthi está no *cache* (termo inglês que significa memória para armazenamento temporário dos dados) do produto *kernel*, em vez de armazenar o erro, como feito normalmente. O armazenamento do erro (veja as equações 2.32 e 2.33 no Capítulo 2) usa o produto escalar do *kernel* que permanecerá o mesmo durante todo o processo de treinamento do SMO. Essa pequena observação permitiu um ganho notável na redução do tempo de treinamento.

## 7.1 Experimentos com NSMO

Na Tabela 7.1 mostramos alguns resultados da escolha das amostras candidatas baseadas na vizinhança (NSMO). A estratégia de vizinhança foi aplicada sobre o SMO de Keerthi.

Comparações foram feitas com a base de dados da Reuters e também com a base de dados binária, cuja distribuição lembra um tabuleiro de xadrez (*chessboard dataset*); veja a Figura 8.3 (do próximo capítulo). Essa base de dados é sintética e muito usada nas aferições de resultados de SVM.

Usando a base de dados xadrez alcançamos uma melhora de 16% no tempo de treinamento. À medida que o tamanho da base cresce, mais evidente fica o ganho do NSMO. Usando a base de dados da Reuters-21578, o ganho foi um pouco menor; mesmo assim, foi significante. Os dados da Tabela 7.1 foram obtidos em cima da classe dinheiro.

**Tabela 7.1 NSMO ganhos em tempo de treinamento comparando com o SMO de Keerthi. Base de dados de Reuters, a Reuters-21578. Foi realizado um treinamento 2-fold com 5 repetições.**

| Class | Keerthi SMO (s) | Keerthi NSMO (s) | Ganho |
|---|---|---|---|
| **Money vs Grain** | 270,17 ± 6,83 | 255,07 ± 3,96 | 5,59% |
| **Money vs Acq** | 307,4 ± 3,12 | 287,3 ± 2,31 | 6,54% |
| **Money vs Crude** | 166,11 ± 4,12 | 151,81 ± 3,17 | 8,55% |
| **Money vs Earn** | 595,5 ± 4,12 | 566,3 ± 3,87 | 5,16% |

A Figura 7.2 mostra alguns resultados comparativos do NSMO, uma versão otimizada da versão SMO de Keerthi. A linha superior com triângulos representa o tempo de treinamento do SMO de Keerthi (chamemos de KSMO), enquanto que a linha abaixo, com quadrados, é o nosso NSMO.

A base de dados utilizada foi a *chessboard* com 130 amostras no início. Foram então acrescidas amostras, artificialmente e aleatoriamente, nas proximidades de cada uma dessas 130 amostras.

À medida que o tamanho do conjunto aumenta, torna-se mais evidente a diferença entre os tempos de treinamento, entre o NSMO e o KSMO. Nesse teste foi utilizada uma função *kernel* RBF com $\gamma$ de 0.3, em um SVM do tipo *hard-margin*.

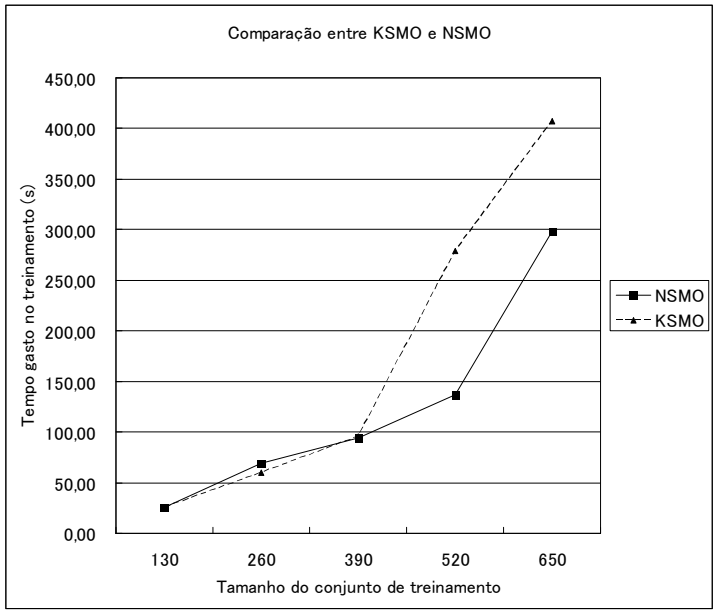



**Figura 7.2 Tempo de treinamento para o SMO de Keerthi (triângulo) e o nosso SMO com vizinhança - NSMO (quadrados) à medida que se aumenta o conjunto de treinamento. A base de dados usada foi a *chessboard*. NSMO tem um tempo de treinamento menor e essa diferença**

## 7.2 Experimentos - Classificação de uma-classe, usando classificadores binários

A Tabela 7.2 mostra os resultados obtidos na classificação binária com abordagem de uma-classe. Como comparação, temos os resultados de Joachims' (JOACHIMS. 2000) e Dumais à esquerda. Pode-se perceber que há uma melhora na performance para a classe "money-fx", uma diferença pequena para as classes "earn", "acq" e "crude", e uma piora expressiva para a classe "grain".

**Tabela 7.2. À esquerda, resultados da classificação de Joachims 2000, ao centro, Naïve Bayes, Bayes Net, Find Similar, Decision Tree e SVM de Dumais,e a direita os resultados da classificação usando a correlação total ponto-a-ponto com atualização da vizinhança (NSVM). Em itálico e fonte de texto menor, os resultados inferiores à nossa técnica e em negrito em fonte de texto maior, resultados que superam a nossa técnica.**

| | Joa00 | Naïve Bayes | Bayes Net | Find Similar | Decision Trees | SVM-Dumais | NSVM |
|---|---|---|---|---|---|---|---|
| **Earn** | 96.90 | 95,90 | 95,80 | *92,90* | 97,80 | **98,00** | 94.50 |
| **Acq** | 93.10 | *87,80* | *88,30* | *64,70* | 89,70 | **93,60** | 89.14 |
| **Crude** | 83.90 | *79,50* | *79,60* | *70,10* | 85,00 | **88,90** | 81.87 |
| **Grain** | 89.90 | *78,80* | 81,40 | *67,50* | *85,00* | **94,60** | 80.56 |
| **Money** | *80.90* | *56,60* | *56,80* | *46,70* | *66,20* | *74,50* | **82.91** |

As classes "grain" e "crude" são as duas que apresentam o menor volume de amostras para treinamento. Esse fato comprometeu um pouco os resultados; no entanto, os resultados

obtidos pelo nosso classificador, o NSVM, apresentou resultados melhores do que outras técnicas, como listado na Tabela 7.2.

Na Tabela 7.2 foi utilizado o NSVM (*neighbor* SVM), SVM modificado proposto neste trabalho porque, como mostraremos na Seção 8.3 NSVM, não se altera a sua capacidade de generalização quando comparado ao SVM que usamos como referência, o SMO de Keerthi.

## 7.3 Experimentos - Classificação usando base de dados 20Newsgroup

Comparação do NSMO e SMO. O SMO é o SMO alterado por Keerthi e o NSMO é uma versão modificada dessa mesma versão de Keerthi.

Pode-se verificar na tabela 7.3 que também há um ganho no tempo de treinamento, em muitos casos, bem expressivo. O importante a lembrar nos resultados obtidos com o NSVM é que a capacidade de generalização do SVM treinado praticamente não é afetada, mostraremos logo a seguir.

**Tabela 7.3 NSMO ganhos em tempo de treinamento comparando com o SMO de Keerthi. Base de dados 20-*Newsgroup*. Análise preliminar.**

| Classe | Keerthi SMO (s) | Keerthi NSMO (s) | Ganho |
|---|---|---|---|
| alt vs comp-graphics | 279.40 | 188.80 | 32,43% |
| alt vs comp_os | 214.40 | 194.90 | 9,10% |
| alt vs comp_sysibm | 223.20 | 233.80 | **-4,75%** |
| alt vs comp_sysmac | 331.10 | 182.20 | 44,97% |
| alt vs comp_windows | 596.30 | 352.60 | 40,87% |
| alt vs misc_forsale | 225.00 | 244.90 | **-8,84%** |
| alt vs rec_autos | 336.40 | 183.10 | 45,57% |
| alt vs rec_motorycycles | 362.10 | 206.90 | 42,86% |
| alt vs rec_baseball | 279.20 | 279.20 | **0%** |

As Tabelas 7.4 a 7.6 mostram resultados dos experimentos repetidos 5 vezes. Em cada treinamento/teste as amostras foram aleatoriamente separadas em conjunto de treino e teste. Percebe-se uma grande redução no tempo de treinamento em muitos resultados, enquanto que para alguns, praticamente não houve ganho. A coluna "RBF" informa a dispersão da Gaussiana. Cada um dos experimentos foi realizado de forma pareada e as amostras foram aleatoriamente geradas em grupos de treinamento e teste. Os resultados mostram evidências do ganho na redução no tempo de treinamento do SVM.

**Tabela 7.4 Classe 14 vs 07. Comparação de tempos, NSMO vs SMO.**

| RBF | Tempo (s) NSMO | Tempo (s) SMO | Ganho |
|---|---|---|---|
| 3 | 1065 | 1065,5 | 0,05% |
| 3 | 1308,7 | 1858,3 | 29,58% |
| 3 | 1561,5 | 1743,8 | 10,45% |
| 3 | 1101 | 1101,9 | 0,08% |
| 3 | 1356 | 1356,3 | 0,02% |
| | | | |
| 5 | 758,4 | 1716,3 | 55,81% |
| 5 | 798 | 798,1 | 0,01% |
| 5 | 898,3 | 2061,8 | 56,43% |
| 5 | 820 | 820,5 | 0,06% |
| 5 | 732,3 | 783 | 6,48% |

**Tabela 7.5 Classe 14 vs 02. Comparação de tempos, NSMO vs SMO.**

| RBF | Tempo (s) NSMO | Tempo (s) SMO | Ganho |
|---|---|---|---|
| 3 | 651,137 | 1542,7 | 57,79% |
| 3 | 2291 | 2291,9 | 0,04% |
| 3 | 880,9 | 1869,2 | 52,87% |
| 3 | 1254 | 1254,5 | 0,04% |
| 3 | 572,9 | 950 | 39,69% |
| | | | |
| 8 | 364 | 364,1 | 0,03% |
| 8 | 879,8 | 1052,1 | 16,38% |
| 8 | 315 | 1326,4 | 76,25% |
| 8 | 470,9 | 983,6 | 52,12% |
| 8 | 789,4 | 834,1 | 5,36% |

**Tabela 7.6 Classe 14 vs 01. Comparação de tempos, NSMO vs SMO.**

| RBF | Tempo (s) NSMO | Tempo (s) SMO | Ganho |
|---|---|---|---|
| 2 | 382,00 | 382,50 | 0,13% |
| 2 | 729 | 729,9 | 0,12% |
| 2 | 813,40 | 884,90 | *8,08%* |
| 2 | 940,9 | 1440,5 | *34,68%* |
| 2 | 819,9 | 1062,3 | *22,82%* |
| | | | |
| 4 | 537,797 | 1553,7 | ***65,39%*** |
| 4 | 1026,7 | 1071,4 | ***4,17%*** |
| 4 | 328,1 | 693,8 | ***52,71%*** |
| 4 | 725 | 725,7 | 0,10% |
| 4 | 694,1 | 1406,2 | ***50,64%*** |

A diminuição no tempo de treinamento não é útil sem que se garanta a capacidade de generalização do SVM; desta forma, uma análise deverá ser conduzida. A verificação do classificador será realizada por meio do teste t de Student; a seguinte hipótese nula (detalhes no Apêndice F) será verificada: O valor F1 do NSMO é igual ao do SMO?

Os resultados do teste t de Student estão abaixo.

**Tabela 7.7 Classe 14 vs 18. Comparação entre os valores F1 do NSMO e SMO.**

| SMO F1 | NSMO F1 |
|---|---|
| 0,9201 | 0,8806 |
| 0,9427 | 0,9501 |
| 0,9332 | 0,9418 |
| 0,9588 | 0,9570 |
| 0,9327 | 0,9364 |
| 0,9138 | 0,9444 |
| 0,9233 | 0,9558 |
| 0,9489 | 0,9510 |
| 0,9567 | 0,9582 |
| 0,9501 | 0,9465 |

**Tabela 7.8 Classe 14 vs 10. Comparação entre os valores F1 do NSMO e SMO.**

| SMO F1 | NSMO F1 |
|---|---|
| 0,8875 | 0,8873 |
| 0,9270 | 0,9357 |
| 0,9059 | 0,9071 |
| 0,9214 | 0,9308 |
| 0,9182 | 0,9657 |
| 0,9264 | 0,9427 |
| 0,9598 | 0,9636 |
| 0,9430 | 0,9429 |
| 0,9570 | 0,9701 |
| 0,9604 | 0,9577 |

**Tabela 7.9 Classe 14 vs 07. Comparação entre os valores F1 do NSMO e SMO.**

| SMO F1 | NSMO F1 |
|---|---|
| 0,9057 | 0,9093 |
| 0,9002 | 0,9005 |
| 0,9112 | 0,9112 |
| 0,9039 | 0,9031 |
| 0,8954 | 0,8946 |
| 0,8969 | 0,8719 |
| 0,9305 | 0,9344 |
| 0,9362 | 0,9243 |
| 0,9261 | 0,9291 |
| 0,9392 | 0,9439 |

**Tabela 7.10 Classe 14 vs 02. Comparação entre os valores F1 do NSMO e SMO.**

| SMO F1 | NSMO F1 |
|---|---|
| 0,9142 | 0,8580 |
| 0,9526 | 0,9514 |
| 0,9436 | 0,9412 |
| 0,9407 | 0,9394 |
| 0,9372 | 0,9443 |
| 0,8933 | 0,9048 |
| 0,9561 | 0,9561 |
| 0,9367 | 0,9296 |
| 0,9468 | 0,9357 |
| 0,9495 | 0,9510 |

**Tabela 7.11 Classe 14 vs 01. Comparação entre os valores F1 do NSMO e SMO.**

| SMO F1 | NSMO F1 |
|---|---|
| 0,8720 | 0,8427 |
| 0,8260 | 0,8278 |
| 0,8533 | 0,8555 |
| 0,7950 | 0,8365 |
| 0,8459 | 0,8449 |
| 0,8774 | 0,8937 |
| 0,9501 | 0,9475 |
| 0,9188 | 0,9129 |
| 0,9259 | 0,9156 |
| 0,9403 | 0,9424 |

Os valores de t calculados estão na tabela abaixo.

**Tabela 7.12 $t_{calc}$ dos valores F1 listados nas tabelas 7.7 a 7.11**

| Classe | $t_{calc}$ |
|---|---|
| **14 vs 18** | -0,4511 |
| **14 vs 10** | -0,8062 |
| **14 vs 07** | 0,2545 |
| **14 vs 02** | 0,5060 |
| **14 vs 01** | -0,0648 |

Os dados da tabela 7.12 mostram que não há variação significativa na capacidade de generalização dos classificadores treinados segundo o algoritmo NSMO. Os valores críticos para um nível de significância de 0,05 e 18 graus de liberdade (conforme mostra a tabela da distribuição t de Student no Apêndice F) é $t_{tab} = \pm 2{,}10$. Ou seja, a hipótese $H_0$ é aceita se $-2{,}10 < t_{calc} < 2{,}10$.

A seguir, no próximo capítulo, a terceira proposta de trabalho é apresentada. Esta proposta tem como finalidade permitir o uso do SVM em base de dados muito grande.

# 8 Maximal trace subset SVM

Este capítulo irá apresentar uma proposta para diminuir o tempo de treinamento do SVM e o número de vetores-suporte, mantendo a performance da classificação tão boa quanto a dos modelos similares baseados na subdivisão do conjunto de treinamento.

A grande diferença em nossa proposta está na preservação do mecanismo de subdivisão das amostras, realizada durante o treinamento, e também na fase de teste e em "produção" (ou seja, o sistema final em uso). Dessa forma, conseguimos diminuir a quantidade de vetores-suporte além de diminuir o tempo de treinamento.

A seguir, na próxima seção, uma breve apresentação dos métodos similares que inspiraram a nossa proposta.

## 8.1 Cascade SVM

O grande problema do SVM, mas também dos outros algoritmos de aprendizado de máquina, está no tratamento de um volume muito grande de dados. Quanto mais amostras, mais memórias serão necessárias. Em geral, a complexidade do SVM é $O(k^3)$ (HANS et al., 2004), onde k é o número de amostras de treinamento. Mesmo no caso do SMO (Platt, 1998), no qual se realiza uma otimização por pares de amostra, é necessário ter uma boa capacidade de armazenamento. Isto se deve ao fato da necessidade de agilizar o treinamento; em qualquer programa computacional, acesso à memória é centenas de vezes mais rápido do que acesso ao disco rígido.

Ainda sobre o SMO, quanto maior a quantidade de amostras, mais produtos *kernel* são necessários. Pode-se então armazenar em memória algumas operações já realizadas, portanto, há um alto consumo de memória a se considerar.

Diversas arquiteturas para máquinas de vetores-suporte (SVM) que podem ser paralelizadas têm sido estudadas para podermos trabalhar com volume de dados extremamente grande. Uma dessas arquiteturas foi desenvolvida por Hans (HANS et al., 2004).

O SVM em cascata de Hans tem a seguinte característica (veja também a Figura 8.1):

- Normalmente, um conjunto de treinamento N é dividido em K subconjuntos de tamanho N/K, sendo que cada subconjunto é formado a partir de uma seleção aleatória das amostras; cada um dos subconjuntos é composto por amostras da classe positiva e da classe negativa selecionadas aleatoriamente para compor os subconjuntos;
- Dados os K subconjuntos, realizam-se os treinamentos com SVM; obtém-se, desta forma, os vetores-suporte de cada um dos subconjuntos treinados;
- Por meio de um processo em cascata, criam-se novos conjuntos de treinamento formados por vetores-suporte do nível acima, por exemplo, na Figura 8.1 temos 8 subconjuntos dos quais sv1 e sv2 representam os vetores-suporte após o treinamento dos dois primeiros grupos. Sv1 e sv2 são então unificados para compor um novo subconjunto, e assim o treinamento continua sucessivamente até que ao final, temos um único SVM.

A vantagem em trabalhar com subconjuntos menores está na possibilidade de realizar um processamento paralelo. A composição em forma de cascata garante a solução do sistema, ou seja, um SVM classificador.

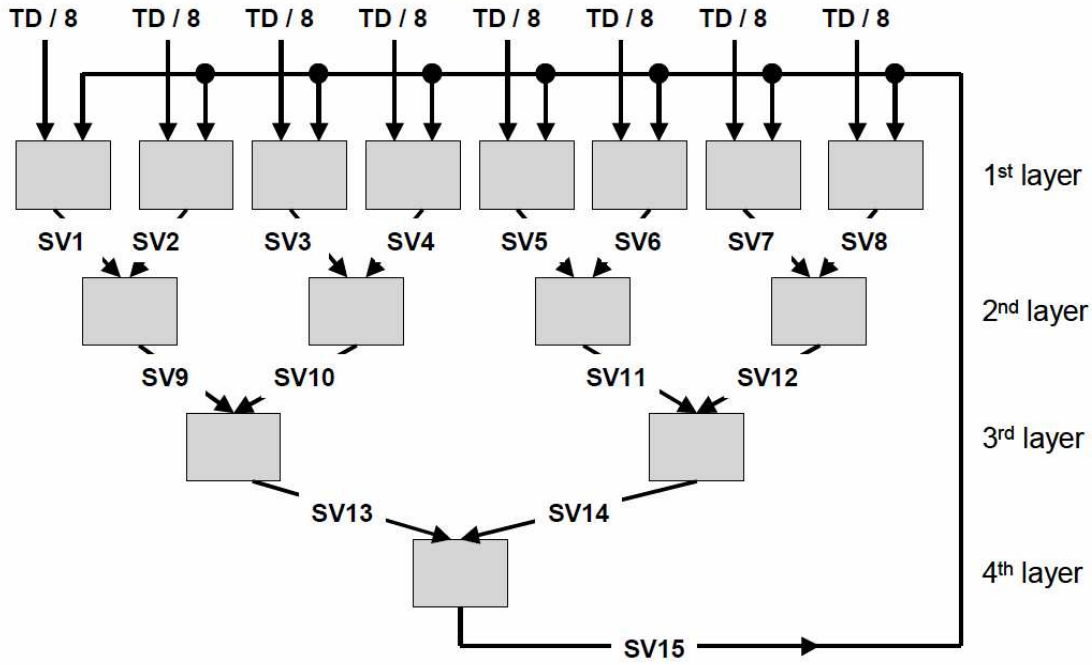



**Figura 8.1 Arquitetura do SVM em cascata binária. Figura extraída de (HANS et al., 2004)**

Experimentos (HANS et al., 2004) mostram que há um ganho de 5x a 10x mais do que treinar um SVM na forma tradicional, em que todas as amostras de N formam um único conjunto.

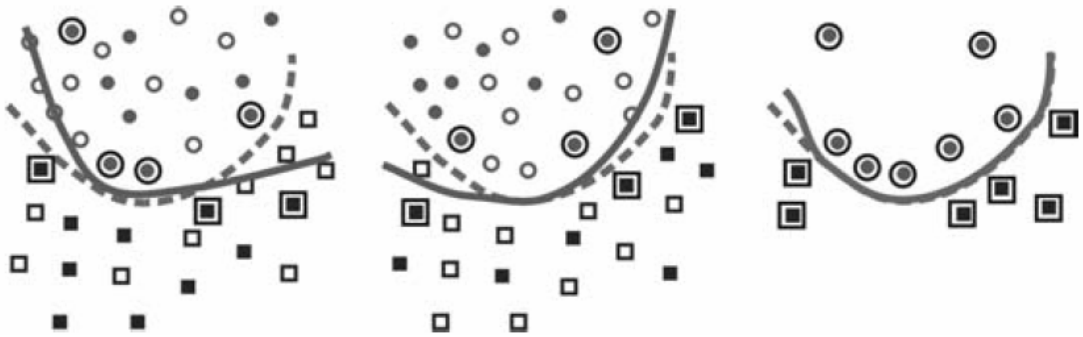


**Figura 8.2 Ilustração do conceito de divisão do conjunto de treino em N subconjuntos. À esquerda e ao centro, dois subconjuntos onde as figuras (círculos e quadrados) sólidas representam as amostras do subconjunto. As figuras com molduras representam os vetores-suporte. Em todos os casos, a linha tracejada representa a superfície separadora ideal, e a linha sólida a superfície separadora obtida. Figura extraída de (Hans et al., 2004).**

A Figura 8.2 ilustra o porquê do funcionamento do SVM em cascata. As amostras nas fronteiras de cada um dos subconjuntos continuam sendo candidatas aos vetores-suporte, e as amostras no "interior" de cada classe continuariam sendo descartadas. Note-se que a grande dificuldade nessa arquitetura e outras similares está na formação do subconjunto ideal. Geralmente, as amostras são escolhidas de forma aleatória. Mas veja na Figura 8.3 que nem sempre será fácil selecionar as amostras.

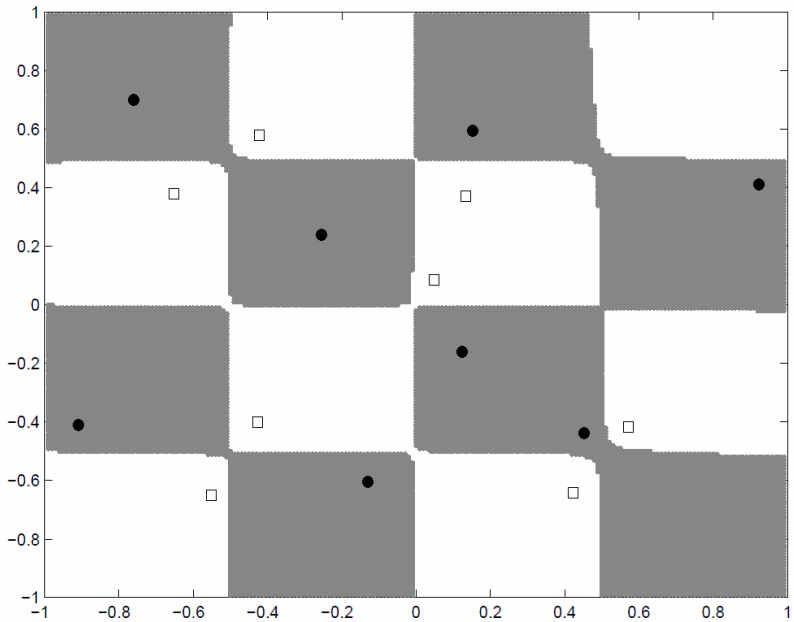



**Figura 8.3 Base de dados no formato de tabuleiro de xadrez. As amostras destacadas em forma de quadrados e círculos representam uma possível formação de um subconjunto. O "tabuleiro" de xadrez representaria o conjunto completo.**

No exemplo apresentado na Figura 8.3, muito utilizado nas provas de conceito de novas arquiteturas SVM, verifica-se que a escolha aleatória resultará em muitos subconjuntos que não poderão convergir para o resultado desejado. Ao contrário do exemplo da Figura 8.3, não há o "interior" das classes positiva e negativa. Temos, na verdade, vários agrupamentos.

Ainda observando a Figura 8.3, as amostras em forma de círculo e quadrado representam um subconjunto formado por elementos selecionados aleatoriamente. Ao treinar um SVM com essas amostras, teremos como resultado vetores-suporte não muito úteis nos processos subsequentes de composição do SVM, tal como mostra a arquitetura ilustrada pela Figura 8.1.

Os autores do SVM em cascata (HANS et al., 2004) garantem a convergência a uma solução global ótima, mas levantam dúvidas quanto à eficiência da arquitetura, ou seja, quão rápido a solução ótima será obtida.

## 8.2 Maximal trace subset SVM – MTS-SVM

A arquitetura proposta neste trabalho quer responder à seguinte pergunta: "todas as amostras são igualmente importantes na classificação?" Esta mesma questão foi levantada por ANTOINE et al. (2005).

Nossa proposta é criar uma nova arquitetura baseada na subdivisão do conjunto de amostras; diferente de outras propostas, a divisão realizada em nossa proposta é permanente, ou seja, não se precisam juntar as subdivisões. Esta característica em nossa arquitetura é viável graças ao uso da medida de entropia como explicado mais adiante. Quando se divide um conjunto de amostras em partes menores, o treinamento torna-se mais simples. Manter essa subdivisão é a nossa proposta, ao contrário dos trabalhos apresentados (veja mais adiante) na literatura. Veja a Figura 8.4.

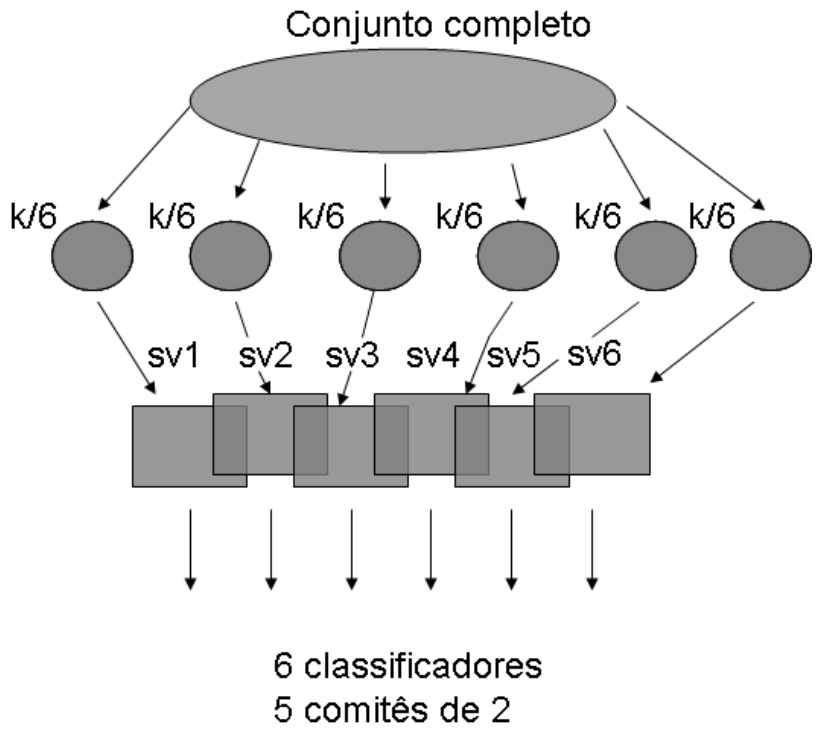



**Figura 8.4 Arquitetura MTS-SVM proposta. K representa a quantidade de amostras do conjunto completo, e os quadrados representam instâncias de SVM.**

A arquitetura proposta neste trabalho está ilustrada na Figura 8.4. Dado um conjunto de treinamento, ele é subdividido em conjuntos distintos menores. Treina-se o SVM para cada um desses subconjuntos. Podemos ter certa sobreposição (cerca de 10%) entre os conjuntos para não limitar a capacidade indutiva da MA nas fronteiras dos conjuntos.

Os subconjuntos são formados ordenando-se todas as amostras do conjunto de treinamento e dividindo-as em conjuntos menores. A ordenação das amostras é realizada escolhendo-se uma das dimensões das amostras.

A melhor dimensão a ser utilizada no processo de criação dos subconjuntos é escolhida por meio do cálculo da entropia de cada uma das dimensões. Buscamos a dimensão com mais alto valor da entropia, pois queremos aquela na qual as amostras estão mais uniformemente distribuídas.

As Figura 8.5 a 9.9 mostram algumas distribuições de amostras e as correspondentes entropias. Na Figura 8.5 há uma distribuição aleatória (entre 0 e 1) de 1000 amostras; o valor absoluto da entropia é 4.5989, bem alta, como esperávamos.

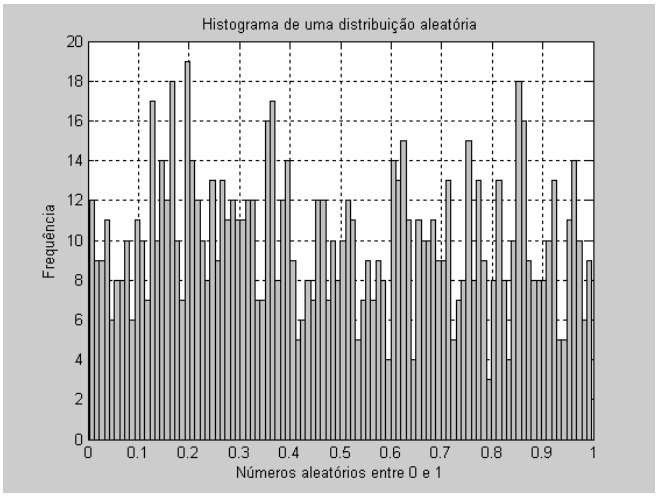



**Figura 8.5 Distribuição aleatória. Entropia de 4.5989**

A Figura 8.6 apresenta uma distribuição aleatória com concentração entre 500 amostras dentre as 1000, ou seja, temos 1500 amostras entre 0 e 1 e 500 amostras entre 0 e 2.5. A entropia absoluta é menor em relação a uma distribuição aleatória.

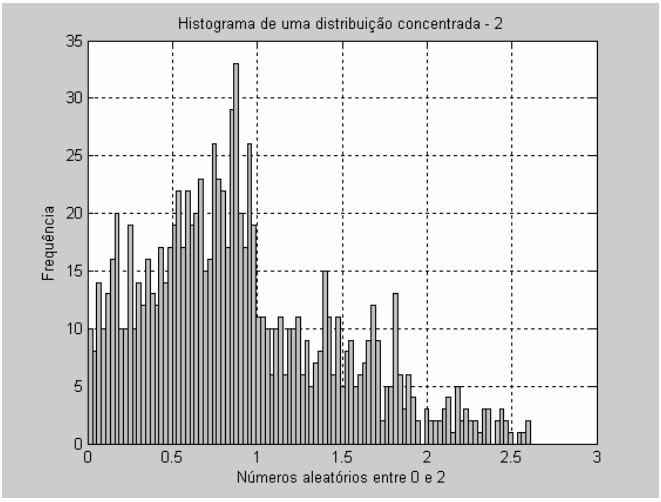



**Figura 8.6 Distribuição concentrada. Entropia de 4.3414**

A Figura 8.7 apresenta outra distribuição concentrada. Sobre a distribuição da Figura 8.6 foram selecionadas mais 250 amostras, cujos valores variam entre 0 e 3.5. A entropia é menor do que a distribuição da Figura 8.7.

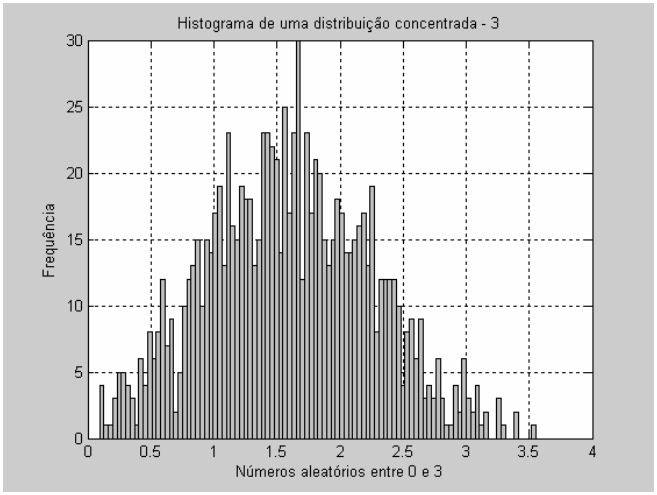



**Figura 8.7 Distribuição concentrada. Entropia de 3.9991**

Na Figura 8.8, há outra distribuição não uniforme. E, como esperado, o valor da entropia é menor, já que existe uma concentração na distribuição das amostras.

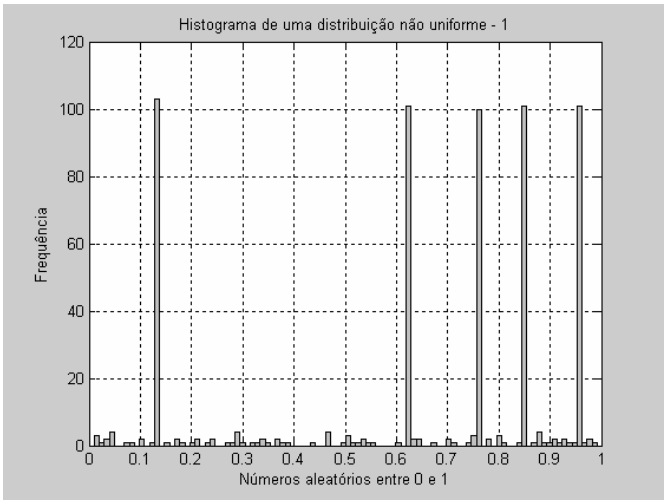



**Figura 8.8 Distribuição não uniforme. Entropia de 4.0351**

Na Figura 8.9 há uma distribuição exponencial. A entropia é bem baixa.

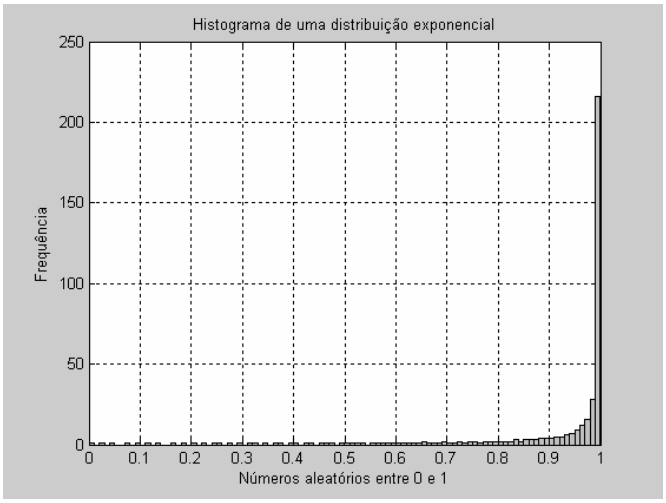



**Figura 8.9 Distribuição exponencial. Entropia de 0.1538**

O tamanho de cada subconjunto é definido segundo a quantidade de amostras que podem ser processadas pelo hardware em uso ou pelos múltiplos de uma unidade, a que damos o nome de traço.

A divisão ordenada proposta neste trabalho permite diminuir em alguns casos a complexidade na separação das classes. O exemplo da Figura 8.10 ilustra este conceito. Temos, nesta figura, três subconjuntos. Enquanto que no problema original (completo) uma função linear não separaria as duas classes representadas pelos símbolos quadrado e círculo, verifica-se que uma função linear separa, sem grandes dificuldades, as duas classes dos subconjuntos 1 e 3.

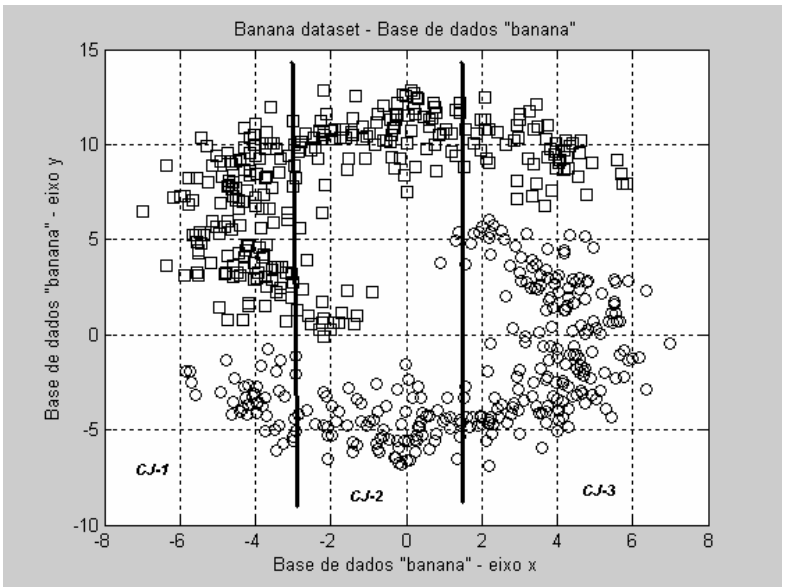



**Figura 8.10 Base de dados cuja distribuição das amostras lembra duas bananas, uma classe representada por quadrados e outra classe por círculos. Subdividindo o conjunto em três partes, podemos usar, neste caso exemplo, um *kernel* linear para a classificação das amostras de cada subconjunto.**

A grande diferença da abordagem tratada neste trabalho com relação a todos os outros similares está no fato da não unificação dos vetores-suporte gerados após o treinamento de cada um dos subconjuntos de treinamento. Se ganha assim tempo, já que não precisamos utilizar as mesmas amostras para um novo processo de treinamento, e também se ganha eficiência na fase de testes e utilização.

Quanto menor for à quantidade dos vetores-suporte menor serão os cálculos necessários (produto *kernel*) na discriminação das classes.

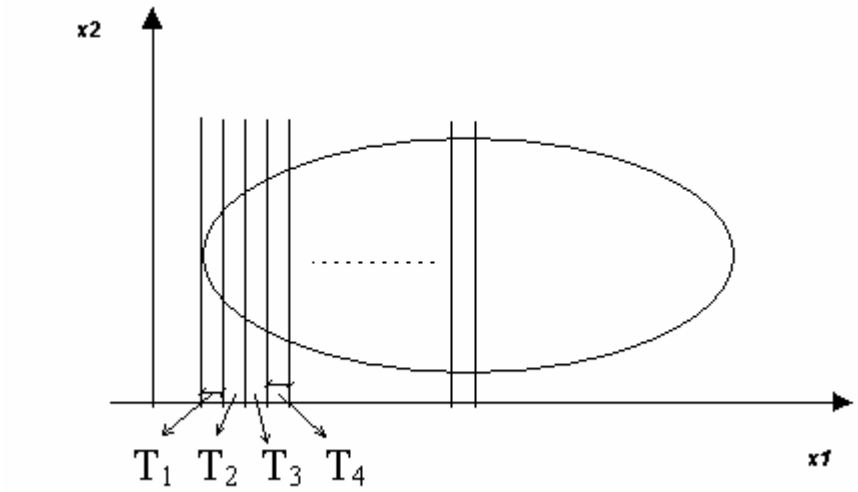



**Figura 8.11 Ilustração do procedimento para obtenção do traço de um segmento de amostras. Divide-se o conjunto completo em pequenas unidades segundo uma dimensão como, por exemplo, a dimensão x1. A subdivisão ideal será um múltiplo da unidade fundamental, o traço.**

Na Figura 8.11 tem uma ilustração do conceito de traço das amostras. T1 corresponde à primeira janela de onde o traço médio será extraído; essa janela nada mais é do que um segmento de amostras numa dimensão selecionada. O tamanho dessa janela foi estabelecido como 5% da variação entre o valor máximo e mínimo nessa mesma dimensão.

**Exemplo**: Na base de dados *UCI-adult dataset*, temos uma dimensão que contém as idades das pessoas que fazem parte da pesquisa. A janela nesse caso será calculada como a maior idade subtraída da menor idade, multiplicado por 5%.

Para cada janela, calculamos o valor do traço médio.

$$\overline{T}_1 = \frac{\sum_{i=1}^{n/2}\left(x_{2i} - x_{2i-1}\right)}{n/2}$$

$$\overline{T}_2 = \frac{\sum_{i=1}^{\frac{n}{2}-1}\left(x_{2i+1} - x_{2i}\right)}{n/2} \qquad \textbf{Eq. 8.1}$$

$$\overline{T} = \frac{\overline{T}_1 + \overline{T}_2}{2}$$

Onde s é a quantidade de amostras de uma janela e $\overline{T}$ o valor do traço médio.

O tamanho de cada subconjunto será dado como múltiplo de $\overline{T}$. O limite inferior será $\overline{T}$ e o limite superior será dado como um *tradeoff* entre a acurácia e o tempo de treinamento. O tempo de treinamento é diretamente proporcional à capacidade computacional do hardware usado.

A Tabela 8.1, apresenta os resultados da simulação realizada da MTS-SVM sobre a base de dados UCI-*Adult dataset*. Podemos verificar que os Grupos 1 e 2 não apresentaram bons índices de classificação. Na verdade, o que ocorreu foi que esses dois grupos somavam apenas 62 amostras positivas de 5005 totais, ou seja, esses dois grupos já estavam desbalanceados, dificultando em muito a classificação.

Em outras palavras, os grupos 1 e 2 da Tabela 8.1 representam os jovens abaixo de 25 anos, cuja renda é inferior a US$ 50.000,00 para 98,76% dos casos. Não há classificação a ser feita nessa faixa etária.

Cada grupo está identificado pela faixa etária - foi a dimensão escolhida. Percebe-se um bom desempenho para todos os outros grupos.

**Tabela 8.1 Base de dados UCI-*adult dataset* dividindo a base em 13 subconjuntos, usando a idade como referência na divisão. Existe uma sobreposição entre os grupos para não limitar a capacidade de generalização do SVM.**

| | Idade | *Overlap* | *Precision* | *Recall* | F1 |
|---|---|---|---|---|---|
| **Grupo 1** | 17 a 22 | 0% | 50,00% | 50,00% | 50,00% |
| **Grupo 2** | 21 a 25 | 40% | 50,00% | 50,00% | 50,00% |
| **Grupo 3** | 24 a 28 | 40% | 63,25% | 61,14% | 62,07% |
| **Grupo 4** | 27 a 31 | 40% | 79,46% | 68,50% | 72,07% |
| **Grupo 5** | 30 a 34 | 40% | 75,77% | 74,48% | 75,08% |
| **Grupo 6** | 33 a 37 | 40% | 78,22% | 75,07% | 76,37% |
| **Grupo 7** | 36 a 40 | 40% | 77,08% | 73,09% | 74,51% |
| **Grupo 8** | 39 a 43 | 40% | 78,84% | 74,99% | 76,26% |
| **Grupo 9** | 43 a 47 | 20% | 77,48% | 76,17% | 76,69% |
| **Grupo 10** | 46 a 50 | 40% | 77,72% | 76,93% | 77,26% |
| **Grupo 11** | 49 a 53 | 40% | 75,07% | 75,37% | 75,21% |
| **Grupo 12** | 53 a 60 | 13% | 73,16% | 74,54% | 73,29% |
| **Grupo 13** | 59 a 100 | 9% | 79,87% | 68,43% | 71,49% |

Na Tabela 8.2 estão apresentados os tempos de treinamento para cada grupo.

**Tabela 8.2 Base de dados UCI-*adult dataset*. Tempo de treinamento de cada um dos 13 subconjuntos. Foi realizado um treinamento 2-fold com 5 repetições.**

| | Tamanho (amostras) | Tempo (s) | Desvio (s) |
|---|---|---|---|
| **Grupo 1** | 3895 | 1099,50 | 12 |
| **Grupo 2** | 4001 | 1545,90 | 11 |
| **Grupo 3** | 4126 | 1152,70 | 12 |
| **Grupo 4** | 4264 | 2363,60 | 10 |
| **Grupo 5** | 4338 | 2397,40 | 10 |
| **Grupo 6** | 4393 | 2769,30 | 13 |
| **Grupo 7** | 4193 | 2954,90 | 11 |
| **Grupo 8** | 3968 | 2992,30 | 12 |
| **Grupo 9** | 3673 | 1728,20 | 11 |
| **Grupo 10** | 3167 | 952,20 | 10 |
| **Grupo 11** | 2716 | 938,20 | 14 |
| **Grupo 12** | 3055 | 1505,10 | 12 |
| **Grupo 13** | 2999 | 1039,30 | 15 |

Para ilustrar a vantagem de dividir um conjunto grande em pequenos subconjuntos, peguemos, por exemplo, o grupo 5 da Tabela 8.2; dividindo-o em outros 5 subgrupos, respectivamente idade=30,31,32,33,34, e treinando o SVM nesses novos grupos, temos um ganho em tempo de treinamento muito grande - veja a Tabela 8.3. Somando-se o tempo de treinamento de cada SVM, temos 149,8 segundos, muito menor do que 2397,40 segundos do Grupo 5. A divisão de um grupo grande em pequenos grupos traz resultados significativamente melhores.

**Tabela 8.3 Treinamento dos 5 subconjuntos do grupo 5 da Tabela 8.2. Ilustração do conceito de divisão de conjuntos. A soma dos tempos de treinamento dos cinco subconjuntos é muito menor do que o tempo do grupo 5 na Tabela 8.2.**

| Grupo | **Tamanho** | **Tempo (s)** | **Vetores-suporte** |
|---|---|---|---|
| **1** | 861 | 39,3 | 860 |
| **2** | 888 | 29 | 888 |
| **3** | 828 | 27,1 | 828 |
| **4** | 875 | 26,5 | 875 |
| **5** | 886 | 27,9 | 886 |
| **Soma** | 4338 | 149,8 | 4337 |

## 8.3 Experimentos - Resultados do MTS-SVM com dados da 20Newsgroup

Nas Tabelas 8.4, 8.5, 8.6, 8.7 e 8.8 SMO refere-se à implementação de Keerthi (KEERTHI et al., 2001), enquanto que o MTS-SVM se refere ao algoritmo proposto neste trabalho. A coluna **RBF** diz respeito ao parâmetro de ajuste utilizado na função Kernel com o mesmo nome. A coluna **tempo** informa quantos segundos levou o processo de treinamento. Já as colunas **treino** e **teste** informam os resultados de acurácia do SVM treinado, os dados da classe positiva estão representados pelo sinal "**+**"e os dados da classe negativa pelo sinal "**-**".

A classe ou categoria utilizada nesta seção foi a Classe 1 (veja a Seção 6.4). O SVM foi treinado para cada uma das combinações da técnica um-contra-um no treinamento multi-classe. Ao final, a medida F1 (que vão de 0 a 1, onde 1 representa uma acurácia de 100%) foi obtida para verificar a capacidade de acurácia dos SVMs.

Dados da Tabela 8.4 mostram que um dos subconjuntos demorou muito (210,7 segundos), um valor próximo ao tempo de treinamento do conjunto todo. Em alguns casos, a complexidade existente na separabilidade das amostras faz com que quase não haja ganho na diminuição do tempo de treinamento. No entanto, é importante observar que ao contrário das técnicas existentes (SYED, et al., 1999; HANS et al., 2004), esta abordagem mantém os vetores-suporte disjuntos, sem uni-los no final do treinamento. Isso traz uma vantagem na redução do tempo de teste ao usar o SVM treinado quando será necessário realizar produtos escalares entre as amostras de teste e cada vetor-suporte.

**Tabela 8.4 Classe 1 contra a classe 2**

| RBF | Tipo | Tempo (s) | treino (+) | teste(+) | treino (-) | teste (-) | F1 |
|---|---|---|---|---|---|---|---|
| 10 | MTSSVM | 210,7 | 97,3214 | 85,6322 | 100 | 97,2826 | 0,9157 |
| 10 | MTSSVM | 159,9 | 98,8235 | 84,2857 | 90,6615 | 98,5714 | 0,9200 |
| 10 | SMO | 259,6 | 97,9123 | 86,9427 | 100 | 97,8395 | 0,9260 |

Cada um dos SVMs em MTS-SVM podem ser treinados e utilizados separadamente. Na Tabela 8.5 pode se ver que o SVM normal demora mais de 1500 segundos para treinar, enquanto que os MTS-SVM demoram bem menos.

**Tabela 8.5 Classe 1 contra a classe 7**

| RBF | Tipo | Tempo (s) | treino (+) | teste(+) | treino (-) | teste (-) | F1 |
|---|---|---|---|---|---|---|---|
| 10 | MTSSVM | 99 | 97,3214 | 85,0575 | 100 | 98,4848 | 0,9077 |
| 10 | MTSSVM | 206,4 | 98,8235 | 81,4286 | 100 | 96,2963 | 0,9377 |
| 2 | SMO | 1502,1 | 98,5386 | 87,2611 | 100 | 96,0573 | 0,9264 |

Dados da Tabela 8.6 mostram que o MTS-SVM em um dos subconjuntos leva pouco mais do que a metade do tempo de treinamento do SVM normal. Já no primeiro subconjunto o tempo e significativamente menor.

**Tabela 8.6 Classe 1 contra a classe 10. Comparação entre MTS-SVM e SVM.**

| RBF | Tipo | Tempo (s) | treino (+) | teste(+) | treino (-) | teste (-) | F1 |
|---|---|---|---|---|---|---|---|
| 7 | MTSSVM | 174,6 | 97,7679 | 85,6322 | 100 | 92,1875 | 0,9337 |
| 7 | MTSSVM | 389,8 | 98,8235 | 82,1429 | 100 | 99,0566 | 0,8268 |
| 7 | SMO | 734,8 | 97,4948 | 84,0764 | 100 | 96,0733 | 0,8919 |

Os resultados apresentados na Tabela 8.7 e 8.8 são semelhantes aos resultados da Tabela 8.6. Temos um subconjunto com tempo muito baixo enquanto que o outro demora metade (ou, um pouco mais) do tempo de treinamento do SVM normal.

**Tabela 8.7 Classe 1 contra a classe 14. Comparação entre MTS-SVM e SVM.**

| RBF | Tipo | Tempo (s) | treino (+) | teste(+) | treino (-) | teste (-) | F1 |
|---|---|---|---|---|---|---|---|
| 7 | MTSSVM | 94,5 | 98,2143 | 84,4828 | 100 | 96,5517 | 0,9218 |
| 7 | MTSSVM | 262,9 | 98,8235 | 80,0000 | 100 | 95,8549 | 0,8580 |
| 7 | SMO | 543,6 | 97,9123 | 84,3949 | 100 | 95,8580 | 0,9015 |

**Tabela 8.8 Classe 1 contra a classe 18. Comparação entre MTS-SVM e SVM.**

| RBF | Tipo | Tempo (s) | treino (+) | teste(+) | treino (-) | teste (-) | F1 |
|---|---|---|---|---|---|---|---|
| 10 | MTSSVM | 134 | 97,3214 | 84,4828 | 99,4186 | 90,5263 | 0,9098 |
| 10 | MTSSVM | 401,5 | 98,8235 | 80,7143 | 99,1346 | 98,8971 | 0,8368 |
| 8 | SMO | 659,5 | 97,4948 | 80,2548 | 100 | 97,2752 | 0,8829 |

As Tabelas 8.9 a 8.13 mostram uma comparação de resultados entre os dois SVM. A coluna **TN** significa "*True-Negative*" e contém a quantidade de amostras da classe negativa que foram classificadas como negativas durante a fase de testes. Já a coluna **TP** significa "*True-Positive*" e contém a quantidade de amostras da classe positiva que foram classificadas como sendo positivas durante a fase de testes.

As amostras em MTS-SVM são pré-agrupadas segundo o classificador em que elas serão submetidas. Essas mesmas amostras foram submetidas para teste no SMO tradicional. O resultado mostra que o MTS-SVM classifica corretamente as amostras. Veja por exemplo os dados da tabela 8.9. Temos TN do subconjunto 1 igual a 158, o TN do subconjunto 2 igual a 118. A soma desses dois valores supera o resultado do SMO tradicional (cujo valor é 282), ou seja, mais amostras foram classificadas corretamente usando-se o MTS-SVM.

**Tabela 8.9 Classe 1 contra a classe 2. Capacidade de generalização.**

| RBF | Tipo | TN | TP |
|---|---|---|---|
| 10 | MTSSVM | 158 | 169 |
| 10 | MTSSVM | 118 | 138 |
| 10 | SMO | 282 | 307 |

**Tabela 8.10 Classe 1 contra a classe 7 Capacidade de generalização.**

| RBF | Tipo | TN | TP |
|---|---|---|---|
| 10 | MTSSVM | 163 | 173 |
| 10 | MTSSVM | 68 | 135 |
| 2 | SMO | 243 | 302 |

**Tabela 8.11 Classe 1 contra a classe 10 Capacidade de generalização.**

| RBF | Tipo | TN | TP |
|---|---|---|---|
| 7 | MTSSVM | 55 | 160 |
| 7 | MTSSVM | 261 | 139 |
| 7 | SMO | 321 | 302 |

**Tabela 8.12 Classe 1 contra a classe 14 Capacidade de generalização.**

| RBF | Tipo | TN | TP |
|---|---|---|---|
| 7 | MTSSVM | 123 | 168 |
| 7 | MTSSVM | 154 | 134 |
| 7 | SMO | 285 | 301 |

**Tabela 8.13 Classe 1 contra a classe 18 Capacidade de generalização.**

| RBF | Tipo | TN | TP |
|---|---|---|---|
| 10 | MTSSVM | 80 | 158 |
| 10 | MTSSVM | 220 | 138 |
| 8 | SMO | 295 | 305 |

Esta seção apresentou resultados dos testes realizados que compravam a eficiência do MTS-SVM diminuindo o tempo de treinamento do SVM. Além da diminuição no tempo de treinamento, o fato de ter conjuntos independentes menores permite fornecer resultados em menos tempo quando o classificador estiver sendo usado na prática.

## 8.4 Experimentos - Resultados do MTS-SVM – Mushroom dataset

Na Tabela 8.14 estão os resultados após o treinamento dos dados de cogumelos (veja a Seção 5.1.4) disponíveis na Universidade de Califórnia – Irvine (UCI – dataset). Essa base de cogumelos é uma base pública.

Utilizando-se da técnica descrita na Seção 8.2, escolheu-se a primeira dimensão como sendo a melhor. Dividiram-se então as amostras em 2 grupos, cada uma complementar a outra. Na tabela 8.14 esses grupos estão indicados como 'Tipo=SSMO'. Já o 'Tipo=SMO' refere-se à base completa. Para chegarmos à conclusão de que a divisão do conjunto de amostras não interfere no resultado final, a soma dos valores de TN e TP para os 'Tipo=SSMO' devem estar bem próxima do TN total e TP total representado pelo 'Tipo=SMO'.

**Tabela 8.14 Resultados experimentais do MTS-SVM para dados de cogumelos.**

| Repetição | Tipo | Tempo (s) | TN | TP |
|---|---|---|---|---|
| 1 | SSMO | 3420,3 | 762 | 619 |
| 1 | SSMO | 209,6 | 507 | 626 |
| 1 | SMO | 6534,9 | 1340 | 1228 |
| | | | | |
| 2 | SSMO | 4669,6 | 710 | 693 |
| 2 | SSMO | 2283,6 | 626 | 578 |
| 2 | SMO | 17275,1 | 1393 | 1227 |
| | | | | |
| 3 | SSMO | 712 | 682 | 646 |
| 3 | SSMO | 2985,9 | 602 | 642 |
| 3 | SMO | 6023,6 | 1347 | 1147 |

O experimento foi repetido três vezes, em cada repetição as amostras foram separadas em conjunto de treino e teste aleatoriamente. É possível observar uma enorme vantagem na redução do tempo de treinamento, mas ao mesmo tempo, muito pouca interferência na capacidade de generalização do classificador.

# 9 Conclusão e comentários finais

A classificação de texto ou documento apresentada neste trabalho foi conduzida utilizando-se o SVM como a MA. Existem várias codificações do SVM, algumas delas utilizam intensamente algoritmos de otimização matemática; outros algoritmos como SMO, que foi usado nesta tese, realizam vários processamentos intensos, simples e rápidos para contornar o problema de trabalhar com matrizes grandes, tão grandes quanto à quantidade de amostras que são usadas no treinamento.

Apresentamos nesta tese três mecanismos para melhorar o treinamento do SVM, em especial aplicado à classificação de documentos. Descrevemos uma nova técnica para indexação das palavras que formam o modelo vetorial local e aperfeiçoamos o SMO de Keerthi. Foi também proposta uma nova arquitetura SVM que permite uma computação paralela para obtenção dos vetores-suporte; com isso, consegue-se uma redução muito grande no tempo de treinamento, teste e no uso do SVM.

SVM é um classificador binário. Desta forma, é comum mencionar amostras da classe positiva e amostras da classe negativa. Em problemas de multiclassificação esses classificadores binários são compostos segundo duas formas básicas:

- Um-contra-um: diversos classificadores binários são treinados separadamente. Ao final, na fase de testes da multiclassificação, um esquema de votação diz a classe à qual uma amostra teste pertence. Esta foi a abordagem usada nesta tese.
- Um-contra-todos ou um-contra-resto: uma das classes é escolhida como sendo a classe positiva e todas as outras são consideradas como sendo a negativa. Aqui temos um problema, a quantidade de amostras das duas classes pode ser bem diferente.

Uma linha de pesquisa crescente é a classificação de uma-classe. Um dos principais aspectos desse classificador é contornar o problema de base dados não balanceados, ou seja, a ausência da mesma quantidade de amostras positivas e negativas. Existe, no entanto, um problema nesta abordagem. Trabalhando-se apenas com as amostras da classe positiva, e desconhecendo o restante do 'universo', corre-se o risco de classificar amostras falsas positivas. Para a classificação de uma-classe funcionar bem, é necessário caracterizar bem o problema extraindo atributos relevantes e usando técnicas de pré-processamento adequadas para essa extração.

A Tabela 5.4 apresenta os resultados dos testes de um classificador de uma-classe “real”, ou seja, aquele que trata apenas amostras da classe positiva. Embora ela tenha sido utilizada com sucesso em várias aplicações, as preocupações levantadas neste trabalho ainda continuam válidas.

A abordagem adotada nesta pesquisa foi treinar um SVM de duas classes, mas visando à classificação bem sucedida de apenas uma das classes. Tendo quatro classificadores um-contra-um, o objetivo é conseguir classificar bem apenas a amostra positiva. Os resultados estão apresentados na Tabela 7.2, e eles se comparam aos números obtidos em outros trabalhos (YANG, 1999), (JOACHIMS, 1998).

Neste trabalho usamos um vetor de espaço local, com técnica de extração de atributos discriminativos para representar as amostras da classe positiva. Usando-se um vetor localizado reduzimos muito a quantidade de elementos esparsos na matriz de atributos final. Soma-se a essa representação a introdução da correlação total ponto-a-ponto para ir além do tradicional *bag-of-words*. As composições de palavras correlatas dão ao sistema um “nível semântico” na caracterização dos documentos de uma classe.

Foi utilizada a base de dados pública da Reuters, a Reuters-21578. As cinco categorias com maior volume de documentos foram usadas; então, a correlação total ponto-a-ponto foi utilizada para a indexação das palavras. Foi gerado um modelo vetorial local por meio do uso da Informação Mútua das palavras obtidas pelo PTC (Seção 6.1); usando o MI, conseguimos verificar a relevância das palavras num contexto global, ou seja, considerando todos os documentos das cinco categorias selecionadas.

Outra base de dados pública utilizada foi a 20Newsgroup (20NG). Os mesmos passos usados na base da Reuters-21578 foram seguidos. Comparando os resultados da base de dados da Reuters com a literatura (Tabela 7.2), percebe-se que os resultados estão próximos, mas não superiores. Os resultados da base de dados 20NG com PTC e sem PTC foram comparados entre si, veja nas Tabelas 6.10 a 6.13 que os resultados também estão próximos. Desta forma, apesar dos resultados não mostrarem uma vantagem da nova representação dos documentos, conseguimos alternativas para representação dos documentos além do tradicional *bag-of-words*.

SMO de Platt (PLATT, 1998) foi alterado e melhorado por diversos pesquisadores (KEERTHI et al., 2001; CHEN; FAN; LIN, 2006). Neste trabalho contribuímos com mais duas propostas para diminuição do custo computacional que existe nos treinamentos dos SVMs. Os algoritmos que codificam o SVM podem consumir muita memória, muito CPU ou ambos. Consequentemente, o tempo final de treinamento pode aumentar, mantendo-se fixo os recursos computacionais, e aumentando o volume de dados.

A primeira contribuição deste trabalho usando o SVM foi na otimização do processo de treinamento do SMO por meio da atualização da vizinhança dos pares que mais violam as condições KKT (Capítulo 7). Podem ser feitas extensões deste conceito de atualização da vizinhança de um vetor-suporte para outros algoritmos SVM como, por exemplo, o modelo de (FINE; SCHEINBERG, 2002; SCHEINBERG, 2006).

A Tabela 7.1 apresenta os resultados em termos do tempo de treinamento da proposta deste trabalho, o NSMO. Com a introdução do conceito de atualização da vizinhança da amostras candidatas, conseguimos eliminar algumas iterações no programa. O ganho aproximado obtido foi de 16% para uma base artificial (veja a Figura 8.3), a *chessboard*, e para os dados da Reuters-21578, foi de 6% (veja as Tabelas 7.1 e 7.2).

Usando uma base de dados mais complexa, a 20NG, pode-se ver nos resultados das Tabelas 7.3 a 7.6 que também houve ganhos significativos na redução do tempo de treinamento. Um fator importante a destacar é a capacidade de generalização do SVM treinado. A redução no tempo de treinamento com uma piora na classificação não seria nada interessante. Assim, para verificação da manutenção da acurácia na classificação foi realizado um teste de hipóteses, o teste t de Student. A Tabela 7.12 mostra que não houve mudança nos valores médios da métrica F1 dos classificadores que usaram o NSMO e dos classificadores que não usaram o NSMO.

No NSMO, com a diminuição da quantidade de iterações “extra”, conseguiu-se atingir bons resultados, ou seja, diminuição do tempo de treinamento. Já o *Subset* SVM (é a segunda contribuição usando SVM apresentada no Capítulo 8) permite 'dividir' um conjunto de amostras para trabalhar com unidades menores de treinamento. A grande diferença da abordagem apresentada nesta tese com as demais (SYED et al., 1999; HANS et al., 2004) está em manter separados os vetores-suporte após o final do treinamento. Trabalhando com

unidades menores, fica evidente a redução no tempo de treinamento (Tabelas 8.4 a 8.9); da mesma forma como no caso do NSMO, é necessário avaliar se a eficiência do classificador continua sendo a mesma. As Tabelas 8.9 a 8.13 mostram os resultados da classificação de cada uma dos subconjuntos e do conjunto completo. Se a soma das classificações corretas (seja, Verdadeiro Positivo, seja Verdadeiro Negativo) estiver próximo ao valor correspondente do classificador treinado com todas as amostras, então foi possível manter a capacidade de generalização do classificador, tal como ocorreu.

O *Subset* SVM pode ser aplicado para outras bases de dados. Para mostrar resultados com bases de dados que não sejam textuais, foram realizados experimentos com uma base de cogumelos (da UCI) (veja a Tabela 8.14) e outra com uma base de dados de renda, a UCI-*adult dataset* (Tabelas 8.1 a 8.3). Nos dois casos, os resultados também foram boas como no caso da base 20NG.

## 9.1 Comentários e Limitações

Algumas limitações devem ser comentadas nos algoritmos propostos. O algoritmo NSMO que visa atualizar, no mesmo ciclo de alteração dos coeficientes de Lagrange, aqueles que mais violam as condições KKT, necessita de ajustes dos parâmetros de controle; a distância máxima (raio) dentro da qual as amostras vizinhas estão contidas, bem como o nível de tolerância para verificação das modificações nos valores dos coeficientes de Lagrange devem ser ajustados empiricamente conforme as características de cada base de dados.

É comum verificar em várias técnicas de aprendizado de máquina a necessidade de ajustes de parâmetros, ou seja, o treinamento é paramétrico. O algoritmo proposto nesta tese também se enquadra nesta categoria.

Já o *Subset* SVM necessita de uma dimensão em que se consegue uma distribuição uniforme de amostras entre os conjuntos de treino positivo e negativo. Se tal dimensão não existir, caímos no problema do treinamento de SVM de classes não balanceadas, o que pode prejudicar muito a capacidade de generalização do classificador.

Quanto ao uso da representação ou vetor local (*local vector-space model*), seu uso é recomendado para diversas outras aplicações pelo simples fato de reduzir o tamanho do vetor de atributos.

## 9.2 Trabalhos Futuros

A correlação total ponto-a-ponto ainda pode trazer boas contribuições na análise semântica dos documentos. A sua abordagem em que combinações correlatas e significativas das palavras são utilizadas para formar o atributo permite uma análise como realizada usando a rede SOM (LIN; SOERGEL; MARCHIONINI, 1991).

As técnicas apresentadas nesta Tese podem ter uma etapa adicional introduzida no início do processo de aprendizado. Pode-se usar a Análise Linear Discriminante (Linear Discriminant Analysis - LDA) para redução da dimensão das amostras. LDA (FISHER, 1936; GUYON, ELISSEEFF, 2003) transforma amostras multivariada em univariada, maximizando a separação entre as amostras de classes diferente, e agrupando as amostras da mesma classe.

Formalmente, LDA maximiza a relação das variâncias entre-classe e intra-classe. A dimensão final é sempre K-1, onde K é o número de classes. Em problemas de classificação binária teremos as amostras projetadas no espaço unidimensional. O problema da não singularidade da matriz das covariâncias (intra-classe) pode ser solucionado utilizando-se diversas técnicas de redução de dimensionalidade como a RFE, *Information Gain*, *Mutal Information* entre outras apresentadas nesta Tese.

Já no caso do *Subset* SVM pode-se utilizar mais uma análise, a Análise das Componentes Principais (Principal Component Analysis - PCA) (JOLLIFE, 1986). Esta análise é uma técnica que busca pela direção com maior variância das amostras, uma série de novas projeções ortogonais são obtidas. A grande diferença do PCA com a LDA é que a primeira busca a projeção com maior variância sem se preocupar com a discriminação das amostras entre as classes às quais elas pertencem. LDA é uma técnica discriminativa.

*Subset* SVM busca pela dimensão de maior variância. PCA preserva a distância relativa entre as amostras, algo fundamental para que os subconjuntos sejam criados em tamanhos homogêneos.

Outra possibilidade a explorar é o uso da rede neural SOM (KOHONEN et al, 2000; KOHONEN, 2001) de Kohonen como por exemplo a sua versão aplicada para agrupamento de notícias das listas de discussão na 'internet', a Websom (LAGUS; KASKI; KOHONEN, 2004). Com ela, as amostras de alta dimensão do espaço das entradas seriam mapeadas no espaço dos mapas. O espaço dos mapas em geral tem o formato de grade bidimensional. Desta forma, amostras correlatas de cada grupo formariam agrupamentos; esses conjuntos de agrupamentos, dois a dois (um conjunto representando amostras da classe positiva e outro conjunto representando amostras da classe negativa), seriam treinados usando o SVM.

## 9.3 Considerações Finais

Nesta tese de doutorado foram apresentados algoritmos (NSMO) e técnicas (*Subset* SVM) para redução do tempo de treinamento de um SVM. Foi apresentada também uma nova forma de representar os documentos (PTC).

O trabalho teve como foco a classificação de documentos, no entanto, algumas bases não 'textuais' foram utilizadas para mostrar que as idéias aqui defendidas podem também ser aplicadas para outros problemas de classificação.

Os resultados experimentais apresentados evidenciam redução no tempo de treinamento do SVM sem perda da capacidade de generalização (NSMO e *Subset* SVM). Já a nova representação dos documentos utilizando-se do PTC contribui para enriquecer a representação dos documentos, além do tradicionamente usado 'bag-of-single-words'.

O trabalho realizado possibilita diversas pesquisas futuras explorando os pontos destacados na seção anterior.

## Bibliografia

[ANTOINE, et al., 2005] BORDES, A.; ERTEKIN, S.; WESTON, J.; BOTTOU, L. "**Fast Kernel Classifiers with Online and Active Learning**", Journal of Machine Learning Research, v. 6, p. 1579–1619, 2005

[ARGONNE, 2005] Argonne National Laboratory é um laboratório especializado em técnicas de otimização matemática para diversas finalidades. Disponível em: <http://www-fp.mcs.anl.gov/otc/Guide/index.html>. Acesso em: 10 de Outubro de 2009

[BEDO; SANDERSON; KOWALCZYK, 2006] BEDO, J.; SANDERSON, C.; KOWALCZYK, A. "**An efficient alternative to SVM based recursive feature elimination with applications in natural language processing and bioinformatics**", In: AI 2006: Advances in Artificial Intelligence Lecture Notes in Computer Science, Springer Berlin / Heidelberg, ISBN 978-3-540-49787-5, , v. 4304, p. 170-180, 2006

[BEKKERMAN et. al, 2003] BEKKERMAN, R.; EL-YANIV, R.; TISHBY, N.; WINTER, Y. "**Distributional Word Clusters vs. Words for Text Categorization**", Journal of Machine Learning Research, v. 3, p. 1183-1208, 2003

[BISHOP, 1995] BISHOP, C. M. "**Neural Networks for Pattern Recognition**", Clarendon Press, Oxford, 1995

[BOTTOU, 2007] BOTTOU, L.; LIN, C. J. "**Support vector machine solvers**", In: BOTTOU, L; CHAPELLE, O.; DECOSTE, D.; WESTON, J. **Large-Scale Kernel Machines** 1st. ed. Cambridge, MA, USA; London, England: The MIT Press, chap. 1, p. 416, 2007

[BUCKLEY, 1993] BUCKLEY, C., "**The Importance of Proper Weighting Methods**", Proceedings of the workshop on Human Language Technology, 1993

[BUSSAB; MORETTIN, 1987] BUSSAB, W. O.; MORETTIN, P. A., "**Estatística Básica**", Atual Editora, 4ed. 1987

[CAROPRESO; MATWIN; SEBASTIANI, 2001] CAROPRESO, M. F.; MATWIN, S.; SEBASTIANI, F., "**A learner-independent evaluation of the usefulness of statistical phrases for automated text categorization**", In Text Databases and Document Management: Theory and Practice, A. G. Chin, ed. Idea Group Publishing, Hershey, PA, 78–102, 2001

[CAUWENBERGHS; POGGIO, 2001] CAUWENBERGHS, G.; POGGIO, T. "**Incremental and Decremental Support Vector Machine Learning**", in Adv. Neural Information Processing Systems (NIPS*2000), Cambridge MA: MIT Press, v. 13, 2001

[CHEN; FAN; LIN, 2006] CHEN, P.-H.; FAN, R.-E.; LIN, C.-J. "**A study on SMO-type decomposition methods for support vector machines**", IEEE Transactions on Neural Networks, 17, 893–908, 2006

[CHEN; ZHOU; HUANG, 2001] CHEN, Y.; ZHOU, X.; HUANG, T. S. "**One-Class SVM for learning in image retrieval**", IEEE Conference on Image Processing, Greece, 2001

[COOPER, 1997] Cooper, G., et al., "**An evaluation of machine-learning methods for predicting pneumonia mortality**", Artificial Intelligence in Medicine, vol.9, pp.107-138, 1997

[CORTES; VAPNIK, 1995] Cortes, C.; Vapnik, V., "**Support vector networks**", Machine Learning, 20.1-25, 1995

[CRISTIANINI; TAYLOR, 2000] CRISTIANINI, N. E; SHAWE-TAYLOR, J., "**An Introduction to Support Vector Machines**", Cambridge University Press, 2000

[DEBOLE; SEBATIANI, 2003] DEBOLE, F.; SEBASTIANI, F. "**Supervised term weighting for automated text categorization**", Proceedings of the 2003 ACM symposium on Applied computing, p. 784-788, 2003

[DEBOLE; SEBATIANI, 2004] DEBOLE, F.; SEBASTIANI, F. "**An analysis of the relative difficulty of Reuters-21578 subsets**", In Proceedings of LREC-04, 4th International Conference on Language Resources and Evaluation, Lisbon, PT, p. 971-974, 2004

[DOMENICONI; GUNOPULOS, 2001] DOMENICONI, C.; GUNOPULOS, D. "**Incremental Support Vector Machine Construction**", IEEE International Conference on Data Mining, San Jose, CA, USA, p. 589-592, 2001

[DOYLE, 1961] DOYLE, L., "**Semantic road maps for literature searchers**", Journal of the ACM, 1961

[DUMAIS et. al, 1998] DUMAIS, S. T.; PLATT, J.; HECKERMAN, D.; SAHAMI, M. "**Inductive learning algorithms and representations for text categorization**", In Proceedings of CIKM'98, 7th ACM International Conference on Information and Knowledge Management, Bethesda, ACM Press, New York, US, p. 148–155, 1998

[ERIKSSON et al., 2005] ERIKSSON, T.; KIM, S.; KANG, HONG-GOO; LEE, C., "**An information-theoretic perspective on feature selection in speaker recognition**", Signal Processing Letters, IEEE, 2005

[FINE; SCHEINBERG, 2002] FINE, S.; SCHEINBERG, K. "**INCAS: An Incremental Active Set Method for SVM**", IBM Technical report, 2002.

[FISHER, 1936] FISHER, R, "**The use of multiple measurements in taxonomic problems**", Ann. Eugen., 7, pp. 179–188, 1936

[FURLANELLO et. al, 2003] FURLANELLO, C.; SERAFINI, M.; MERLER, S.; JURMAN, G. "**Gene selection and classification by Entropy-based Recursive Feature Elimination**", In International Joint Conference on Neural Networks, Portland, Oregon, July 20-24, p. 3077-3082, 2003

[GENTILE, 2004] GENTILE, C. "**Fast Feature Selection from Microarray Expression Data via Multiplicative Large Margin Algorithms**", Advances in Neural Information Processing Systems 16, MIT Press, 2004

[GUYON, ELISSEEFF, 2003] GUYON, I.; ELISSEEFF, A., "**An Introduction to Variable and Feature Selection**", Journal of Machine Learning Research, vol. 3, pp. 1157-1182, 2003

[GUYON et. al, 2002] GUYON, I.; WESTON, J.; BARNHILL, S.; VAPNIK, V. N. "**Gene selection for cancer classification using Support Vector Machines**", Machine Learning, v. 46, p. 389–422, 2002

[HANS, et al., 2004] HANS, P. G; COSATTO, E.; BOTTOU, L.; DURDANOVIC, I.; VAPNIK, V. "**Parallel Support Vector Machines: The Cascade SVM**", Neural Information Processing Systems - NIPS, 2004

[HIROYAMA; MATSUMOTO, 2002] TAKAMURA, H.; MATSUMOTO, Y. "**Two-dimensional Clustering for Text Categorization**", Proceedings of the Sixth Conference on Natural Language Learning - CoNLL2002, Taipei, Taiwan, p. 29-35, 2002

[HONKELA, et. al., 1996] HONKELA, T.; KASKI, S.; LAGUS, K.; KOHONEN, T. "**Newsgroup Exploration with WEBSOM Method and Browsing Interface**", Helsinki University of Technology, Faculty of Information Technology, Laboratory of Computer and Information Science, Report A32, 1996

[JAPKOWICZ; MYERS; GLUCK, 1995] JAPKOWICZ, N.; MYERS, C.; GLUCK, M.A. "**A Novelty Detection Approach to Classification**", Proceedings of the Fourteenth Joint Conference on Artificial Intelligence, p. 518-523, 1995

[JOACHIMS, 1997] JOACHIMS, T. "**A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization**", In D.H. Fisher, editor, Proceedings of ICML'97, 14th International Conference on Machine Learning, Nashville, US, Morgan Kaufmann Publishers, San Francisco, US, p. 143–151, 1997

[JOACHIMS, 1998] JOACHIMS, T. "**Text categorization with suport vector machines: Learning with many relevant features**", In ECML '98: Proceedings of the 10th European Conference on Machine Learning (London, UK, 1998), Springer-Verlag, p. 137-142, 1998

[JOACHIMS, 1999] JOACHIMS, T. "**Making large-scale SVM learning practical**", In Bernhard Schölkopf, Christopher J. Burges, and Alexander J. Smola, editors, Advances in Kernel Methods – Support Vector Learning. The MIT Press, Cambridge, US, ch. 11, p. 169–184, 1999

[JOACHIMS, 2000] JOACHIMS, T. "**Estimating the Generalization Performance of an SVM Efficiently**", Proceedings of the International Conference on Machine Learning (ICML), Morgan Kaufman, 2000

[JOLLIFE, 1986] JOLLIFE, I., "**Principal components analysis**", Springer, Verlag, New York, 1986

[KATAGIRI; ABE, 2006] KATAGIRI, S.; ABE, S. "**Incremental Training of Support Vector Machines Using Hyper spheres**", Elsevier, 2006

[KEERTHI; LIN, 2003] KEERTHI, S. S.; LIN, C. H., "**Asymptotic Behaviors of Support Vector Machines with Gaussian Kernel**", Neural Computation, vol. 15, 2003

[KEERTHI et. al., 2001] KEERTHI, S. S.; SHEVADE, S. K.; BHATTACHARYYA, C.; MURTHY, K. R. K. "**Improvements to Platt's SMO Algorithm for SVM Classifier Design**", Neural Computation v. 13, p. 637–649, 2001

[KOHONEN et al, 2000] KOHONEN, T.; KASKI, S.; LAGUS, K.; SALOJARVI, J.; PAATERO, V.; SAARELA, A. "**Self Organization of a Massive Document Collection**", IEEE Transactions on Neural Networks, v. 11, p. 574-585, 2000

[KOHONEN, 2001] KOHONEN, T., "**Self-Organizing Maps**", Springer-Verlag, 3rd. ed. New York, USA, 2001

[LAAKSONEN; KOSKELA; OJA, 2002] LAAKSONEN, J.; KOSKELA, M.; OJA, E. "**PicSOM—Self-Organizing Image Retrieval with MPEG-7 Content Descriptors**", IEEE Transactions on Neural Networks, v. 13, no. 4, 2002

[LAGUS; KASKI; KOHONEN, 2004] LAGUS, K.; KASKI, S.; KOHONEN, T., "Mining massive document collections by the WEBSOM method", Information Sciences, Vol 163/1-3, pp. 135-156, 2004

[LANG, 1995] LANG, K. "**Newsweeder: Learning to filter netnews**", International Conference on Machine Learning (ICML), pp. 331–339, 1995

[LEHMANN et al., 2005] LEHMANN, T. M.; GULDA, M. O.; DESELAERSB, T.; KEYSERSB, D.; SCHUBERTC, H.; SPITZERA, K.; NEYB, H.; WEINC, B. B., "**Automatic categorization of medical images for content-based retrieval and data mining**", Computerized Medical Imaging and Graphics, v. 29, pp.143-155, 2005

[LEE; PARK; CHOI, 2001] LEE, K-S.; Park, Y-C.; Choi, K-S., "**Re-ranking model based on document clusters**", Information Processing and Management, vol. 37, 2001

[LEWIS, 1997] LEWIS, D. "**Reuters-21578 text categorization test collection**", (*http://www.research.att.com/~lewis*), 1997

[LI; HAN; PEI, 2001] LI, W.; HAN, J.; PEI, J. "**CMAR: Accurate and Efficient Classification Based on Multiple Class-Association Rules**", Proceedings IEEE International Conference on International Conference on Data Mining, p. 369-376, 2001.

[LIN; SOERGEL; MARCHIONINI, 1991] LIN, X.; SOERGEL, D.; MARCHIONINI, G. "**A self-organizing semantic map for information retrieval**", in Proc. 14th Ann. Int. ACM/SIGIR Conf., on R&D In Information Retrieval, p. 262-269, 1991

[LING; LEUNG; LAM, 2006] LING, S.H. LEUNG, F.H.F. LAM, H.K., "**A Variable Node-to-Node-Link Neural Network and Its Application to Hand-Written Recognition**", International Joint Conference on Neural Networks, 2006

[MANEVITZ; YOUSEF, 2001] MANEVITZ, L. M.; YOUSEF, M. "**One-Class SVMs for Document Classification**", Journal of Machine Learning Research v. 2, p. 139-154, 2001

[MANEVITZ; YOUSEF, 2006] MANEVITZ, L. M.; YOUSEF, M. "**One-class document classification via Neural Networks**", Neurocomputing, Elsevier Science Publishers, v. 70, p. 1466-1481, 2007

[MCHUGH; CHRISTIE; ALLEN, 2000] MCHUGH, J.; CHRISTIE, A.; ALLEN, J. "**Defending yourself: The role of intrusion detection systems**". IEEE Software, p. 42–51, 2000

[MITCHELL, 1997] MITCHELL, T. M. "**Machine Learning**", McGraw-Hill, 1997

[MORENO; HO; VASCONCELOS, 2004] MORENO, P. J.; HO, P. P.; VASCONCELOS, N.; "**A Kullback-Leibler Divergence Based Kernel for SVM Classification in Multimedia Applications**", IN ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS 16, MIT Press, 2004

[OSUNA; FREUND; GIROSI, 1997] OSUNA, E.; FREUND, R.; GIROSI, F. "**Improved training algorithm for support vector machines**", Neural Network for Signal Processing, 1997

[PERDISCI; GU; LEE, 2006] PERDISCI, R.; GU, G.; LEE, W. "**Using an Ensemble of One-Class SVM Classifiers to Harden Payload-based Anomaly Detection Systems**", Proceedings of the Sixth International Conference on Data Mining (ICDM'06), 2006

[PEREIRA; TISHBY; LEE, 1993] PEREIRA, F.; TISHBY, N.; LEE, L. "**Distributional clustering of English words**", In Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics, p. 183–190, 1993.

[PLATT, 1999] John C. PLATT, "**Fast training of support vector machines using sequential minimal optimization**", MIT Press Cambridge, MA, USA, pp. 185 – 208, 1999

[POMERLEAU, 1989] Pomerleau, D. A., "**ALVINN: Anautonomous land vehicle in a neural network**", Technical Report CMU-CS-89-107, Pittsburgh, PA: Carnegie Mellon University, 1989

[PORTER, 1980] PORTER, M.F. "**An Algorithm for Suffix Stripping**", Program, 14(3): 130-137, 1980

[QUINLAN; CAMERON-JONES, 1993] QUINLAN, J. R.; CAMERON-JONES, R. M. "**FOIL: A midterm report**", Springer-Verlag, In Proceedings of the European Conference on Machine Learning, p. 3-20, 1993

[RAHMAN et al., 2007] RAHMAN, M. M.; BHATTACHARYA, P.; DESAI, B. C., "**A framework for medical image retrieval using machine learning and statistical similarity matching techniques with relevance feedback**", IEEE Transactions on information technology in biomedicine, v. 11, n. 1, pp. 58-69, 2007

[RITTER; KOHONEN, 1989] RITTER, H.; KOHONEN, T., "**Self-organizing semantic maps**", Biological Cybernetics, v. 61 p. 241-254, 1989

[SADAT; YOSHIKAWA; SHUNSUKE, 2003] SADAT, F.; YOSHIKAWA, M.; Uemura, S., "**Learning Bilingual Translations from Comparable Corpora to Cross-Language Information Retrieval: Hybrid Statistics-based and Linguistics-based Approach**", In Proceedings of the Sixth International Workshop on Information Retrieval with Asian Languages IRAL, 2003

[SEBASTIANI, 2002] SEBASTIANI, F. "**Machine learning in automated text categorization**", ACM Computing Surveys, v. 34(1), p. 1–47, 2002

[SCHEINBERG, 2006] SCHEINBERG, K., "**An Efficient Implementation of an Active Set Method for SVMs**", MIT Press Cambridge, MA, USA, v.7, p. 2237–2257, 2006

[SCHÖLKOPF et. al, 1999] SCHÖLKOPF, B.; PLATT, J. C.; SHAWE-TAYLOR, J.; SMOLA, A.J.; Williamson, R. C. "**Estimating the support of a high-dimensional distribution**", Technical report, Microsoft Research, MSR-TR-99-87, 1999.

[SILVA et al., 2008] SILVA, L. A. da; DEL-MORAL-HERNANDEZ, E.; RANGAYYAN, R. M., "**Classification of breast masses using a committee machine of artificial neural networks**", Journal of Electronic Imaging, v. 17 n. 1, pp. 13-17, 2008

[SILVA, 2004] Silva, E. C. N., "**Otimização Aplicada ao Projeto de Sistemas Mecânicos**", EPUSP, notas de aula do curso PMR – 5215, 2004

[SOARES; PRATI; MONARD, 2009] SOARES, M. V. B.; PRATI, R. C.; MONARD, M. C., "Improvements on the Porter's Stemming Algorithm for Portugues", IEEE LATIN AMERICA TRANSACTIONS, VOL. 7, NO. 4, 2009

[SYED et al., 1999] SYED, N.; LIU, H.; SUNG, K. "**Incremental learning with support vector machines**", In Proceedings of the Workshop on Support Vector Machines at the International Joint Conference on Artificial Intelligence (IJCAI-99), Stockholm, Sweden, 1999

[TAX; DUIN, 1999] TAX, D.; DUIN, R. "***Data Domain Description using Support Vectors***", European Symposium on Artificial Neural Networks, 1999

[ZHUANG; DAI, 2006] ZHUANG, L.; DAI, H. "**Parameter Optimization of Kernel-Based One-Class Classifier on Imbalance Text Learning**", Lecture notes in computer science – Springer Verlag, n. 4099, pp. 434-443, 2006

[ZHANG; LIN; BO ZHANG, 2001] ZHANG, L.; LIN, F.; ZHANG, B. "**SUPPORT VECTOR MACHINE LEARNING FOR IMAGE RETRIEVAL**", IEEE International Conference on Image Processing (ICIP 2001), Thessaloniki, Greece, October, p. 721-724, 2001

[WAIBEL 1989] Waibel, A.; Hanazawa, T.; Hinton, G., "**Phoneme Recognition Using Time-Delay Neural Networks**", IEEE Transactions on Acoustics, Speech and Signal Processing , v. ASSP-37, n. 3 (March), pp. 328-339, 1989

[WANG; STOLFO, 2004] WANG, K.; STOLFO, S. "**Anomalous payload-based network intrusion detection**", In Recent Advances in Intrusion Detection (RAID), 2004

[WEI; YANG; NISHIKAWA, 2009] Wei, L.Y.; Yang, Y.Y.; Nishikawa, R.M., "**Microcalcification Classification Assisted by Content-Based Image Retrieval for Breast Cancer Diagnosis**", PR(42), No. 6, pp. 1126-1132, 2009

[TESAURO, 1992] Tesauro, G., "**Practical issues in temporal difference learning**", Machine Learning 8, pp. 257-277, 1992

[TESAURO, 1995] Tesauro, G., "**Temporal difference learning and TD-Gammon**", Comm. of the ACM 38:3, pp. 58-68, 1995

[TISHBY; PEREIRA; BIALEK, 1999] TISHBY, N.; PEREIRA, F.; BIALEK, W. "**The information bottleneck method**", Invited paper to the 37th annual Allerton Conference on Communication, Control, and Computing, 1999

[WEISS et al, 1999] WEISS, S. M.; APTÉ, C.; DAMERAU, F. J.; JOHNSON, D. E.; OLES, F. J.; GOETZ, T.; HAMPP, T. "**Maximizing text-mining performance**", IEEE Intelligent Systems, 14(4), p.63–69, 1999

[YANG, 1999] YANG, Y. "**An evaluation of statistical approaches to text categorization**", Information Retrieval, 1(1-2):69–90, 1999

[YANG; PEDERSEN, 1997] YANG, Y.; PEDERSEN, J. O. "**Comparative Study on Feature Selection in Text Categorization**", Proceedings of ICML-97, 14th International Conference on Machine Learning, 1997

[YIN; HAN, 2003] YIN, X.; HAN, J. "**CPAR: Classification based on predictive association rules**", Proceedings of the Third SIAM International Conference on Data Mining, 2003

# Apêndice A - A teoria da otimização

Este apêndice tem por finalidade complementar a teoria fundamental do SVM, a otimização matemática.

A teoria da otimização é um ramo da matemática encarregado de caracterizar classes de soluções de funções com algumas restrições e do desenvolvimento de algoritmos eficientes para o seu cálculo.

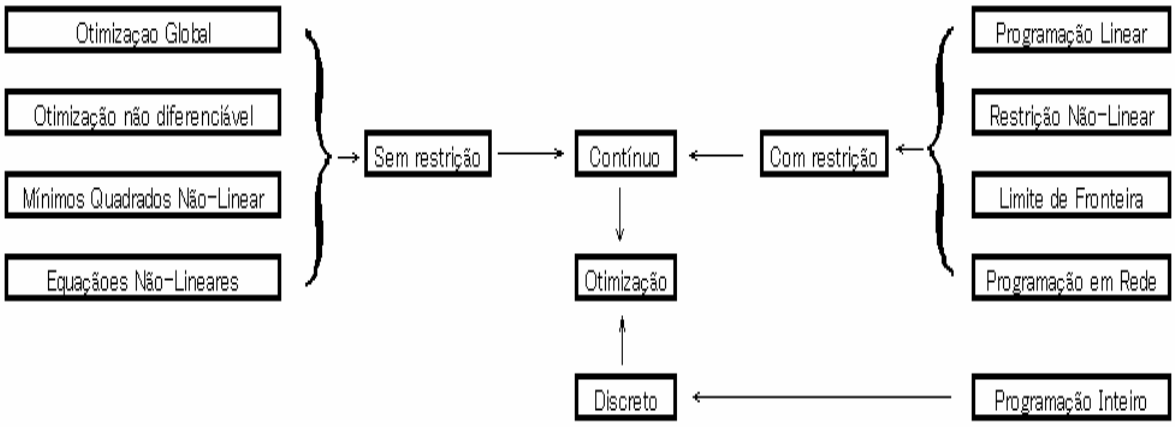



**Figura A.1. Árvore ilustrando a ciência da otimização. Adaptado de (ARGONNE, 2005).**

A Figura A.1 acima ilustra os diferentes tipos de otimização que existem. Eles se dividem basicamente em dois grandes grupos: aqueles cujas variáveis de projeto são contínuas e aqueles cujas variáveis de projeto são discretas. Uma variável de projeto é um parâmetro do problema que pode ser alterado para atender aos requisitos especificados. Em poucas palavras, otimizar significa encontrar o máximo ou mínimo de uma função, a função objetiva (ela quantifica o que desejamos otimizar em função das variáveis de projeto).

Finalmente todo problema de otimização pode ter restrições de desigualdade e restrições de igualdade. As restrições são funções que limitam o universo das soluções possíveis, elas definem então o domínio viável, como mostra a Figura A.2.

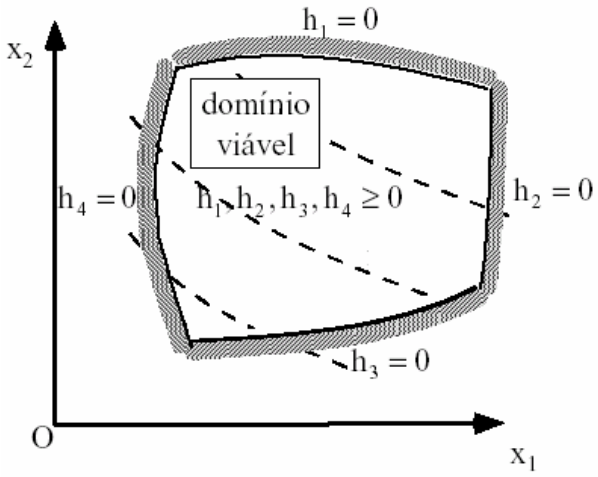



**Figura A.2. Regiões de domínio viável (fonte: notas de aula Prof. Dr. Emílio Silva [SILVA, 2004]). O domínio viável corresponde o interior da região delimitada pelas superfícies $h_1$, $h_2$, $h_3$ e $h_4$. As linhas tracejadas são as curvas de níveis.**

Na Figura A.2 $h_n$ (onde n é igual a 1, 2, 3 ou 4) representam as restrições que formam o domínio viável em um exemplo bidimensional.

Quando as máquinas de vetores-suporte (SVM) originalmente foram apresentadas por Vapnik em 1979, um problema sério deveria ser contornado, a questão da otimização. Uma das formas para resolver esse problema era através da programação quadrática (*quadratic programming* - QP). Além da QP existem outras técnicas para encontrar os vetores-suporte de um SVM, por exemplo, a programação linear (*linear programming* - LP) também é muito usada. Maiores detalhes nas próximas Seções onde também será apresentada a base matemática de um SVM, fundamentada fortemente na teoria da otimização.

Existem diversos pacotes (*softwares* e bibliotecas para *softwares*) para resolução da QP ou LP. Veja abaixo:

- CPLEX, da ILOG é um *software* comercial que resolve problemas de otimização, tanto QP quanto LP, usando diversos métodos dentre os quais se podem destacar o *CPLEX Simplex Optimizers*
  http://www.ilog.com
- MINOS, da *Stanford Business Software Inc.* em sua versão 5.5 é um pacote de *software* desenvolvido por Bruce Murtagh e Michael Saunders que resolve problemas de otimização em larga escala.
  http://www.sbsi-sol-optimize.com

- LOQO, desenvolvido por Robert J. Vanderbei da Universidade de Princeton está em sua versão 6.07. LOQO é um *software* que resolve problemas de otimização suave (linear, não linear, convexa, não convexa, restrita e irrestrita).
  http://www.princeton.edu/~rvdb
- KNITRO, da Ziena é um *software* pare resolução de problemas de otimização desenvolvida por Jorge Nocedal da Northwestern *University* e colaboradores.
  http://www.ziena.com/knitro.html

MINOS e LOQO apresentam algumas restrições (BOTTOU, 2007) quanto ao tamanho da matriz a armazenar, além do tempo para convergência à solução ser mais demorada.

## O problema de otimização (*primal*)

SVM tenta resolver um problema de otimização dual (veja Capítulo 2). Veja nesta seção a analogia entre um problema primal e dual.

Dada uma função f, $g_i$, i = 1,... , k, e $h_i$, i = 1,.... , m definidas no domínio $\Omega \subseteq R^n$, queremos,

$$\min imizar \quad f(\mathbf{w}) \qquad \textbf{Eq. A.1}$$

Sujeito às seguintes restrições:

$g_i(\mathbf{w}) \leq 0$, i = 1, ..... , k, (restrição de desigualdade)
$h_i(\mathbf{w}) = 0$, i = 1, …. , m (restrição de igualdade)

Sendo $\mathbf{w} \in \Omega$,

Onde $f(\mathbf{w})$ é chamada de função objetiva, e as relações seguintes são chamadas respectivamente de restrições de desigualdade e restrições de igualdade. O valor ótimo da função objetiva é chamado de valor do problema de otimização.

A região no domínio onde a função objetiva está definida e onde todas as restrições são satisfeitas é chamada de domínio viável (*feasible region)* e será denotada por:

$$R = \{\mathbf{w} \in \Omega : g(\mathbf{w}) \leq 0, h(\mathbf{w}) = 0\}$$

Por simplicidade foram omitidos os índices i de ambas as funções g e h.

A solução do problema de otimização é um ponto $\mathbf{w}^* \in$ R tal que, $f(\mathbf{w}^*) < f(\mathbf{w})$, para $\mathbf{w}$ diferente de $\mathbf{w}^*$. Esse ponto também é chamado de ponto de mínimo global quando f for uma função convexa (ver adiante).

Quando uma função objetiva for linear, e suas restrições de desigualdade e igualdade também, então temos um problema de programação linear (PL), mas quando a função objetiva é do tipo quadrática (com restrições lineares), temos uma programação quadrática (PQ).

**Definição:** função convexa

Uma função de valores reais $f(\mathbf{w})$ é dita convexa para $\mathbf{w} \in \Re^n$ se para $\forall$ w, u $\in$ a $\Re^n$, e para qualquer $\Theta \in (0,1)$,

$$f(\boldsymbol{\theta} \cdot \mathbf{w} + (1-\boldsymbol{\theta})u) \leq \boldsymbol{\theta} \cdot f(w) + (1-\boldsymbol{\theta}) f(u) \qquad \textbf{Eq. A.2}$$

Uma função que for duas vezes diferençável é convexa se a sua matriz Hessiana for positiva semi-definida.

A função objetiva de um SVM é do tipo convexo, veja e a Equação A.2, assim quando se encontra uma solução ótima temos certeza de que ela é ótima globalmente.

**Definição:** função afim

Uma função afim é um tipo especial de função linear dado por:

$$f(\mathbf{w}) = A\mathbf{w} + b \qquad \textbf{Eq. A.3}$$

Onde A é uma matriz e b, um vetor diferente de nulo. Veja que a função discriminante de um SVM (Equação A.3) é uma função afim, assim as propriedades de uma função afim são válidas para a função discriminante de um SVM.

Uma função afim é convexa porque ela tem Hessiana zero (definição).

Se uma função *f* for convexa, qualquer mínimo local $\mathbf{w}^*$ de um problema de otimização não restrito com função objetiva f é também um mínimo global, porque para qualquer $\mathbf{u} \neq \mathbf{w}^*$, pela definição de um mínimo local, existe θ suficientemente perto de 1 em que,

$$f(\mathbf{w}^*) \le f(\boldsymbol{\theta}\cdot\mathbf{w}^* + (1-\boldsymbol{\theta})\mathbf{u}) \le \boldsymbol{\theta}\cdot f(\mathbf{w}^*) + (1-\boldsymbol{\theta}) f(\mathbf{u}) \quad \textbf{Eq. A.4}$$

Da Equação acima, chegamos à seguinte conclusão:

$$f(\mathbf{w}^*) < f(\mathbf{u}) \quad \textbf{Eq. A.5}$$

## Teorema de Fermat

De acordo com o teorema de Fermat, desenvolvido em 1629, uma condição necessária para $\mathbf{w}^*$ ser o mínimo de f($\mathbf{w}$), f $\in$ $C^1$, é $\frac{\partial f(\mathbf{w}^*)}{\partial \mathbf{w}}$ =0. Esta condição, juntamente com a convexidade de *f*, é também uma condição suficiente (CRISTIANINI; TAYLOR, 2000).

## Método dos multiplicadores de Lagrange

O método dos multiplicadores de Lagrange foi desenvolvido por Lagrange em 1797 para problemas na área da mecânica; é uma generalização do resultado de Fermat (1629), tratando apenas das restrições de igualdade. Em 1951, Kuhn e Tucker estenderam o método (teoria de Kuhn Tucker), incluindo as restrições de desigualdade (CRISTIANINI; TAYLOR, 2000).

Em problemas com restrições, deve-se definir uma função, denominada de lagrangeano, que incorpora informações da função objetiva e das restrições (um lagrangeano é a soma da função objetiva com uma combinação linear das restrições em que seus coeficientes são chamados de multiplicadores de Lagrange).

Considere o seguinte problema:

$$\textit{mínimo} \quad F(\mathbf{x}) \quad \textbf{Eq. A.6}$$

Tal que $h_j(\mathbf{x}) = 0$

$j=1, \ldots, n_e$ $\quad \mathbf{x}=(x_1, \ldots, x_n)^T$

$n_e < n$ (número de equações menor que o número de incógnitas)

dF pode então ser expresso por:

$$dF = \left(\frac{\delta F}{\delta \mathbf{x}_1}\right) d\mathbf{x}_1 + \left(\frac{\delta F}{\delta \mathbf{x}_2}\right) d\mathbf{x}_2 + \cdots + \left(\frac{\delta F}{\delta \mathbf{x}_n}\right) d\mathbf{x}_n = 0 \qquad \textbf{Eq. A.7}$$

Quando não houvesse restrições (ou seja, as variáveis $x_1$, $x_2$, ... , $x_n$ são independentes uma das outras), o valor mínimo de dF = 0 seria obtido quando todas as derivadas parciais de dF fossem zero.

Com a introdução das restrições, elas devem ser removidas de alguma forma para que se possa resolver o problema de forma simples. Isso é possível multiplicando-se o dh por um coeficiente λ, e somando a df.

$$L = dF + \lambda dh \qquad \textbf{Eq. A.8}$$

A nova função L é chamada de função Lagrangeano.

No caso de um SVM, a função Lagrangeano seria expressa por:

$$L(\mathbf{w};\alpha) = \frac{\|\mathbf{w}\|^2}{2} + \sum_{t=1}^{N} \alpha_i \left(1 - y_t \left[b + \mathbf{x}_t^T \mathbf{w}_t\right]\right) \qquad \textbf{Eq. A.9}$$

Onde α são os multiplicadores de Lagrange.

Dada uma função Lagrangeano com suas restrições de igualdade $h_t(w) = 0$ para t=1, ... , N:

$$L(\mathbf{w}, \alpha) = F(\mathbf{w}) + \sum_{t=1}^{N} \alpha_t h_t(\mathbf{w}) \qquad \textbf{Eq. A.10}$$

Os coeficientes $\alpha_i$ são chamados de multiplicadores de Lagrange.

"Dado que a função é convexa e a restrição é linear, o ponto de mínimo será uma tangente da restrição linear sob a função objetiva". Assim, se um $w^*$ é um mínimo local para um problema de otimização com apenas restrições de igualdade, seria possível obter um $\frac{\partial F(\mathbf{w}^*)}{\partial \mathbf{w}} \neq 0$, mas a direção para a qual *f* reduz viola uma ou mais condição das restrições.

Para respeitar a condição de restrição de igualdade $h_t$, deve-se mover perpendicularmente a $\frac{\partial h_t(\mathbf{w}^*)}{\partial \mathbf{w}}$; assim, para satisfazer a todas as restrições, deve-se mover-se perpendicularmente ao sub-espaço V (*spanned*) estendido por:

$$\left\{ \frac{\partial h_t(\mathbf{w}^*)}{\partial \mathbf{w}} : t = 1,....., N \right\}$$

Se $\frac{\partial h_t(\mathbf{w}^*)}{\partial \mathbf{w}}$ for linearmente independente, nenhum movimento legal pode alterar o valor da função objetiva, enquanto $\frac{\partial f(\mathbf{w}^*)}{\partial \mathbf{w}}$ estiver no subespaço V; em outras palavras, enquanto existir βt, tal que,

$$\frac{\partial f(\mathbf{w}^*)}{\partial \mathbf{w}} + \sum_{t=1}^{N} \alpha_t h_t(\mathbf{w}^*) = 0 \qquad \textbf{Eq. A.11}$$

Retomando agora a Equação A.9, derivando-a em relação ao vetor ***w*** de forma que a sua derivada seja zero, pode-se obter o valor ótimo de ***w***.

$$\frac{\partial}{\partial w_1} L(\mathbf{w};\alpha) = w_1 - \sum_{t=1}^{N} \alpha_t y_t \mathbf{x}_t = 0 \qquad \textbf{Eq. A.12}$$

$$\frac{\partial}{\partial b} L(\mathbf{w};\alpha) = -\sum_{t=1}^{N} \alpha_t y_t = 0 \qquad \textbf{Eq. A.13}$$

Onde ***w*** é um dos componentes que será obtido do treino de um SVM.

## O problema de otimização (dual)

O SVM tenta resolver um problema de otimização dual. Um problema "re-escrito" em sua forma dual muitas vezes é mais fácil para resolvê-lo.

O problema dual da Equação A.6 pode ser expresso por:

$$\text{maximizar} \quad \Theta(\alpha, \beta) \qquad \textbf{Eq. A.14}$$

Sujeito a, $\alpha \geq 0$

Onde $\Theta(\alpha,\beta)$ é o $\inf_{w \in \Omega} L(w, \alpha,\beta)$. O valor da função objetiva no ponto de solução ótima é chamado de o valor do problema. A vantagem de se converter um problema para o seu dual está na possibilidade de inverter a relação entre as quantidades de restrição (n restrições) e variáveis de projeto (m variáveis de projeto). Assim, se no problema *primal* m > n, no dual, a relação se inverte para n > m.

O problema dual da Equação A.9 é da seguinte forma:

$$\max_{\alpha} W(\alpha) = \max_{\alpha} \left\{ \min_{w,b} \{L(\mathbf{w};\alpha)\} \right\} \qquad \textbf{Eq. A.15}$$

e o mínimo do lagrangeano com relação a **w** e b podem ser vistos, respectivamente, nas Equações A.12 e A.13.

Assim, das Equações A.12, A.13 e A.15, resulta o seguinte problema dual:

$$\max_{\alpha} W(\alpha) = \max_{\alpha} \left\{ -\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (\mathbf{x_i} \cdot \mathbf{x_j}) + \sum_{i=1}^{n} \alpha_i \right\} \qquad \textbf{Eq. A.16}$$

# Apêndice B – Lógica Proposicional

Definimos disjunção e conjunção de conjuntos da seguinte maneira:

- Disjunção, também conhecido como operador "ou", é um operador lógico que representa o conceito de união de conjuntos numéricos. Ela é representada pelo símbolo V.
- Conjunção, também conhecido como operador "e", é um operador lógico que representa o conceito de interseção. Ela é representada pelo símbolo Λ

Agora, podemos definir a Forma Normal Conjuntiva – CNF:

Uma formula ϕ é uma CNF-fórmula (do inglês, *conjunctive normal form*) se ϕ é uma conjunção de disjunções de literais.

Assim, $\phi = \phi_1 \wedge \phi_2 \wedge \phi_3 \cdots \wedge \phi_n$ onde $\phi_i = \varpi_1 \vee \varpi_2 \vee \varpi_3 \cdots \vee \varpi_m$

Uma literal $\varpi_j$ é uma variável ou a sua negação ($\overline{\varpi}_j$).

## Apêndice C – Formula Sherman-Morrison-Woodbury

A fórmula **Sherman-Morrison-Woodbury** é usada na inversão de matrizes quando já se tem uma inversão e deseja-se acrescentar novos parâmetros à matriz já invertida.
Seja A uma matriz n por n e inversível e seja U e V matrizes m por n com m ≤ n. Então:

$A+UV^t$ é inversível se, e somente se, $I_n+V^tA^{-1}U$ for inversível, com

$$\left(A+UV^t\right)^{-1}=A^{-1}-A^{-1}U\left(I_n+V^tA^{-1}U\right)^{-1}V^tA^{-1}$$

Ou seja, tendo $A^{-1}$, quero uma nova matriz inversa de A somado/subtraído alguns valores, $A+UV^t$. Para obter a inversa dessa nova matriz, podemos usar a fórmula SMW.

Já no contexto do SVM (um problema de classificação de duas classes), imagine m amostras no espaço n-dimensional $\Re^n$ representado por uma matriz E m x n. Seja D uma matriz diagonal m por m contendo a classificação de cada uma das m amostras em E, sendo +1 e -1 a representação das duas classes.

Então temos:

$H=D\left[E-e\right]$, com D = m x m, E = m x n => D*E = m x n

$Q=\frac{I}{\nu}+HH'$, com H = m x n, H*H' = m x m

Pela equação SMW modificado, temos que:

$$\left(\frac{I}{\nu}+HH'\right)^{-1}=v\left(I-H\left(\frac{I}{v}+H'H\right)^{-1}H'\right)$$

Precisamos calcular a inversa de uma matriz n x n

# Apêndice D – Termos técnicos usados em redes de comunicação

*Payload*, ou carga útil (essencial ou fundamental, maiores detalhes no Apêndice D), em protocolos de comunicação refere-se a menor unidade de informação (geralmente agrupados em pacotes) do dado real sendo transmitido. Ele é seguido por um cabeçalho que identifica o transmissor e o receptor (protocolo TCP/IP) do dado sendo transportado e é logo descartado assim que chega ao destinatário.

O *payload* da rede é apenas um conjunto de fluxo de bytes. E ao contrário de cabeçalho de pacotes, o *payload* não apresenta um formato fixo e também não apresenta faixa de valores fixo. Qualquer caractere ou byte poderá aparecer em qualquer posição do fluxo do datagrama (ou pacote).

Para iniciar o processo de modelagem de um *payload*, precisamos agrupá-los seguindo algum critério. Por exemplo, as informações que trafegam na porta 80 apresentam características diferentes daquelas que trafegam na porta 22, nesta última, as informações são encriptadas.

Além da informação do número da porta, o fluxo de informação poderá ser de entrada (*inbound*) ou de saída (*outbound*). Finalmente, o tamanho do *payload* também poderá ser diferente, mesmo utilizando-se da mesma porta.

Porta é um termo utilizado para indicar um número inteiro utilizado na comunicação entre sistemas de comunicação que se seguem o protocolo TCP/IP. Através da identificação das portas, os sistemas finais podem realizar comunicação ponto a ponto.

## Análise n-grama

Para facilitar a modelagem do *payload*, é comum utilizar-se da técnica n-grama. Em (WANG; STOLFO, 2004) foi realizado um estudo do n-grama através da distribuição dos valores de bytes. Um n-grama é uma sequência de n bytes adjacentes numa unidade de *payload*.

Uma janela deslizante é passada por todos os bytes do *payload* e então a ocorrência de cada n-grama é contabilizada. Assim, constrói se o vetor de atributos que representa um *payload*. Cada elemento desse vetor representa um n-grama e seu valor é um número relativo ao total de n-grama.

Wang e Stolfo, (2004) utilizaram-se do modelo de 1-grama, calculando a média e a variância de cada atributo. O modelo de 1-grama representa os caracteres ASCII, de 0 a 255.

# Apêndice E – Representação dos atributos – 20Newsgroup

Neste Apêndice estão listados os resultados de treinamento e teste realizados usando-se a base de dados 20Newsgroup (LANG, 1995).

O objetivo dos testes aqui apresentados era escolher a melhor representação dos documentos. Foram usadas 3 representações a NNN, NTC e a ATN. Maiores detalhes em (SALTON, 1971) e também na Seção 5.2.

A coluna RBF diz respeito ao parâmetro de dispersão da função kernel usada no SVM. Foram testados, exaustivamente, diversos valores. A coluna "Treino (+)" representa os resultados de validação do classificador treinado, usando as próprias amostras de treino. O símbolo "+" indica amostras da classe positiva, e o símbolo "-", amostras da classe negativa. Já a coluna "Teste (+)" representa os resultados de validação de amostras não usadas no treinamento, mas já previamente identificadas como sendo da classe positiva ou negativa. A coluna F1 representa a média harmônica entre os valores de precisão e revocação (detalhes na Seção 5.2).

O conjunto de tabelas será representado com uma legenda principal para facilitar a localização das informações pelo índice de tabelas.

**Tabela E.1 Classe 01, representação dos documentos utilizando a combinação ATN.**

**Classe 01 vs Classe 02**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 98,956 | 95,541 | 100 | 83,333 | 0,886 |
| **2** | 97,912 | 86,624 | 99,591 | 95,987 | 0,915 |
| **3** | 98,329 | 88,853 | 100 | 95,679 | **0,923** |
| **4** | 98,121 | 85,031 | 99,795 | 96,913 | 0,912 |
| **5** | 98,329 | 86,942 | 100 | 95,987 | 0,916 |
| **6** | 98,329 | 87,579 | 100 | 95,987 | 0,919 |
| **7** | 98,121 | 87,261 | 100 | 95,987 | 0,918 |
| **8** | 98,121 | 87,579 | 100 | 95,987 | **0,919** |
| **9** | 97,912 | 86,942 | 100 | 95,987 | 0,916 |
| **10** | 97,912 | 86,942 | 100 | 97,839 | **0,926** |

**Classe 1 vs Classe 7**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 99,164 | 94,267 | 99,800 | 82,078 | 0,897 |
| **2** | 98,538 | 87,261 | 100 | 96,057 | **0,915** |
| **3** | 98,329 | 83,439 | 100 | 98,566 | **0,903** |
| **4** | 98,329 | 82,484 | 100 | 98,207 | 0,896 |
| **5** | 97,077 | 84,394 | 96,414 | 93,906 | **0,889** |
| **6** | 98,121 | 82,484 | 100 | 98,207 | 0,896 |
| **7** | 97,703 | 81,528 | 100 | 98,566 | 0,892 |
| **8** | 97,077 | 80,891 | 100 | 99,283 | 0,891 |
| **9** | 97,077 | 81,847 | 100 | 98,566 | 0,893 |
| **10** | 96,868 | 80,573 | 100 | 98,924 | 0,887 |

**Classe 01 vs Classe 10**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 99,165 | 93,949 | 99,818 | 74,084 | 0,833 |
| **2** | 98,747 | 84,713 | 100 | 93,194 | 0,878 |
| **3** | 98,329 | 81,210 | 100 | 96,335 | 0,875 |
| **4** | 98,747 | 82,166 | 99,818 | 93,980 | 0,867 |
| **5** | 98,121 | 81,847 | 100 | 95,550 | 0,874 |
| **6** | 98,121 | 83,439 | 100 | 95,288 | **0,882** |
| **7** | 97,495 | 84,076 | 100 | 96,073 | **0,890** |
| **8** | 97,286 | 82,484 | 97,996 | 92,932 | 0,863 |
| **9** | 97,077 | 82,166 | 100 | 96,597 | **0,882** |
| **10** | 97,286 | 80,255 | 100 | 97,120 | 0,874 |

**Classe 01 vs Classe 14**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 98,538 | 93,630 | 100 | 73,964 | 0,844 |
| **2** | 98,329 | 87,261 | 100 | 91,716 | **0,889** |
| **3** | 98,329 | 84,076 | 100 | 94,970 | 0,887 |
| **4** | 98,329 | 83,121 | 100 | 95,266 | 0,883 |
| **5** | 98,121 | 82,802 | 100 | 96,153 | 0,885 |
| **6** | 98,121 | 83,121 | 100 | 96,153 | 0,887 |
| **7** | 97,912 | 84,394 | 100 | 95,858 | **0,893** |
| **8** | 97,286 | 83,439 | 100 | 96,745 | **0,892** |
| **9** | 97,286 | 82,484 | 100 | 96,449 | 0,885 |
| **10** | 97,286 | 81,528 | 100 | 96,745 | 0,881 |

**Classe 01 vs Classe 18**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 98,956 | 92,993 | 100 | 46,049 | 0,726 |
| **2** | 98,538 | 86,942 | 100 | 80,926 | 0,831 |
| **3** | 98,329 | 83,121 | 100 | 91,553 | 0,861 |
| **4** | 98,329 | 82,165 | 100 | 94,277 | **0,870** |
| **5** | 98,329 | 80,573 | 99,816 | 93,733 | 0,857 |
| **6** | 97,912 | 80,891 | 99,816 | 95,095 | 0,866 |
| **7** | 97,703 | 80,254 | 99,816 | 96,457 | 0,870 |
| **8** | 97,494 | 80,254 | 100 | 97,275 | **0,875** |
| **9** | 97,286 | 79,936 | 100 | 97,002 | **0,871** |
| **10** | 97,077 | 75,477 | 100 | 95,912 | 0,837 |

**Tabela E.2 Classe 01, representação dos documentos utilizando a combinação NTC.**

**Classe 01 vs Classe 02**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 92,760 | 80,935 | 99,350 | 90,196 | **0,914** |
| **2** | 95,701 | 90,287 | 88,961 | 68,627 | 0,797 |
| **3** | 92,081 | 82,733 | 98,051 | 89,215 | **0,912** |
| **4** | 95,475 | 91,007 | 94,155 | 75,490 | 0,844 |
| **5** | 100 | 100 | 20,129 | 7,843 | 0,145 |
| **6** | 100 | 100 | 17,532 | 6,862 | 0,128 |
| **7** | 88,235 | 78,417 | 98,701 | 92,156 | **0,921** |
| **8** | 95,248 | 89,568 | 89,610 | 73,529 | 0,829 |
| **9** | 100 | 100 | 9,090 | 3,921 | 0,075 |
| **10** | 100 | 100 | 6,493 | 2,941 | 0,057 |

**Classe 01 vs Classe 07**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 88,687 | 73,381 | 97,491 | 90,243 | 0,832 |
| **2** | 90,271 | 82,014 | 87,096 | 81,707 | **0,875** |
| **3** | 80,090 | 68,345 | 98,207 | 96,341 | 0,806 |
| **4** | 89,819 | 82,733 | 85,304 | 84,146 | **0,882** |
| **5** | 75,339 | 60,071 | 98,566 | 98,780 | 0,748 |
| **6** | 74,434 | 57,913 | 98,566 | 97,561 | 0,730 |
| **7** | 85,067 | 76,978 | 93,190 | 89,024 | 0,854 |
| **8** | 71,493 | 56,834 | 98,566 | 98,780 | 0,723 |
| **9** | 73,755 | 59,712 | 98,566 | 97,561 | 0,744 |
| **10** | 90,724 | 82,374 | 86,738 | 80,487 | **0,875** |

**Classe 01 vs Classe 10**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 91,629 | 86,690 | 86,055 | 83,043 | 0,863 |
| **2** | 86,425 | 70,863 | 99,601 | 97,826 | 0,820 |
| **3** | 92,760 | 84,892 | 92,031 | 88,260 | **0,872** |
| **4** | 79,411 | 63,309 | 99,203 | 98,695 | 0,770 |
| **5** | 92,760 | 84,532 | 88,446 | 87,391 | **0,867** |
| **6** | 81,674 | 75,179 | 94,820 | 92,608 | 0,829 |
| **7** | 93,438 | 86,690 | 87,649 | 86,956 | **0,878** |
| **8** | 73,755 | 59,712 | 99,601 | 99,130 | 0,744 |
| **9** | 73,981 | 59,712 | 99,601 | 99,130 | 0,744 |
| **10** | 94,796 | 85,971 | 86,055 | 81,739 | 0,855 |

**Classe 01 vs Classe 14**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 88,914 | 73,741 | 99,601 | 89,781 | 0,824 |
| **2** | 85,067 | 71,942 | 99,601 | 91,970 | 0,818 |
| **3** | 90,724 | 83,453 | 89,641 | 82,481 | **0,868** |
| **4** | 77,828 | 64,388 | 99,601 | 92,700 | 0,766 |
| **5** | 76,018 | 62,589 | 99,601 | 92,700 | 0,753 |
| **6** | 90,950 | 82,374 | 86,055 | 79,562 | **0,856** |
| **7** | 90,950 | 81,654 | 86,454 | 78,832 | **0,850** |
| **8** | 86,199 | 76,978 | 96,812 | 89,781 | 0,845 |
| **9** | 98,868 | 98,201 | 13,147 | 6,569 | 0,804 |
| **10** | 80,316 | 69,064 | 98,804 | 92,700 | 0,800 |

**Classe 01 vs Classe 18**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 80,769 | 62,230 | 97,861 | 90,942 | 0,726 |
| **2** | 80,543 | 67,625 | 98,395 | 90,942 | 0,765 |
| **3** | 88,009 | 78,777 | 93,850 | 88,405 | **0,828** |
| **4** | 75,339 | 60,791 | 98,128 | 93,840 | 0,728 |
| **5** | 87,782 | 82,374 | 91,978 | 89,130 | **0,852** |
| **6** | 87,330 | 81,295 | 90,107 | 82,971 | **0,820** |
| **7** | 67,420 | 53,237 | 97,593 | 94,927 | 0,672 |
| **8** | 71,493 | 57,913 | 97,326 | 94,202 | 0,707 |
| **9** | 59,502 | 45,323 | 98,663 | 97,101 | 0,611 |
| **10** | 89,140 | 83,812 | 84,492 | 74,275 | 0,800 |

**Tabela E.1 Classe 01, representação dos documentos utilizando a combinação NNN.**

**Classe 01 vs Classe 02**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 97,737 | 91,366 | 86,363 | 65,686 | 0,778 |
| **2** | 96,832 | 89,208 | 91,558 | 67,647 | 0,788 |
| **3** | 95,022 | 88,129 | 87,010 | 71,568 | 0,813 |
| **4** | 95,475 | 83,812 | 95,454 | 84,313 | **0,886** |
| **5** | 95,701 | 90,647 | 95,454 | 77,451 | 0,856 |
| **6** | 94,343 | 85,971 | 97,402 | 79,411 | 0,860 |
| **7** | 90,271 | 84,172 | 88,311 | 67,647 | 0,780 |
| **8** | 92,081 | 80,575 | 96,753 | 77,451 | 0,839 |
| **9** | 94,796 | 83,812 | 98,051 | 81,372 | **0,868** |
| **10** | 94,796 | 84,172 | 96,753 | 81,372 | **0,869** |

**Classe 01 vs Classe 07**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 73,981 | 26,978 | 98,207 | 95,122 | 0,420 |
| **2** | 95,022 | 75,539 | 88,17 | 76,829 | 0,828 |
| **3** | 94,796 | 79,136 | 98,566 | 81,707 | 0,857 |
| **4** | 95,701 | 86,330 | 79,569 | 63,414 | 0,875 |
| **5** | 93,891 | 80,575 | 94,265 | 76,829 | 0,859 |
| **6** | 92,533 | 80,215 | 98,207 | 85,365 | 0,869 |
| **7** | 96,606 | 90,647 | 78,136 | 51,219 | **0,884** |
| **8** | 91,402 | 82,733 | 96,057 | 84,146 | **0,882** |
| **9** | 95,248 | 90,647 | 53,046 | 35,365 | 0,864 |
| **10** | 86,651 | 80,215 | 95,698 | 90,243 | **0,876** |

**Classe 01 vs Classe 10**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 96,606 | 88,129 | 92,031 | 72,173 | 0,834 |
| **2** | 94,796 | 81,295 | 86,852 | 80,869 | 0,824 |
| **3** | 92,986 | 79,856 | 86,454 | 83,043 | 0,823 |
| **4** | 92,533 | 81,295 | 90,836 | 85,652 | 0,841 |
| **5** | 94,796 | 80,215 | 93,227 | 87,391 | 0,841 |
| **6** | 92,081 | 79,496 | 97,609 | 92,173 | **0,854** |
| **7** | 91,62 | 80,935 | 93,227 | 89,565 | **0,853** |
| **8** | 93,665 | 82,374 | 88,844 | 86,521 | 0,851 |
| **9** | 89,592 | 76,978 | 98,406 | 96,521 | **0,856** |
| **10** | 91,62 | 78,417 | 92,031 | 90,434 | 0,841 |

**Classe 01 vs Classe 14**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 97,058 | 96,762 | 49,800 | 21,897 | 0,822 |
| **2** | 92,986 | 83,812 | 98,804 | 77,372 | **0,859** |
| **3** | 96,380 | 92,44 | 39,442 | 43,795 | 0,839 |
| **4** | 94,570 | 83,093 | 85,657 | 67,883 | 0,835 |
| **5** | 96,832 | 91,007 | 64,143 | 58,394 | 0,860 |
| **6** | 89,592 | 78,77 | 95,617 | 88,321 | 0,853 |
| **7** | 91,855 | 79,856 | 99,601 | 87,591 | **0,858** |
| **8** | 95,022 | 86,330 | 74,103 | 61,313 | 0,840 |
| **9** | 90,950 | 80,935 | 98,804 | 86,861 | **0,863** |
| **10** | 86,877 | 76,25 | 92,031 | 86,861 | 0,834 |

**Classe 01 vs Classe 18**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 93,665 | 79,136 | 96,524 | 60,869 | 0,7261 |
| **2** | 47,285 | 17,985 | 97,058 | 97,101 | 0,2976 |
| **3** | 93,665 | 82,374 | 93,048 | 63,768 | 0,7545 |
| **4** | 92,081 | 79,856 | 99,465 | 78,623 | **0,7943** |
| **5** | 94,117 | 85,251 | 91,711 | 67,391 | 0,7835 |
| **6** | 89,81 | 75,899 | 91,176 | 72,826 | 0,7482 |
| **7** | 88,687 | 77,697 | 95,989 | 81,159 | **0,7912** |
| **8** | 93,891 | 85,611 | 73,796 | 58,333 | 0,7544 |
| **9** | 83,257 | 74,100 | 91,711 | 85,507 | 0,7863 |
| **10** | 85,972 | 76,978 | 93,048 | 82,971 | **0,7941** |

**Tabela E.2 Classe 02, representação dos documentos utilizando a combinação ATN.**

**Classe 02 vs Classe 01**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 99,646 | 96,195 | 84,601 | 19,293 | 0,3149 |
| **2** | 99,823 | 92,391 | 84,424 | 67,119 | 0,7732 |
| **3** | 96,991 | 84,239 | 83,362 | 82,065 | 0,8395 |
| **4** | 99,646 | 90,489 | 84,424 | 78,260 | 0,8398 |
| **5** | 99,469 | 88,587 | 84,424 | 78,260 | 0,8326 |
| **6** | 99,292 | 91,576 | 84,247 | 79,347 | **0,8508** |
| **7** | 99,115 | 90,489 | 84,247 | 79,619 | 0,8482 |
| **8** | 98,230 | 89,402 | 83,716 | 81,250 | **0,8539** |
| **9** | 98,761 | 90,217 | 83,362 | 79,891 | **0,8488** |
| **10** | 98,230 | 87,771 | 83,716 | 79,891 | 0,8395 |

**Classe 02 vs Classe 07**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 99,469 | 94,565 | 95,378 | 30,973 | 0,7326 |
| **2** | 99,115 | 87,771 | 87,061 | 70,501 | 0,8167 |
| **3** | 99,115 | 85,597 | 98,521 | 77,286 | **0,8289** |
| **4** | 97,876 | 84,510 | 97,227 | 78,466 | 0,8271 |
| **5** | 93,982 | 73,369 | 98,151 | 87,610 | 0,7941 |
| **6** | 97,345 | 82,337 | 96,672 | 80,826 | 0,8234 |
| **7** | 91,681 | 75,815 | 97,227 | 86,430 | 0,8052 |
| **8** | 95,221 | 82,337 | 91,312 | 79,056 | 0,8167 |
| **9** | 93,628 | 80,706 | 96,303 | 88,495 | **0,8437** |
| **10** | 93,451 | 81,521 | 97,042 | 87,020 | **0,8427** |

**Classe 02 vs Classe 10**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 100 | 36,413 | 100 | 94,162 | 0,5105 |
| **2** | 100 | 91,847 | 100 | 79,695 | 0,8601 |
| **3** | 99,469 | 90,489 | 100 | 92,893 | **0,9136** |
| **4** | 99,115 | 87,771 | 99,133 | 92,639 | 0,8972 |
| **5** | 95,575 | 86,413 | 98,266 | 95,431 | 0,9034 |
| **6** | 98,584 | 87,771 | 98,960 | 92,132 | 0,8947 |
| **7** | 96,283 | 87,771 | 98,613 | 95,177 | **0,9099** |
| **8** | 96,637 | 91,032 | 94,627 | 88,324 | 0,8945 |
| **9** | 94,336 | 89,402 | 97,400 | 95,177 | **0,9190** |
| **10** | 97,345 | 88,858 | 98,613 | 92,893 | 0,9046 |

**Classe 02 vs Classe 14**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 99,292 | 91,304 | 99,647 | 51,831 | 0,7680 |
| **2** | 97,699 | 75,543 | 98,591 | 90,704 | 0,8189 |
| **3** | 98,230 | 80,434 | 99,119 | 89,859 | 0,8457 |
| **4** | 93,805 | 84,239 | 98,415 | 94,084 | **0,8870** |
| **5** | 93,805 | 78,804 | 97,711 | 92,112 | 0,8455 |
| **6** | 93,451 | 83,152 | 97,183 | 92,112 | 0,8718 |
| **7** | 93,982 | 82,608 | 94,366 | 88,450 | 0,8527 |
| **8** | 94,159 | 85,869 | 99,295 | 92,676 | **0,8901** |
| **9** | 93,097 | 86,141 | 99,119 | 93,521 | **0,8955** |
| **10** | 93,982 | 87,228 | 88,556 | 78,309 | 0,8381 |

**Classe 02 vs Classe 18**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 100 | 94,837 | 99,816 | 26,415 | 0,7051 |
| **2** | 100 | 91,304 | 99,633 | 84,366 | **0,8819** |
| **3** | 34,690 | 35,597 | 98,717 | 94,609 | 0,5048 |
| **4** | 95,752 | 79,891 | 98,717 | 91,644 | 0,8485 |
| **5** | 96,991 | 83,423 | 97,435 | 92,183 | 0,8722 |
| **6** | 93,451 | 80,706 | 98,168 | 93,531 | 0,8621 |
| **7** | 98,761 | 85,869 | 98,351 | 91,105 | 0,8815 |
| **8** | 96,460 | 85,597 | 97,985 | 92,991 | **0,8886** |
| **9** | 97,345 | 88,858 | 98,351 | 90,566 | **0,8959** |
| **10** | 96,991 | 86,684 | 96,153 | 90,027 | 0,8812 |

**Tabela E.3 Classe 02, representação dos documentos utilizando a combinação NTC.**

**Classe 02 vs Classe 01**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 90,816 | 80,232 | 98,557 | 97,407 | **0,8771** |
| **2** | 94,132 | 84,883 | 96,874 | 92,962 | **0,8831** |
| **3** | 88,010 | 77,131 | 98,798 | 97,777 | 0,8596 |
| **4** | 93,112 | 83,720 | 96,394 | 93,333 | **0,8780** |
| **5** | 82,653 | 72,868 | 99,278 | 99,259 | 0,8393 |
| **6** | 78,571 | 70,930 | 99,519 | 99,259 | 0,8262 |
| **7** | 73,214 | 65,503 | 99,759 | 99,629 | 0,7897 |
| **8** | 72,193 | 66,279 | 99,759 | 99,629 | 0,7953 |
| **9** | 80,612 | 74,418 | 99,519 | 99,259 | 0,8496 |
| **10** | 67,091 | 64,728 | 99,759 | 99,629 | 0,7840 |

**Classe 02 vs Classe 07**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 88,520 | 78,682 | 96,692 | 87,790 | **0,8423** |
| **2** | 93,367 | 84,496 | 86,514 | 77,325 | **0,8466** |
| **3** | 90,816 | 77,90 | 94,656 | 85,465 | **0,8306** |
| **4** | 81,377 | 70,542 | 96,692 | 91,279 | 0,8000 |
| **5** | 76,020 | 67,829 | 96,946 | 91,279 | 0,7813 |
| **6** | 75,255 | 66,279 | 97,964 | 92,441 | 0,7738 |
| **7** | 90,306 | 80,232 | 90,839 | 79,651 | 0,8280 |
| **8** | 65,561 | 58,527 | 97,964 | 93,604 | 0,7190 |
| **9** | 63,010 | 57,364 | 98,218 | 94,186 | 0,7115 |
| **10** | 68,367 | 64,341 | 97,455 | 93,023 | 0,7615 |

**Classe 02 vs Classe 10**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 92,346 | 81,395 | 98,695 | 97,352 | **0,8805** |
| **2** | 91,071 | 79,069 | 98,695 | 97,941 | 0,8699 |
| **3** | 87,755 | 76,356 | 99,565 | 99,411 | 0,8621 |
| **4** | 98,724 | 95,736 | 61,956 | 55 | 0,7508 |
| **5** | 88,775 | 78,294 | 98,695 | 99,411 | 0,8745 |
| **6** | 93,367 | 84,496 | 95,652 | 93,529 | **0,8755** |
| **7** | 88,010 | 77,519 | 98,478 | 99,117 | 0,8677 |
| **8** | 91,071 | 80,620 | 97,391 | 98,235 | **0,8814** |
| **9** | 86,224 | 77,907 | 99,130 | 99,117 | 0,8701 |
| **10** | 62,755 | 58,914 | 99,782 | 100 | 0,7415 |

**Classe 02 vs Classe 14**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 86,224 | 78,294 | 96,457 | 94,444 | 0,8559 |
| **2** | 88,775 | 76,744 | 98,092 | 96,759 | **0,8553** |
| **3** | 86,989 | 75,581 | 98,365 | 98,611 | **0,8553** |
| **4** | 81,632 | 70,930 | 98,910 | 98,611 | 0,8243 |
| **5** | 93,367 | 85,658 | 85,831 | 86,111 | 0,8684 |
| **6** | 83,928 | 77,131 | 93,188 | 95,833 | 0,8541 |
| **7** | 91,581 | 81,782 | 89,645 | 92,592 | **0,8701** |
| **8** | 88,010 | 75,193 | 96,730 | 97,222 | 0,8472 |
| **9** | 84,183 | 72,868 | 98,092 | 97,685 | 0,8337 |
| **10** | 62,5 | 57,751 | 100 | 100 | 0,7322 |

**Classe 02 vs Classe 18**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 88,520 | 79,457 | 96,153 | 95,899 | **0,8613** |
| **2** | 87,244 | 75,96 | 98,076 | 99,684 | **0,8615** |
| **3** | 86,734 | 73,643 | 98,076 | 99,684 | 0,8463 |
| **4** | 77,806 | 72,480 | 98,290 | 99,053 | 0,8348 |
| **5** | 88,265 | 77,519 | 97,435 | 99,684 | **0,8715** |
| **6** | 75 | 66,666 | 98,931 | 100 | 0,8000 |
| **7** | 72,704 | 64,341 | 99,359 | 100 | 0,7830 |
| **8** | 72,704 | 65,503 | 99,145 | 100 | 0,7916 |
| **9** | 63,010 | 57,751 | 99,359 | 100 | 0,7322 |
| **10** | 58,418 | 54,263 | 99,572 | 100 | 0,7035 |

**Tabela E.4 Classe 02, representação dos documentos utilizando a combinação NNN.**

**Classe 02 vs Classe 01**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 97,959 | 48,449 | 98,557 | 91,851 | 0,6173 |
| **2** | 98,214 | 66,279 | 98,557 | 89,259 | 0,7467 |
| **3** | 95,918 | 70,930 | 98,557 | 92,222 | 0,7922 |
| **4** | 95,408 | 81,782 | 93,269 | 75,555 | 0,7888 |
| **5** | 95,918 | 75,581 | 97,836 | 91,111 | 0,8176 |
| **6** | 93,367 | 75,969 | 99,278 | 91,851 | 0,8235 |
| **7** | 90,816 | 79,457 | 92,307 | 83,333 | 0,8071 |
| **8** | 94,89 | 81,782 | 95,913 | 85,555 | **0,8307** |
| **9** | 94,387 | 82,945 | 94,951 | 90,740 | **0,8612** |
| **10** | 94,642 | 82,558 | 95,673 | 92,222 | **0,8659** |

**Classe 02 vs Classe 07**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 96,683 | 91,085 | 94,402 | 47,093 | 0,8048 |
| **2** | 98,214 | 83,720 | 87,022 | 68,604 | 0,8182 |
| **3** | 91,581 | 65,116 | 92,875 | 80,232 | 0,7304 |
| **4** | 94,642 | 79,845 | 94,147 | 83,720 | **0,8374** |
| **5** | 92,602 | 77,907 | 95,928 | 87,790 | **0,8375** |
| **6** | 96,428 | 80,620 | 90,076 | 78,488 | **0,8270** |
| **7** | 91,326 | 74,031 | 94,910 | 85,465 | 0,8059 |
| **8** | 88,775 | 74,806 | 90,076 | 80,814 | 0,7975 |
| **9** | 91,836 | 77,519 | 80,152 | 62,790 | 0,7663 |
| **10** | 96,173 | 81,395 | 85,496 | 65,697 | 0,7970 |

**Classe 02 vs Classe 10**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 95,918 | 45,348 | 99,347 | 96,764 | 0,6062 |
| **2** | 96,173 | 81,782 | 93,043 | 74,411 | 0,7590 |
| **3** | 95,663 | 70,542 | 97,826 | 91,176 | 0,7745 |
| **4** | 95,408 | 82,945 | 93,913 | 82,941 | 0,8075 |
| **5** | 95,918 | 80,232 | 96,739 | 88,823 | 0,8231 |
| **6** | 95,408 | 83,720 | 97,173 | 90 | **0,8504** |
| **7** | 82,908 | 71,317 | 99,347 | 99,117 | 0,8270 |
| **8** | 94,642 | 75,193 | 98,695 | 95,294 | **0,8291** |
| **9** | 96,173 | 79,069 | 96,956 | 92,941 | **0,8395** |
| **10** | 85,459 | 74,031 | 97,608 | 95,882 | 0,8251 |

**Classe 02 vs Classe 14**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 96,428 | 90,310 | 92,915 | 38,888 | 0,7480 |
| **2** | 67,091 | 46,124 | 98,910 | 87,037 | 0,5877 |
| **3** | 92,346 | 68,604 | 86,103 | 71,296 | 0,7123 |
| **4** | 80,867 | 62,015 | 90,190 | 79,629 | 0,6926 |
| **5** | 88,520 | 76,744 | 94,550 | 82,870 | **0,8032** |
| **6** | 97,704 | 92,635 | 56,403 | 52,777 | 0,7980 |
| **7** | 87,500 | 70,542 | 96,185 | 93,518 | 0,8018 |
| **8** | 86,734 | 77,519 | 82,561 | 74,074 | 0,7782 |
| **9** | 90,051 | 74,806 | 94,822 | 95,833 | **0,8391** |
| **10** | 93,367 | 80,232 | 94,550 | 86,111 | **0,8364** |

**Classe 02 vs Classe 18**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 98,214 | 92,248 | 87,606 | 24,921 | 0,6485 |
| **2** | 97,959 | 65,503 | 98,076 | 93,059 | 0,7528 |
| **3** | 95,408 | 67,441 | 97,008 | 94,952 | 0,7768 |
| **4** | 93,367 | 72,093 | 97,008 | 97,160 | 0,8212 |
| **5** | 92,091 | 79,457 | 94,444 | 92,113 | 0,8402 |
| **6** | 94,898 | 81,395 | 94,444 | 90,220 | **0,8417** |
| **7** | 83,163 | 74,806 | 93,589 | 93,690 | 0,8195 |
| **8** | 71,938 | 64,728 | 97,649 | 95,899 | 0,7626 |
| **9** | 94,642 | 81,007 | 97,222 | 95,899 | **0,8708** |
| **10** | 97,193 | 84,883 | 95,512 | 90,851 | **0,8656** |

**Tabela E.5 Classe 18, representação dos documentos utilizando a combinação ATN.**

**Classe 18 vs Classe 01**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 100 | 96 | 100 | 22,082 | 0,7332 |
| **2** | 100 | 86,666 | 100 | 71,293 | **0,8217** |
| **3** | 99,820 | 80,8 | 100 | 87,697 | **0,8452** |
| **4** | 99,640 | 76 | 100 | 88,012 | **0,8166** |
| **5** | 99,640 | 73,333 | 100 | 89,274 | 0,8041 |
| **6** | 99,640 | 73,066 | 100 | 88,959 | 0,8012 |
| **7** | 99,461 | 72,266 | 100 | 89,589 | 0,7982 |
| **8** | 98,743 | 70,933 | 100 | 90,851 | 0,7940 |
| **9** | 98,025 | 72,266 | 100 | 92,429 | 0,8090 |
| **10** | 98,025 | 70,133 | 100 | 90,536 | 0,7874 |

**Classe 18 vs Classe 02**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 100 | 95,733 | 100 | 57,922 | 0,8013 |
| **2** | 99,820 | 91,2 | 100 | 93,766 | **0,9231** |
| **3** | 99,461 | 88,266 | 100 | 97,402 | **0,9246** |
| **4** | 99,461 | 87,466 | 100 | 97,402 | **0,9201** |
| **5** | 99,102 | 86,933 | 100 | 96,883 | 0,9144 |
| **6** | 99,102 | 86,933 | 100 | 96,623 | 0,9132 |
| **7** | 99,102 | 86,666 | 100 | 98,181 | 0,9194 |
| **8** | 98,204 | 86,933 | 100 | 96,883 | 0,9144 |
| **9** | 98,025 | 86,666 | 100 | 97,922 | 0,9181 |
| **10** | 98,384 | 87,466 | 99,652 | 94,805 | 0,9073 |

**Classe 18 vs Classe 7**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 68,402 | 2,933 | 69,244 | 99,208 | 0,0566 |
| **2** | 67,504 | 88,8 | 68,900 | 81,002 | **0,8538** |
| **3** | 76,122 | 91,466 | 61,512 | 80,738 | **0,8673** |
| **4** | 86,355 | 85,066 | 48,453 | 84,168 | **0,8462** |
| **5** | 73,608 | 76 | 60,137 | 87,335 | 0,8051 |
| **6** | 72,172 | 74,4 | 62,543 | 87,071 | 0,7937 |
| **7** | 54,578 | 52 | 74,570 | 92,084 | 0,6500 |
| **8** | 63,913 | 63,466 | 65,979 | 89,445 | 0,7289 |
| **9** | 58,168 | 61,066 | 72,508 | 89,182 | 0,7101 |
| **10** | 69,120 | 75,2 | 64,604 | 88,918 | 0,8069 |

**Classe 18 vs Classe 10**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 100 | 94,933 | 100 | 35,606 | 0,7221 |
| **2** | 100 | 87,733 | 100 | 86,616 | **0,8692** |
| **3** | 99,820 | 84,266 | 100 | 95,202 | **0,8901** |
| **4** | 99,820 | 83,2 | 100 | 94,444 | 0,8801 |
| **5** | 99,640 | 83,466 | 100 | 94,949 | **0,8842** |
| **6** | 99,461 | 82,666 | 100 | 93,939 | 0,8745 |
| **7** | 98,922 | 82,133 | 100 | 95,959 | 0,8813 |
| **8** | 98,384 | 82,133 | 100 | 95,707 | 0,8800 |
| **9** | 98,563 | 83,2 | 100 | 94,949 | 0,8826 |
| **10** | 98,025 | 80,533 | 100 | 96,212 | 0,8728 |

**Classe 18 vs Classe 14**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 100 | 94,666 | 100 | 50,773 | 0,7709 |
| **2** | 100 | 89,866 | 100 | 89,433 | 0,8951 |
| **3** | 99,820 | 86,933 | 100 | 93,556 | 0,8981 |
| **4** | 99,461 | 85,333 | 100 | 94,587 | 0,8939 |
| **5** | 99,461 | 85,333 | 100 | 94,072 | 0,8914 |
| **6** | 98,563 | 86,133 | 100 | 95,618 | **0,9035** |
| **7** | 98,384 | 84,8 | 100 | 96,907 | **0,9021** |
| **8** | 98,204 | 83,733 | 100 | 97,422 | 0,8984 |
| **9** | 97,845 | 84,533 | 100 | 96,649 | **0,8993** |
| **10** | 97,127 | 83,466 | 100 | 96,649 | 0,8930 |

**Tabela E.6 Classe 18, representação dos documentos utilizando a combinação NTC.**

**Classe 18 vs Classe 01**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 59,960 | 51,032 | 99,521 | 99,253 | 0,6732 |
| **2** | 89,062 | 69,026 | 94,258 | 90,298 | 0,7813 |
| **3** | 84,570 | 67,846 | 95,215 | 95,895 | **0,7931** |
| **4** | 89,453 | 70,796 | 90,669 | 89,179 | 0,7895 |
| **5** | 90,234 | 69,616 | 90,909 | 87,313 | 0,7750 |
| **6** | 75,976 | 59,587 | 98,803 | 98,134 | 0,7399 |
| **7** | 85,546 | 68,141 | 94,019 | 94,776 | **0,7911** |
| **8** | 83,789 | 67,256 | 94,736 | 96,268 | **0,7903** |
| **9** | 65,429 | 55,162 | 99,760 | 98,507 | 0,7057 |
| **10** | 79,492 | 64,601 | 96,890 | 97,388 | 0,7752 |

**Classe 18 vs Classe 02**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 94,531 | 92,625 | 90,411 | 79,487 | 0,8959 |
| **2** | 85,351 | 84,365 | 99,178 | 98,290 | **0,9094** |
| **3** | 89,843 | 88,790 | 96,438 | 94,444 | **0,9219** |
| **4** | 91,015 | 89,970 | 96,986 | 94,871 | **0,9299** |
| **5** | 76,367 | 75,811 | 100 | 99,145 | 0,8595 |
| **6** | 73,242 | 73,746 | 100 | 99,145 | 0,8460 |
| **7** | 88,281 | 89,085 | 98,356 | 97,435 | 0,9335 |
| **8** | 79,882 | 78,171 | 100 | 99,145 | 0,8746 |
| **9** | 82,226 | 82,005 | 100 | 98,717 | 0,8968 |
| **10** | 71,875 | 73,746 | 100 | 99,145 | 0,8460 |

**Classe 18 vs Classe 07**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 67,773 | 73,746 | 55,752 | 75,925 | 0,7800 |
| **2** | 45,117 | 54,277 | 79,646 | 92,129 | 0,6815 |
| **3** | 71,289 | 79,646 | 54,424 | 81,944 | **0,8333** |
| **4** | 44,531 | 48,967 | 76,769 | 87,963 | 0,6252 |
| **5** | 52,929 | 52,507 | 71,017 | 92,592 | 0,6679 |
| **6** | 79,687 | 84,365 | 46,238 | 77,314 | **0,8487** |
| **7** | 44,140 | 48,967 | 76,991 | 91,203 | 0,6336 |
| **8** | 45,117 | 53,392 | 75,000 | 92,129 | 0,6741 |
| **9** | 58,593 | 63,716 | 63,495 | 87,037 | 0,7410 |
| **10** | 82,226 | 88,495 | 43,805 | 75 | **0,8658** |

**Classe 18 vs Classe 10**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 88,867 | 82,890 | 97,624 | 90,544 | 0,8606 |
| **2** | 84,375 | 78,761 | 99,568 | 97,707 | 0,8697 |
| **3** | 88,867 | 81,710 | 97,624 | 95,702 | **0,8780** |
| **4** | 75,976 | 73,746 | 99,784 | 99,713 | 0,8475 |
| **5** | 80,859 | 76,106 | 99,352 | 99,140 | 0,8600 |
| **6** | 69,140 | 66,076 | 100 | 100 | 0,7957 |
| **7** | 86,328 | 82,890 | 96,328 | 93,982 | **0,8768** |
| **8** | 77,929 | 74,926 | 99,136 | 98,853 | 0,8509 |
| **9** | 86,914 | 83,185 | 95,896 | 93,982 | **0,8785** |
| **10** | 77,343 | 76,106 | 99,136 | 99,426 | 0,8614 |

**Classe 18 vs Classe 14**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 86,718 | 81,120 | 99,054 | 97,709 | **0,8871** |
| **2** | 83,593 | 78,466 | 99,763 | 99,236 | 0,8764 |
| **3** | 79,101 | 75,811 | 100 | 100 | 0,8624 |
| **4** | 76,171 | 72,861 | 100 | 100 | 0,8430 |
| **5** | 90,820 | 87,905 | 92,435 | 89,313 | **0,8962** |
| **6** | 92,187 | 88,790 | 86,052 | 83,969 | 0,8827 |
| **7** | 83,789 | 80,826 | 98,8180 | 99,236 | **0,8911** |
| **8** | 69,140 | 67,256 | 100 | 100 | 0,8042 |
| **9** | 80,273 | 78,171 | 99,290 | 99,618 | 0,8760 |
| **10** | 80,273 | 78,466 | 99,290 | 99,618 | 0,8779 |

**Tabela E.7 Classe 18, representação dos documentos utilizando a combinação NNN.**

**Classe 18 vs Classe 01**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 60,937 | 4,719 | 96,89 | 96,268 | 0,0877 |
| **2** | 98,632 | 76,991 | 100 | 56,343 | 0,7280 |
| **3** | 98,437 | 71,681 | 100 | 71,268 | 0,7375 |
| **4** | 98,046 | 70,206 | 100 | 73,134 | 0,7334 |
| **5** | 98,242 | 67,256 | 97,368 | 73,134 | 0,7136 |
| **6** | 96,289 | 65,781 | 98,086 | 82,835 | **0,7336** |
| **7** | 95,898 | 69,616 | 96,411 | 82,835 | **0,7601** |
| **8** | 90,039 | 63,126 | 92,823 | 87,686 | 0,7304 |
| **9** | 93,554 | 65,781 | 96,172 | 81,716 | 0,7300 |
| **10** | 93,75 | 65,781 | 96,650 | 82,835 | **0,7336** |

**Classe 18 vs Classe 02**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | ND | | | | |
| **2** | 94,921 | 84,070 | 97,260 | 79,914 | 0,8495 |
| **3** | 97,265 | 82,890 | 99,726 | 86,752 | 0,8633 |
| **4** | 96,679 | 84,365 | 99,452 | 85,470 | 0,8680 |
| **5** | 96,484 | 87,905 | 99,452 | 87,179 | 0,8936 |
| **6** | 95,898 | 87,315 | 99,726 | 87,179 | 0,8902 |
| **7** | 95,507 | 87,610 | 99,452 | 88,461 | **0,8959** |
| **8** | 96,093 | 90,265 | 91,780 | 82,906 | 0,8934 |
| **9** | ND | | | | |
| **10** | ND | | | | |

Alguns treinamentos da classe 18 contra a classe 02 não convergiram.

**Classe 18 vs Classe 07**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 69,335 | 96,755 | 57,964 | 30,092 | 0,8020 |
| **2** | 83,593 | 82,890 | 41,814 | 70,370 | **0,8216** |
| **3** | 62,890 | 86,135 | 64,380 | 60,648 | **0,8156** |
| **4** | 73,437 | 80,531 | 51,548 | 76,851 | **0,8248** |
| **5** | 54,687 | 68,731 | 73,230 | 80,092 | 0,7577 |
| **6** | 70,507 | 78,466 | 55,531 | 72,685 | 0,8012 |
| **7** | 52,929 | 60,177 | 66,150 | 83,333 | 0,7047 |
| **8** | 36,523 | 49,852 | 84,070 | 84,722 | 0,6248 |
| **9** | 50,781 | 55,162 | 70,354 | 86,111 | 0,6727 |
| **10** | 35,742 | 37,463 | 80,531 | 90,277 | 0,5216 |

**Classe 18 vs Classe 10**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 97,265 | 87,905 | 77,753 | 47,277 | 0,7259 |
| **2** | 97,070 | 79,941 | 100 | 71,346 | 0,7634 |
| **3** | 97,070 | 76,991 | 99,784 | 80,802 | 0,7826 |
| **4** | 96,289 | 77,286 | 100 | 85,100 | 0,8025 |
| **5** | 94,921 | 77,876 | 95,464 | 82,808 | 0,7964 |
| **6** | 95,312 | 79,646 | 99,784 | 85,386 | 0,8182 |
| **7** | 92,773 | 75,811 | 99,784 | 91,977 | **0,8237** |
| **8** | 93,554 | 78,466 | 97,624 | 87,679 | 0,8210 |
| **9** | 95,117 | 83,480 | 96,328 | 81,661 | **0,8251** |
| **10** | 93,359 | 82,890 | 98,488 | 87,679 | **0,8477** |

**Classe 18 vs Classe 14**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 98,046 | 91,150 | 98,345 | 44,274 | 0,7783 |
| **2** | 98,046 | 84,365 | 94,799 | 71,374 | 0,8171 |
| **3** | 98,242 | 79,941 | 99,290 | 81,679 | 0,8237 |
| **4** | 96,875 | 81,710 | 99,763 | 83,206 | 0,8394 |
| **5** | 96,289 | 82,005 | 97,872 | 82,824 | 0,8399 |
| **6** | 95,312 | 82,300 | 99,290 | 89,694 | 0,8651 |
| **7** | 94,921 | 84,660 | 96,690 | 86,641 | 0,8684 |
| **8** | 97,656 | 90,265 | 82,269 | 80,152 | **0,8780** |
| **9** | 93,164 | 84,365 | 98,581 | 93,893 | **0,8924** |
| **10** | 93,164 | 82,595 | 96,926 | 95,038 | **0,8861** |

**Tabela E.8 Classe 10, representação dos documentos utilizando a combinação ATN.**

**Classe 10 vs Classe 01**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 100 | 98,740 | 100 | 6,624 | 0,7226 |
| **2** | 100 | 94,458 | 100 | 55,836 | 0,8224 |
| **3** | 100 | 87,909 | 100 | 78,233 | 0,8564 |
| **4** | 100 | 84,886 | 100 | 84,542 | **0,8608** |
| **5** | 100 | 84,382 | 100 | 85,173 | **0,8601** |
| **6** | 100 | 82,619 | 100 | 87,066 | **0,8564** |
| **7** | 99,829 | 82,619 | 100 | 85,804 | 0,8519 |
| **8** | 99,829 | 83,375 | 100 | 85,489 | 0,8553 |
| **9** | 99,829 | 84,886 | 100 | 82,965 | 0,8553 |
| **10** | 99,659 | 83,123 | 99,791 | 85,489 | 0,8538 |

**Classe 10 vs Classe 02**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 100 | 96,725 | 100 | 29,123 | 0,7273 |
| **2** | 100 | 94,710 | 100 | 78,350 | 0,8775 |
| **3** | 100 | 91,687 | 100 | 92,268 | **0,9204** |
| **4** | 100 | 89,672 | 100 | 93,041 | 0,9128 |
| **5** | 100 | 88,665 | 100 | 92,525 | 0,9049 |
| **6** | 99,829 | 88,413 | 100 | 93,814 | 0,9093 |
| **7** | 99,319 | 87,657 | 100 | 93,299 | 0,9027 |
| **8** | 98,809 | 87,153 | 100 | 96,391 | 0,9141 |
| **9** | 98,639 | 89,420 | 100 | 96,649 | **0,9281** |
| **10** | 98,129 | 88,161 | 100 | 96,649 | **0,9211** |

**Classe 10 vs Classe 07**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 100 | 96,977 | 100 | 37,301 | 0,7556 |
| **2** | 100 | 92,191 | 100 | 83,597 | **0,8873** |
| **3** | 100 | 89,168 | 100 | 92,063 | **0,9065** |
| **4** | 100 | 84,886 | 100 | 91,798 | **0,8810** |
| **5** | 99,829 | 83,375 | 100 | 92,592 | 0,8757 |
| **6** | 99,829 | 80,100 | 100 | 94,709 | 0,8653 |
| **7** | 99,659 | 82,115 | 100 | 94,179 | 0,8752 |
| **8** | 99,659 | 82,115 | 100 | 94,444 | 0,8763 |
| **9** | 99,659 | 81,864 | 100 | 91,798 | 0,8632 |
| **10** | 97,959 | 76,826 | 99,484 | 94,444 | 0,8437 |

**Classe 10 vs Classe 14**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 100 | 96,473 | 100 | 32,216 | 0,7344 |
| **2** | 100 | 92,947 | 100 | 79,123 | 0,8713 |
| **3** | 100 | 87,405 | 100 | 91,237 | **0,8920** |
| **4** | 100 | 84,886 | 100 | 90,721 | 0,8753 |
| **5** | 100 | 83,375 | 100 | 90,206 | 0,8642 |
| **6** | 100 | 84,634 | 100 | 90,721 | 0,8739 |
| **7** | 99,659 | 84,382 | 100 | 90,463 | 0,8713 |
| **8** | 98,979 | 83,123 | 100 | 92,783 | 0,8742 |
| **9** | 98,469 | 84,886 | 100 | 93,299 | **0,8868** |
| **10** | 97,789 | 83,879 | 100 | 94,845 | **0,8880** |

**Classe 10 vs Classe 18**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 100 | 97,984 | 100 | 10,133 | 0,6928 |
| **2** | 100 | 91,435 | 100 | 77,600 | 0,8602 |
| **3** | 100 | 88,916 | 100 | 93,333 | **0,9110** |
| **4** | 100 | 84,886 | 100 | 93,333 | **0,8880** |
| **5** | 100 | 83,123 | 100 | 93,333 | 0,8777 |
| **6** | 100 | 82,619 | 100 | 93,333 | 0,8747 |
| **7** | 99,829 | 83,123 | 100 | 94,666 | 0,8835 |
| **8** | 99,659 | 83,123 | 100 | 93,866 | 0,8800 |
| **9** | 98,809 | 81,612 | 100 | 95,733 | 0,8792 |
| **10** | 98,809 | 82,367 | 100 | 96,266 | **0,8862** |

**Tabela E.9 Classe 10, representação dos documentos utilizando a combinação NTC.**

**Classe 10 vs Classe 01**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 87,767 | 84,109 | 88,111 | 82,014 | **0,8504** |
| **2** | 92,038 | 90,411 | 81,585 | 74,460 | **0,8616** |
| **3** | 79,805 | 73,150 | 93,706 | 89,208 | 0,8066 |
| **4** | 86,601 | 79,452 | 89,510 | 83,812 | 0,8286 |
| **5** | 82,524 | 73,424 | 93,473 | 90,647 | 0,8134 |
| **6** | 97,087 | 95,342 | 62,937 | 51,798 | 0,8217 |
| **7** | 92,621 | 90,137 | 76,923 | 70,863 | **0,8490** |
| **8** | 96,893 | 95,068 | 63,170 | 51,438 | 0,8194 |
| **9** | 87,184 | 83,835 | 86,946 | 80,215 | 0,8430 |
| **10** | 98,252 | 96,438 | 53,613 | 44,244 | 0,8073 |

**Classe 10 vs Classe 02**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 89,320 | 81,095 | 97,022 | 93,023 | 0,8719 |
| **2** | 87,184 | 77,534 | 96,526 | 94,186 | 0,8537 |
| **3** | 86,019 | 80,821 | 94,540 | 89,534 | 0,8588 |
| **4** | 91,068 | 90,958 | 91,315 | 87,596 | **0,9108** |
| **5** | 80,582 | 72,328 | 97,270 | 96,124 | 0,8263 |
| **6** | 93,786 | 94,794 | 84,367 | 83,720 | **0,9190** |
| **7** | 95,728 | 96,712 | 78,411 | 78,294 | **0,9121** |
| **8** | 96,310 | 96,712 | 76,674 | 74,806 | 0,9017 |
| **9** | 91,456 | 89,315 | 91,563 | 87,209 | 0,9006 |
| **10** | 98,835 | 98,904 | 41,439 | 41,860 | 0,8242 |

**Classe 10 vs Classe 7**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 84,660 | 73,150 | 91,428 | 82,352 | 0,7958 |
| **2** | 81,941 | 73,698 | 93,406 | 90,950 | **0,8226** |
| **3** | 74,757 | 63,561 | 96,923 | 91,855 | 0,7545 |
| **4** | 40,388 | 33,698 | 99,120 | 96,832 | 0,4970 |
| **5** | 87,961 | 83,561 | 78,241 | 76,470 | **0,8449** |
| **6** | 78,640 | 69,041 | 95,384 | 90,950 | 0,7912 |
| **7** | 60,776 | 48,767 | 98,461 | 94,570 | 0,6414 |
| **8** | 62,135 | 53,972 | 98,022 | 93,665 | 0,6840 |
| **9** | 80,194 | 71,780 | 92,967 | 88,235 | **0,8025** |
| **10** | 68,737 | 59,726 | 96,703 | 90,497 | 0,7219 |

**Classe 10 vs Classe 14**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 85,242 | 73,698 | 96,789 | 89,781 | 0,8127 |
| **2** | 86,796 | 79,452 | 90,367 | 83,211 | 0,8274 |
| **3** | 86,019 | 81,917 | 87,614 | 84,306 | **0,8458** |
| **4** | 72,038 | 63,835 | 98,623 | 95,985 | 0,7652 |
| **5** | 88,349 | 85,753 | 82,110 | 76,642 | **0,8437** |
| **6** | 80,388 | 75,890 | 95,412 | 90,510 | **0,8293** |
| **7** | 69,126 | 62,739 | 98,394 | 95,985 | 0,7570 |
| **8** | 83,301 | 75,890 | 95,642 | 90,510 | 0,8293 |
| **9** | 60,970 | 55,616 | 99,082 | 97,810 | 0,7073 |
| **10** | 76,310 | 67,397 | 97,706 | 93,795 | 0,7834 |

**Classe 10 vs Classe 18**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 85,242 | 74,794 | 96,734 | 93,498 | 0,8285 |
| **2** | 84,271 | 74,246 | 96,122 | 93,188 | 0,8237 |
| **3** | 75,145 | 62,191 | 98,571 | 97,832 | 0,7579 |
| **4** | 80,000 | 70,137 | 95,918 | 95,046 | 0,8038 |
| **5** | 85,242 | 80,547 | 91,020 | 90,712 | **0,8534** |
| **6** | 60,000 | 51,232 | 98,979 | 99,380 | 0,6751 |
| **7** | 84,466 | 79,726 | 90,816 | 92,260 | **0,8546** |
| **8** | 59,029 | 50,137 | 98,979 | 99,380 | 0,6655 |
| **9** | 86,601 | 84,931 | 86,938 | 89,783 | **0,8757** |
| **10** | 48,155 | 43,013 | 99,387 | 99,380 | 0,5992 |

**Tabela E.10 Classe 10, representação dos documentos utilizando a combinação NNN.**

**Classe 10 vs Classe 01**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 98,640 | 96,986 | 85,314 | 8,992 | 0,7284 |
| **2** | 96,699 | 83,561 | 95,804 | 61,151 | 0,7841 |
| **3** | 89,902 | 55,342 | 98,135 | 89,208 | 0,6767 |
| **4** | 97,281 | 84,383 | 94,871 | 68,705 | 0,8105 |
| **5** | 98,640 | 81,643 | 88,111 | 75,179 | **0,8142** |
| **6** | 93,786 | 83,835 | 91,142 | 75,539 | **0,8281** |
| **7** | 92,621 | 78,904 | 95,338 | 78,417 | 0,8079 |
| **8** | 92,038 | 77,808 | 92,307 | 80,215 | 0,8068 |
| **9** | 92,427 | 84,383 | 86,713 | 73,741 | **0,8257** |
| **10** | 84,466 | 70,411 | 95,571 | 87,769 | 0,7835 |

**Classe 10 vs Classe 02**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 55,145 | 10,137 | 91,563 | 88,372 | 0,1713 |
| **2** | 78,058 | 59,178 | 93,796 | 83,720 | 0,6934 |
| **3** | 96,699 | 83,287 | 97,518 | 74,806 | 0,8283 |
| **4** | 89,708 | 63,835 | 97,766 | 87,209 | 0,7385 |
| **5** | 92,815 | 85,205 | 95,781 | 79,845 | **0,8544** |
| **6** | 95,922 | 86,301 | 89,578 | 76,744 | 0,8514 |
| **7** | 94,174 | 83,561 | 95,285 | 81,007 | 0,8484 |
| **8** | 87,961 | 83,835 | 90,818 | 82,170 | **0,8536** |
| **9** | 88,543 | 79,452 | 95,781 | 86,434 | 0,8406 |
| **10** | 91,068 | 86,301 | 90,818 | 79,845 | **0,8607** |

**Classe 10 vs Classe 07**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 97,087 | 88,767 | 96,044 | 48,416 | **0,8070** |
| **2** | 98,446 | 85,479 | 89,011 | 56,561 | **0,8072** |
| **3** | 94,174 | 60,000 | 93,186 | 75,113 | 0,6854 |
| **4** | 89,708 | 66,027 | 87,692 | 73,303 | 0,7248 |
| **5** | 94,174 | 70,411 | 91,868 | 73,755 | 0,7559 |
| **6** | 86,213 | 67,123 | 94,725 | 85,067 | 0,7621 |
| **7** | 91,068 | 69,863 | 95,604 | 84,162 | 0,7786 |
| **8** | 89,320 | 72,328 | 91,428 | 76,018 | 0,7742 |
| **9** | 94,563 | 84,657 | 77,802 | 63,800 | **0,8196** |
| **10** | 87,767 | 73,424 | 92,087 | 79,638 | 0,7906 |

**Classe 10 vs Classe 14**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 97,475 | 94,794 | 92,201 | 19,343 | 0,7425 |
| **2** | 92,233 | 52,054 | 95,642 | 76,277 | 0,6129 |
| **3** | 92,621 | 78,082 | 94,036 | 74,452 | 0,7917 |
| **4** | 97,087 | 84,931 | 80,733 | 62,043 | 0,7959 |
| **5** | 96,699 | 81,095 | 90,137 | 75,182 | **0,8121** |
| **6** | 86,990 | 74,520 | 94,036 | 86,496 | 0,8071 |
| **7** | 94,951 | 81,095 | 85,779 | 77,007 | **0,8177** |
| **8** | 86,213 | 75,342 | 95,871 | 83,211 | 0,8017 |
| **9** | 85,048 | 74,520 | 98,394 | 85,766 | 0,8047 |
| **10** | 92,815 | 82,739 | 86,926 | 72,627 | **0,8140** |

**Classe 10 vs Classe 18**

| RBF | Treino (+) | Teste (+) | Treino (-) | Teste (-) | F1 |
|---|---|---|---|---|---|
| **1** | 95,922 | 12,876 | 96,530 | 90,402 | 0,212 |
| **2** | 91,650 | 49,589 | 93,265 | 78,018 | 0,586 |
| **3** | 95,728 | 66,301 | 97,755 | 84,520 | 0,736 |
| **4** | 97,087 | 81,643 | 89,591 | 72,136 | 0,791 |
| **5** | 90,873 | 75,890 | 88,571 | 79,876 | 0,783 |
| **6** | 93,592 | 78,082 | 90,408 | 80,185 | **0,798** |
| **7** | 92,233 | 72,876 | 91,224 | 82,043 | 0,772 |
| **8** | 90,873 | 77,260 | 92,857 | 83,591 | **0,805** |
| **9** | 83,301 | 66,849 | 93,877 | 89,783 | 0,760 |
| **10** | 92,427 | 81,369 | 89,387 | 86,068 | **0,840** |

# Apêndice F – Teste de Hipóteses utilizando a distribuição t de Student

O objetivo do teste estatístico de hipótese (BUSSAB, 1987) é fornecer ferramentas (regras) que nos permitam aceitar ou rejeitar uma hipótese com base em elementos amostrais.

Teremos sempre duas hipóteses, são elas:

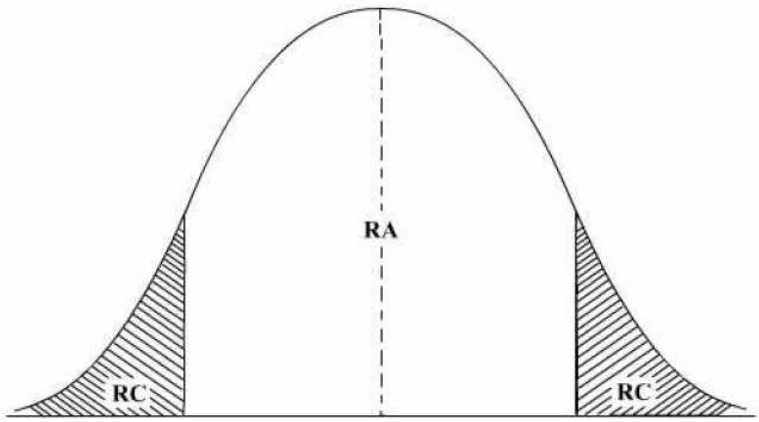
- Hipótese nula ($H_0$) é a hipótese que estamos colocando à prova.
- Hipótese alternativa ($H_1$) é a hipótese que será considerada como aceitável caso $H_0$ seja rejeitado.

Dois tipos de erros podem ser cometidos nos testes de hipótese, estes erros podem ser:

- Erro Tipo I ($\alpha$), é a situação em que se rejeita a hipótese nula quando esta é verdadeira. Geralmente a probabilidade de cometer este erro é representada por $\alpha$ (nível de significância do teste). Geralmente trabalhamos com este erro.
- Erro Tipo II ($\beta$), é a situação em que se aceita a hipótese nula quando esta é falsa.

Quando o valor observado da estatística pertence à região crítica (RC), rejeitamos $H_0$. Esta região é sempre construída sob a hipótese de $H_0$ ser verdadeira. O limite da região de aceitação (RA) e RC é normalmente tabelado.

Gráfico do teste bilateral:




Se $-Z_{tab} < Z_{calc} < Z_{tab}$, aceitamos $H_0$. Z é a estatística de teste.

A distribuição t de Student é uma das distribuições mais importantes para a inferência sobre médias populacionais em intervalos de confiança para uma população com as seguintes características:

- O desvio padrão da população é desconhecido
- A quantidade de amostras é inferior a 30

A distribuição t é similar à distribuição normal, ela apresenta as seguintes características:

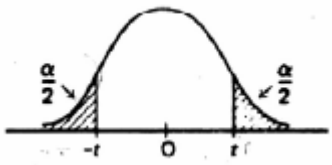
- Ela é simétrica em torno do zero
- Ela tem o formato de um sino
- Ela tem uma cauda mais comprida quando comparada à distribuição normal
- Ela tem o seu formato dependente de um parâmetro denominado 'graus de liberdade' (DF), quanto maior o valor do DF, menor a sua cauda, aproximando-se desta forma a uma distribuição normal

O teste de hipótese usando a distribuição t de Student fornece valores de t para médias populacionais em função da média amostral, do desvio padrão amostral, dos graus de liberdade da amostra e do intervalo de confiança desejado.

O grau de liberdade é normalmente representado por ν varia com a quantidade de amostras utilizadas nos experimentos. Para amostras menores que 30 elementos, o grau de liberdade é N-1, em que N é a quantidade de elementos medidos no experimento. O nível de confiança do intervalo de confiança é uma validação de quão confiável estamos de que a verdadeira média da população esteja dentro do intervalo.

Assim para um processo de decisão é feita uma comparação entre o t calculado ($t_{calc}$) e o t tabelado ($t_{tab}$). Se o $t_{calc}$ estiver entre $-t_{tab}$ e $+t_{tab}$ então aceitamos a hipótese $H_0$, caso contrário, rejeitamos.

TABELA DA DISTRIBUIÇÃO t DE STUDENT




| φ \ α | 0,50 | 0,25 | 0,10 | 0,05 | 0,025 | 0,01 | 0,005 |
|---|---|---|---|---|---|---|---|
| 1 | 1,00000 | 2,4142 | 6.3138 | 12.706, | 25,542 | 63,657 | 127,32 |
| 2 | 0,81650 | 1,6036 | 2,9200 | 4,3127 | 6,2053 | 9,9248 | 14,089 |
| 3 | 0,76489 | 1,4226 | 2,3534 | 3,1825 | 4,1765 | 5,8409 | 7,4533 |
| 4 | 0,74070 | 1,3444 | 2,1318 | 2,7764 | 3,4954 | 4,6041 | 5,5976 |
| 5 | 0,72669 | 1,3009 | 2,0150 | 2,5706 | 3,1634 | 4,0321 | 4,7733 |
| 6 | 0,71756 | 1,2733 | 1,9432 | 2,4469 | 2,9687 | 3,7074 | 4,3168 |
| 7 | 0,71114 | 1,2543 | 1,8946 | 2,3646 | 2,8412 | 3,4995 | 4,0293 |
| 8 | 0,70639 | 1,2403 | 1,8595 | 2,3060 | 2,7515 | 3,3554 | 3,8325 |
| 9 | 0,70272 | 1,2297 | 1,8331 | 2,2622 | 2,6850 | 3,2498 | 3,6897 |
| 10 | 0,69981 | 1,2213 | 1,8125 | 2,2281 | 2,6338 | 3,1693 | 3,5814 |
| 11 | 0,69745 | 1,2145 | 1,7959 | 2,2010 | 2,5931 | 3,1058 | 3,4966 |
| 12 | 0,69548 | 1,2089 | 1,7823 | 2,1788 | 2,5600 | 3,9545 | 3,4284 |
| 13 | 0,69384 | 1,2041 | 1,7709 | 2,1604 | 2,5326 | 3,0123 | 3,3725 |
| 14 | 0,692 | 1,2001 | 1,7613 | 2,1448 | 2,5096 | 2,9768 | 3,3257 |
| 15 | 0,69120 | 1,1967 | 1,7530 | 2,1315 | 2,4899 | 2,9467 | 3,2860 |
| 16 | 0,69013 | 1,1937 | 1,7459 | 2,1199 | 2,4729 | 2,9208 | 3,2520 |
| 17 | 0,68919 | 1,1910 | 1,7396 | 2,1098 | 2,4581 | 2,8982 | 3,2225 |
| 18 | 0,68837 | 1,1887 | 1,7341 | 2,1009 | 2,4450 | 2,8784 | 3,1966 |
| 19 | 0,68763 | 1,1866 | 1,7291 | 2,0930 | 2,4334 | 2,8609 | 3,1737 |
| 20 | 0,68696 | 1,1848 | 1,7247 | 2,0860 | 2,4231 | 2,8453 | 3,1534 |
| 21 | 0,68635 | 1,1831 | 1,7207 | 2,0796 | 2,4138 | 2,8314 | 3,1352 |
| 22 | 0,68580 | 1,1816 | 1,7171 | 2,0739 | 2,4055 | 2,8188 | 3,1188 |
| 23 | 0,68531 | 1,1802 | 1,7139 | 2,0687 | 2,3979 | 2,8073 | 3,1040 |
| 24 | 0,68485 | 1,1789 | 1,7109 | 2,0639 | 2,3910 | 2,7969 | 3,0905 |
| 25 | 0,68443 | 1,1777 | 1,7081 | 2,0595 | 2,3846 | 2,7874 | 3,0782 |
| 26 | 0,68405 | 1,1766 | 1,7056 | 2,0555 | 2,3788 | 2,7787 | 3,0669 |
| 27 | 0,68370 | 1,1757 | 1,7033 | 2,0518 | 2,3734 | 2,7707 | 3,0565 |
| 28 | 0,68335 | 1,1748 | 1,7011 | 2,0484 | 2,3685 | 2,7633 | 3,0469 |
| 29 | 0,68304 | 1,1739 | 1,6991 | 2,0452 | 2,3638 | 2,7564 | 3,0380 |
| 30 | 0,68276 | 1,1731 | 1,6973 | 2,0423 | 2,3596 | 2,7500 | 3,0298 |
| 40 | 0,68066 | 1,1673 | 1,6839 | 2,0211 | 2,3289 | 2,7045 | 2,9712 |
| 60 | 0,67862 | 1,1616 | 1,6707 | 2,0003 | 2,2991 | 2,6603 | 2,9146 |
| 120 | 0,67656 | 1,1559 | 1,6577 | 1,9799 | 2,2699 | 2,6174 | 2,8599 |
| ∞ | 0,67449 | 1,1503 | 1,6449 | 1,9600 | 2,2414 | 2,5758 | 2,8070 |

Extraída do livro “Curso de Estatística”-Jairo Simon da Fonseca & Gilberto de Andrade Martins-Editora Atlas