Uma Introdução às *Support Vector Machines*

Ana Carolina Lorena [1]
André C. P. L. F. de Carvalho [2]

Resumo: Neste artigo é apresentada uma introdução às Máquinas de Vetores de Suporte (SVMs, do Inglês *Support Vector Machines*), técnica de Aprendizado de Máquina que vem recebendo crescente atenção nos últimos anos. As SVMs vêm sendo utilizadas em diversas tarefas de reconhecimento de padrões, obtendo resultados superiores aos alcançados por outras técnicas de aprendizado em várias aplicações.
Palavras-chave: Aprendizado de Máquina, Classificação, Máquinas de Vetores de Suporte (*Support Vector Machines*)

Abstract: This paper presents an introduction to the Support Vector Machines (SVMs), a Machine Learning technique that has received increasing attention in the last years. The SVMs have been applied to several pattern recognition tasks, obtaining results superior to those of other learning techniques in various applications.
Keywords: Machine Learning, Classification, Support Vector Machines

1 Introdução

As Máquinas de Vetores de Suporte (SVMs, do Inglês *Support Vector Machines*) constituem uma técnica de aprendizado que vem recebendo crescente atenção da comunidade de Aprendizado de Máquina (AM) [27]. Os resultados da aplicação dessa técnica são comparáveis e muitas vezes superiores aos obtidos por outros algoritmos de aprendizado, como as Redes Neurais Artificiais (RNAs) [4, 14]. Exemplos de aplicações de sucesso podem ser encontrados em diversos domínios, como na categorização de textos [19], na análise de imagens [20, 33] e em Bioinformática [30, 34].

As SVMs são embasadas pela teoria de aprendizado estatístico, desenvolvida por Vapnik [41] a partir de estudos iniciados em [43]. Essa teoria estabelece uma série de princípios que devem ser seguidos na obtenção de classificadores com boa generalização, definida como a sua capacidade de prever corretamente a classe de novos dados do mesmo domínio em que o aprendizado ocorreu.

[1]Centro de Matemática, Computação e Cognição, Universidade Federal do ABC, Rua Catequese, 242, CEP 09090-400, Santo André, SP
ana.lorena@ufabc.edu.br
[2]Departamento de Ciências de Computação, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, Caixa Postal 668, CEP 13560-970, São Carlos, SP
andre@icmc.usp.br

Iniciando este artigo, na Seção 2 são apresentados alguns conceitos básicos de AM. Uma breve introdução aos principais conceitos da teoria de aprendizado estatístico é então apresentada na Seção 3. A partir deles, na Seção 4 as SVMs são formuladas para a definição de fronteiras lineares para a separação de conjuntos de dados binários. A seguir, na Seção 5 as SVMs da Seção 4 são estendidas de forma a definir fronteiras não lineares. Concluindo, na Seção 6 são apresentadas algumas discussões dos conceitos vistos e as considerações finais deste artigo.

2 Conceitos Básicos de Aprendizado de Máquina

As técnicas de AM empregam um princípio de inferência denominado indução, no qual obtém-se conclusões genéricas a partir de um conjunto particular de exemplos. O aprendizado indutivo pode ser dividido em dois tipos principais: supervisionado e não- supervisionado.

No aprendizado supervisionado tem-se a figura de um professor externo, o qual apresenta o conhecimento do ambiente por conjuntos de exemplos na forma: entrada, saída desejada [14]. O algoritmo de AM extrai a representação do conhecimento a partir desses exemplos. O objetivo é que a representação gerada seja capaz de produzir saídas corretas para novas entradas não apresentadas previamente.

No aprendizado não-supervisionado não há a presença de um professor, ou seja, não existem exemplos rotulados. O algoritmo de AM aprende a representar (ou agrupar) as entradas submetidas segundo uma medida de qualidade. Essas técnicas são utilizadas principalmente quando o objetivo for encontrar padrões ou tendências que auxiliem no entendimento dos dados [39].

O tipo de aprendizado abordado neste trabalho é o supervisionado. Neste caso, dado um conjunto de exemplos rotulados na forma $(\mathbf{x}_i, y_i)$, em que $\mathbf{x}_i$ representa um exemplo e y_i denota o seu rótulo, deve-se produzir um classificador, também denominado modelo, preditor ou hipótese, capaz de predizer precisamente o rótulo de novos dados. Esse processo de indução de um classificador a partir de uma amostra de dados é denominado treinamento. O classificador obtido também pode ser visto como uma função f, a qual recebe um dado $\mathbf{x}$ e fornece uma predição y.

Os rótulos ou classes representam o fenômeno de interesse sobre o qual se deseja fazer previsões. Neste trabalho, considera-se o caso em que os rótulos assumem valores discretos $1, \ldots, k$. Tem-se então um problema de classificação. Caso os rótulos possuam valores contínuos, tem-se uma regressão [27]. Um problema de classificação no qual $k = 2$ é denominado binário. Para $k > 2$, configura-se um problema multiclasses.

Cada exemplo, também referenciado por dado ou caso, é tipicamente representado

por um vetor de características. Cada característica, também denominada atributo, expressa um determinado aspecto do exemplo [29]. Normalmente, há dois tipos básicos de atributos: nominal e contínuo. Um atributo é definido como nominal (ou categórico) quando não existe uma ordem entre os valores que ele pode assumir (por exemplo, entre cores). No caso de atributos contínuos, é possível definir uma ordem linear nos valores assumidos (por exemplo, entre pesos $\in \Re$).

Um requisito importante para as técnicas de AM é que elas sejam capazes de lidar com dados imperfeitos, denominados ruídos. Muitos conjuntos de dados apresentam esse tipo de caso, sendo alguns erros comuns a presença de dados com rótulos e/ou atributos incorretos. A técnica de AM deve idealmente ser robusta a ruídos presentes nos dados, procurando não fixar a obtenção dos classificadores sobre esse tipo de caso. Deve-se também minimizar a influência de *outliers* no processo de indução. Os *outliers* são exemplos muito distintos dos demais presentes no conjunto de dados. Esses dados podem ser ruídos ou casos muito particulares, raramente presentes no domínio.

Os conceitos referentes à geração de um classificador a partir do aprendizado supervisionado são representados de forma simplificada na Figura 1. Tem-se nessa figura um conjunto com n dados. Cada dado $\mathbf{x}_i$ possui m atributos, ou seja, $\mathbf{x}_i = (x_{i1}, \ldots, x_{im})$. As variáveis y_i representam as classes. A partir dos exemplos e as suas respectivas classes, o algoritmo de AM extrai um classificador. Pode-se considerar que o modelo gerado fornece uma descrição compacta dos dados fornecidos [1].

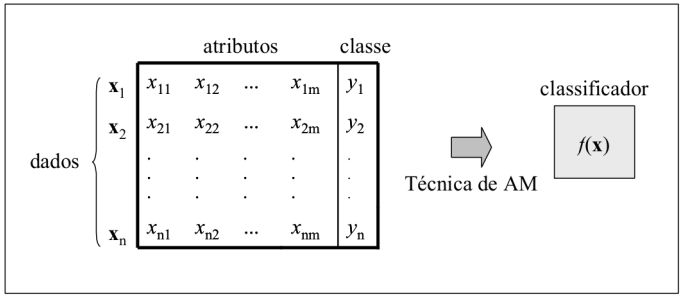

Figura 1. Indução de classificador em aprendizado supervisionado

A obtenção de um classificador por um algoritmo de AM a partir de uma amostra de dados também pode ser considerada um processo de busca [27]. Procura-se, entre todas as hipóteses que o algoritmo é capaz de gerar a partir dos dados, aquela com melhor capacidade de descrever o domínio em que ocorre o aprendizado.

Para estimar a taxa de predições corretas ou incorretas (também denominadas taxa de acerto e taxa de erro, respectivamente) obtidas por um classificador sobre novos dados, o con-

junto de exemplos é, em geral, dividido em dois subconjuntos disjuntos: de treinamento e de teste. O subconjunto de treinamento é utilizado no aprendizado do conceito e o subconjunto de teste é utilizado para medir o grau de efetividade do conceito aprendido na predição da classe de novos dados.

Um conceito comumente empregado em AM é o de generalização de um classificador, definida como a sua capacidade de prever corretamente a classe de novos dados. No caso em que o modelo se especializa nos dados utilizados em seu treinamento, apresentando uma baixa taxa de acerto quando confrontado com novos dados, tem-se a ocorrência de um super-ajustamento (*overfitting*). É também possível induzir hipóteses que apresentem uma baixa taxa de acerto mesmo no subconjunto de treinamento, configurando uma condição de sub-ajustamento (*underfitting*). Essa situação pode ocorrer, por exemplo, quando os exemplos de treinamento disponíveis são pouco representativos ou quando o modelo obtido é muito simples [29]. Na Seção 3, esses conceitos são ilustrados e discutidos novamente. São feitas então considerações e motivações sobre a escolha de classificadores com boa capacidade de generalização.

3 A Teoria de Aprendizado Estatístico

Seja f um classificador e F o conjunto de todos os classificadores que um determinado algoritmo de AM pode gerar. Esse algoritmo, durante o processo de aprendizado, utiliza um conjunto de treinamento T, composto de n pares $(\mathbf{x_i}, y_i)$, para gerar um classificador particular $\hat{f} \in F$.

Considere, por exemplo, o conjunto de treinamento da Figura 2 [38]. O objetivo do processo de aprendizado é encontrar um classificador que separe os dados das classes "círculo" e "triângulo". As funções ou hipóteses consideradas são ilustradas na figura por meio das bordas, também denominadas fronteiras de decisão, traçadas entre as classes.

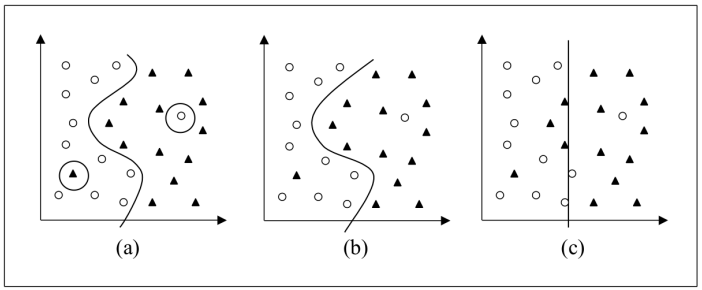

Figura 2. Conjunto de treinamento binário e três diferentes hipóteses

Na imagem da Figura 2a, tem-se uma hipótese que classifica corretamente todos os exemplos do conjunto de treinamento, incluindo dois possíveis ruídos. Por ser muito específica para o conjunto de treinamento, essa função apresenta elevada sucetibilidade a cometer erros quando confrontada com novos dados. Esse caso representa a ocorrência de um super-ajustamento do modelo aos dados de treinamento.

Um outro classificador poderia desconsiderar pontos pertencentes a classes opostas que estejam muito próximos entre si. A ilustração da Figura 2c representa essa alternativa. A nova hipótese considerada, porém, comete muitos erros, mesmo para casos que podem ser considerados simples. Tem-se assim a ocorrência de um sub-ajustamento, pois o classificador não é capaz de se ajustar mesmo aos exemplos de treinamento.

Um meio termo entre as duas funções descritas é representado na Figura 2b. Esse preditor tem complexidade intermediária e classifica corretamente grande parte dos dados, sem se fixar demasiadamente em qualquer ponto individual.

A Teoria de Aprendizado Estatístico (TAE) estabelece condições matemáticas que auxiliam na escolha de um classificador particular $\hat{f}$ a partir de um conjunto de dados de treinamento. Essas condições levam em conta o desempenho do classificador no conjunto de treinamento e a sua complexidade, com o objetivo de obter um bom desempenho também para novos dados do mesmo domínio.

3.1 Considerações sobre a Escolha do Classificador

Na aplicação da TAE, assume-se inicialmente que os dados do domínio em que o aprendizado está ocorrendo são gerados de forma independente e identicamente distribuída (i.i.d.) de acordo com uma distribuição de probabilidade $P(\mathbf{x}, y)$, que descreve a relação entre os dados e os seus rótulos [5, 38]. O erro (também denominado risco) esperado de um classificador f para dados de teste pode então ser quantificado pela Equação 1 [28]. O risco esperado mede então a capacidade de generalização de f [31]. Na Equação 1, $c(f(\mathbf{x}), y)$ é uma função de custo relacionando a previsão $f(\mathbf{x})$ quando a saída desejada é y. Um tipo de função de custo comumente empregada em problemas de classificação é a 0/1, definida por $c(f(\mathbf{x}), y) = \frac{1}{2}\left|y - f\left(\mathbf{x}\right)\right|$ [38]. Essa função retorna o valor 0 se $\mathbf{x}$ é classificado corretamente e 1 caso contrário.

$$R\left(f\right) = \int c(f\left(\mathbf{x}\right), y) dP\left(\mathbf{x}, y\right) \tag{1}$$

Infelizmente, não é possível minimizar o risco esperado apresentado na Equação 1 diretamente, uma vez que em geral a distribuição de probabilidade $P(\mathbf{x}, y)$ é desconhecida [28]. Tem-se unicamente a informação dos dados de treinamento, também amostrados de $P(\mathbf{x}, y)$. Normalmente utiliza-se o princípio da indução para inferir uma função $\hat{f}$ que mi-

nimize o erro sobre esses dados e espera-se que esse procedimento leve também a um menor erro sobre os dados de teste [38]. O risco empírico de f, fornecido pela Equação 2, mede o desempenho do classificador nos dados de treinamento, por meio da taxa de classificações incorretas obtidas em T [28].

$$R_{emp}(f) = \frac{1}{n}\sum_{i=1}^{n} c(f(\mathbf{x}_i), y_i) \tag{2}$$

Esse processo de indução com base nos dados de treinamento conhecidos constitui o princípio de minimização do risco empírico [38]. Assintoticamente, com $n \to \infty$, é possível estabelecer condições para o algoritmo de aprendizado que garantam a obtenção de classificadores cujos valores de risco empírico convergem para o risco esperado [28]. Para conjuntos de dados menores, porém, geralmente não é possível determinar esse tipo de garantia. Embora a minimização do risco empírico possa levar a um menor risco esperado, nem sempre isso ocorre. Considere, por exemplo, um classificador que memoriza todos os dados de treinamento e gera classificações aleatórias para outros exemplos [37]. Embora seu risco empírico seja nulo, seu risco esperado é 0,5.

A noção expressa nesses argumentos é a de que, permitindo que $\hat{f}$ seja escolhida a partir de um conjunto de funções amplo F, é sempre possível encontrar uma f com pequeno risco empírico. Porém, nesse caso os exemplos de treinamento podem se tornar pouco informativos para a tarefa de aprendizado, pois o classificador induzido pode se super-ajustar a eles. Deve-se então restringir a classe de funções da qual $\hat{f}$ é extraída. Existem diversas abordagens para tal. A TAE lida com essa questão considerando a complexidade (também referenciada por capacidade) da classe de funções que o algoritmo de aprendizado é capaz de obter [38]. Nessa direção, a TAE provê diversos limites no risco esperado de uma função de classificação, os quais podem ser empregados na escolha do classificador. A próxima seção relaciona alguns dos principais limites sobre os quais as SVMs se baseiam.

3.2 Limites no Risco Esperado

Um limite importante fornecido pela TAE relaciona o risco esperado de uma função ao seu risco empírico e a um termo de capacidade. Esse limite, apresentado na Equação 3, é garantido com probabilidade $1 - \theta$, em que $\theta \in [0, 1]$ [5].

$$R(f) \leqslant R_{emp}(f) + \sqrt{\frac{h\left(\ln\left(2n/h\right) + 1\right) - \ln\left(\theta/4\right)}{n}} \tag{3}$$

Nessa equação, h denota a dimensão Vapnik-Chervonenkis (VC) [41] da classe de

funções F à qual f pertence, n representa a quantidade de exemplos no conjunto de treinamento T e a parcela de raiz na soma é referenciada como termo de capacidade.

A dimensão VC h mede a capacidade do conjunto de funções F [5]. Quanto maior o seu valor, mais complexas são as funções de classificação que podem ser induzidas a partir de F. Dado um problema de classificação binário, essa dimensão é definida como o número máximo de exemplos que podem ser particionados em duas classes pelas funções contidas em F, para todas as possíveis combinações binárias desses dados.

Para ilustrar esse conceito, considere os três dados apresentados na Figura 3 [38]. Pode-se verificar que, para qualquer conformação arbitrária dos rótulos "círculo" e "triângulo" que esses dados possam assumir, é possível determinar retas capazes de separá-los. Porém, para os quatro pontos em $\Re^2$ ilustrados na Figura 4, existem rótulos para os dados que podem ser separados por uma reta (Figura 4a), mas também é possível definir rótulos tal que uma só reta seja incapaz de realizar a separação em classes (Figura 4b) [14]. Para uma divisão binária arbitrária desses quatro pontos em $\Re^2$, deve-se então recorrer a funções de complexidade superior à das retas. Essa observação se aplica a quaisquer quatro pontos no espaço bidimensional. Portanto, a dimensão VC do conjunto de funções lineares no espaço bidimensional é 3, uma vez que existe (pelo menos) uma configuração de três pontos nesse espaço que pode ser particionada por retas em todas as $2^3 = 8$ combinações binárias de rótulos.

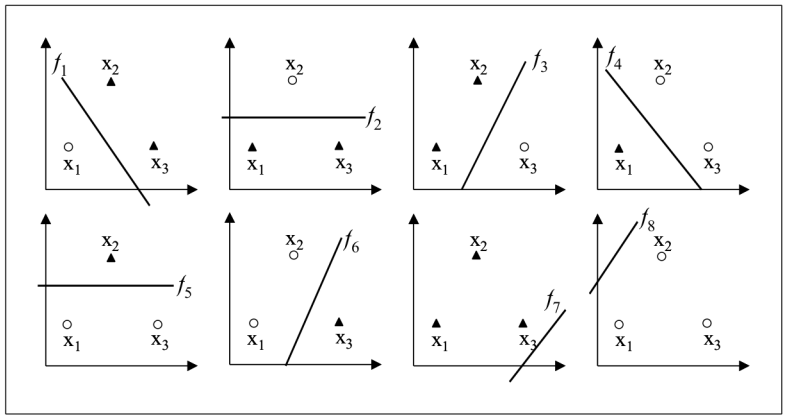

Figura 3. Separações de três dados em $\Re^2$ por meio de retas

A contribuição principal da Inequação 3 está em afirmar a importância de se controlar a capacidade do conjunto de funções F do qual o classificador é extraído. Interpretando-a em termos práticos, tem-se que o risco esperado pode ser minimizado pela escolha adequada,

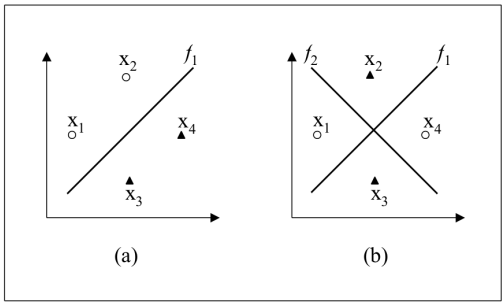

Figura 4. Separações de quatro dados em $\Re^2$ por meio de retas

por parte do algoritmo de aprendizado, de um classificador $\hat{f}$ que minimize o risco empírico e que pertença a uma classe de funções F com baixa dimensão VC h. Com esses objetivos, definiu-se um princípio de indução denominado minimização do risco estrutural [42].

Como no limite apresentado o termo de capacidade diz respeito à classe de funções F e o risco empírico refere-se a um classificador particular f, para minimizar ambas as parcelas divide-se inicialmente F em subconjuntos de funções com dimensão VC crescente [42]. É comum referir-se a esse processo como introduzir uma estrutura em F, sendo os subconjuntos definidos também denominados estruturas [38]. Minimiza-se então o limite sobre as estruturas introduzidas.

Considera-se subconjuntos F_i da seguinte forma: $F_0 \subset F_1 \subset \ldots \subset F_q \subset F$. Como cada F_i é maior com o crescimento do índice i, a capacidade do conjunto de funções que ele representa também é maior à medida que i cresce, ou seja, $h_0 < h_1 < \ldots < h_q < h$. Para um subconjunto particular F_k, seja $\widehat{f}_k$ o classificador com o menor risco empírico. A medida que k cresce, o risco empírico de $\widehat{f}_k$ diminui, uma vez que a complexidade do conjunto de classificadores é maior. Porém, o termo de capacidade aumenta com k. Como resultado, deve haver um valor ótimo $\bar{k}$ em que se obtém uma soma mínima do risco empírico e do termo de capacidade, minimizando assim o limite sobre o risco esperado. A escolha da função $\widehat{f}_{\bar{k}}$ constitui o princípio da minimização do risco estrutural. Os conceitos discutidos são ilustrados na Figura 5.

Embora o limite representado na Equação 3 tenha sido útil na definição do procedimento de minimização do risco estrutural, na prática surgem alguns problemas. Em primeiro lugar, computar a dimensão VC de uma classe de funções geralmente não é uma tarefa trivial. Soma-se a isso o fato de que o valor de h poder ser desconhecido ou infinito [28].

Para funções de decisão lineares do tipo $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x}$, entretando, existem resultados

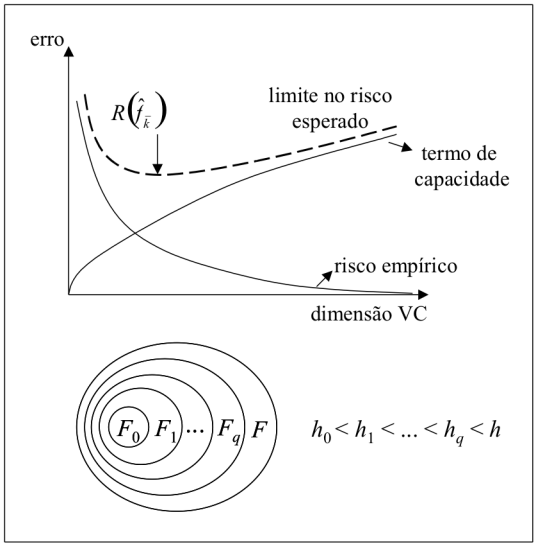

Figura 5. Princípio de minimização do risco estrutural [38]

alternativos que relacionam o risco esperado ao conceito de margem [37]. A margem de um exemplo tem relação com sua distância à fronteira de decisão induzida, sendo uma medida da confiança da previsão do classificador. Para um problema binário, em que $y_i \in \{-1, +1\}$, dada uma função f e um exemplo $\mathbf{x}_i$, a margem $\varrho(f(\mathbf{x}_i), y_i)$ com que esse dado é classificado por f pode ser calculada pela Equação 4 [37]. Logo, um valor negativo de $\varrho(\mathbf{x}_i, y_i)$ denota uma classificação incorreta.

$$\varrho\left(f(\mathbf{x}_i), y_i\right) = y_i f\left(\mathbf{x}_i\right) \tag{4}$$

Para obter a margem geométrica de um dado $\mathbf{x}_i$ em relação a uma fronteira linear $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b$, a qual mede efetivamente a distância de $\mathbf{x}_i$ à fronteira de decisão, divide-se o termo à direita da Equação 4 pela norma de $\mathbf{w}$, ou seja, por $\|\mathbf{w}\|$ [38]. Para exemplos incorretamente classificados, o valor obtido equivale à distância com sinal negativo. Para realizar uma diferenciação, a margem da Equação 4 será referenciada como margem de confiança.

A partir do conceito introduzido, é possível definir o erro marginal de uma função f ($R_\rho(f)$) sobre um conjunto de treinamento. Esse erro fornece a proporção de exemplos de treinamento cuja margem de confiança é inferior a uma determinada constante $\rho > 0$ (Equação 5) [37].

$$R_\rho(f) = \frac{1}{n}\sum_{i=1}^{n} I\left(y_i f(\mathbf{x}_i) < \rho\right) \tag{5}$$

Na Equação 5, $I(q) = 1$ se q é verdadeiro e $I(q) = 0$ se q é falso.

Existe uma constante c tal que, com probabilidade $1 - \theta \in [0, 1]$, para todo $\rho > 0$ e F correspondendo à classe de funções lineares $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x}$ com $\|\mathbf{x}\| \leq R$ e $\|\mathbf{w}\| \leq 1$, o seguinte limite se aplica [37]:

$$R(f) \leqslant R_\rho(f) + \sqrt{\frac{c}{n}\left(\frac{R^2}{\rho^2}\log^2\left(\frac{n}{\rho}\right) + \log\left(\frac{1}{\theta}\right)\right)} \tag{6}$$

Como na Equação 3, tem-se na Expressão 6 novamente o erro esperado limitado pela soma de uma medida de erro no conjunto de treinamento, neste caso o erro marginal, a um termo de capacidade. A interpretação do presente limite é de que uma maior margem ρ implica em um menor termo de capacidade. Entretanto, a maximização da margem pode levar a um aumento na taxa de erro marginal, pois torna-se mais difícil obedecer à restrição de todos os dados de treinamento estarem distantes de uma margem maior em relação ao hiperplano separador. Um baixo valor de ρ, em contrapartida, leva a um erro marginal menor, porém aumenta o termo de capacidade. Deve-se então buscar um compromisso entre a maximização da margem e a obtenção de um erro marginal baixo.

Como conclusão tem-se que, na geração de um classificador linear, deve-se buscar um hiperplano que tenha margem ρ elevada e cometa poucos erros marginais, minimizando assim o erro sobre os dados de teste e de treinamento, respectivamente. Esse hiperplano é denominado ótimo [38].

Existem diversos outros limites reportados na literatura, assim como outros tipos de medida de complexidade de uma classe de funções [28]. Um exemplo é a dimensão *fat-shattering*, que caracteriza o poder de um conjunto de funções em separar os dados com uma margem ρ [35]. Os limites apresentados anteriormente, embora possam ser considerados simplificados, provêm uma base teórica suficiente à compreenção das SVMs.

4 SVMs Lineares

As SVMs surgiram pelo emprego direto dos resultados fornecidos pela TAE. Nesta seção é apresentado o uso de SVMs na obtenção de fronteiras lineares para a separação de dados pertencentes a duas classes. A primeira formulação, mais simples, lida com problemas linearmente separáveis, definidos adiante [3]. Essa formulação foi posteriormente estendida

para definir fronteiras lineares sobre conjuntos de dados mais gerais [10]. A partir desses conceitos iniciais, na Seção 5 descreve-se a obtenção de fronteiras não lineares com SVMs, por meio de uma extensão das SVMs lineares.

4.1 SVMs com Margens Rígidas

As SVMs lineares com margens rígidas definem fronteiras lineares a partir de dados linearmente separáveis. Seja T um conjunto de treinamento com n dados $\mathbf{x}_i \in \mathrm{X}$ e seus respectivos rótulos $y_i \in \mathrm{Y}$, em que X constitui o espaço dos dados e $\mathrm{Y} = \{-1, +1\}$. T é linearmente separável se é possível separar os dados das classes $+1$ e -1 por um hiperplano [38].

Classificadores que separam os dados por meio de um hiperplano são denominados lineares [6]. A equação de um hiperplano é apresentada na Equação 7, em que $\mathbf{w} \cdot \mathbf{x}$ é o produto escalar entre os vetores $\mathbf{w}$ e $\mathbf{x}$, $\mathbf{w} \in \mathrm{X}$ é o vetor normal ao hiperplano descrito e $\frac{b}{\|\mathbf{w}\|}$ corresponde à distância do hiperplano em relação à origem, com $b \in \Re$.

$$f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b = 0 \tag{7}$$

Essa equação divide o espaço dos dados X em duas regiões: $\mathbf{w} \cdot \mathbf{x} + b > 0$ e $\mathbf{w} \cdot \mathbf{x} + b < 0$. Uma função sinal $g(\mathbf{x}) = \mathrm{sgn}(f(\mathbf{x}))$ pode então ser empregada na obtenção das classificações, conforme ilustrado na Equação 8 [37].

$$g(\mathbf{x}) = \mathrm{sgn}(f(\mathbf{x})) = \begin{cases} +1 \text{ se } \mathbf{w} \cdot \mathbf{x} + b > 0 \\ -1 \text{ se } \mathbf{w} \cdot \mathbf{x} + b < 0 \end{cases} \tag{8}$$

A partir de $f(\mathbf{x})$, é possível obter um número infinito de hiperplanos equivalentes, pela multiplicação de $\mathbf{w}$ e b por uma mesma constante [31]. Define-se o hiperplano canônico em relação ao conjunto T como aquele em que $\mathbf{w}$ e b são escalados de forma que os exemplos mais próximos ao hiperplano $\mathbf{w} \cdot \mathbf{x} + b = 0$ satisfaçam a Equação 9 [28].

$$|\mathbf{w} \cdot \mathbf{x}_i + b| = 1 \tag{9}$$

Essa forma implica nas inequações 10, resumidas na Expressão 11.

$$\begin{cases} \mathbf{w} \cdot \mathbf{x}_i + b \geqslant +1 & \text{se } y_i = +1 \\ \mathbf{w} \cdot \mathbf{x}_i + b \leqslant -1 & \text{se } y_i = -1 \end{cases} \tag{10}$$

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \geqslant 0, \ \ \forall(\mathbf{x}_i, y_i) \in T \tag{11}$$

Seja $\mathbf{x}_1$ um ponto no hiperplano H_1:$\mathbf{w} \cdot \mathbf{x} + b = +1$ e $\mathbf{x}_2$ um ponto no hiperplano H_2:$\mathbf{w} \cdot \mathbf{x} + b = -1$, conforme ilustrado na Figura 6. Projetando $\mathbf{x}_1 - \mathbf{x}_2$ na direção de $\mathbf{w}$, perpendicular ao hiperplano separador $\mathbf{w} \cdot \mathbf{x} + b = 0$, é possível obter a distância entre os hiperplanos H_1 e H_2 [6]. Essa projeção é apresentada na Equação 12.

$$(\mathbf{x}_1 - \mathbf{x}_2) \left(\frac{\mathbf{w}}{\|\mathbf{w}\|} \cdot \frac{(\mathbf{x}_1 - \mathbf{x}_2)}{\|\mathbf{x}_1 - \mathbf{x}_2\|} \right) \tag{12}$$

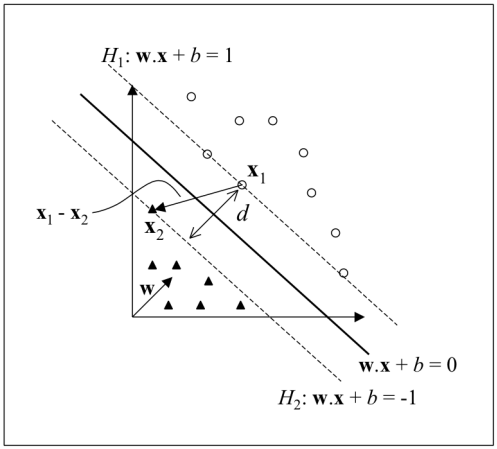

Figura 6. Cálculo da distância d entre os hiperplanos H_1 e H_2 [15]

Tem-se que $\mathbf{w} \cdot \mathbf{x}_1 + b = +1$ e $\mathbf{w} \cdot \mathbf{x}_2 + b = -1$. A diferença entre essas equações fornece $\mathbf{w} \cdot (\mathbf{x}_1 - \mathbf{x}_2) = 2$ [15]. Substituindo esse resultado na Equação 12, tem-se:

$$\frac{2\,(\mathbf{x}_1 - \mathbf{x}_2)}{\|\mathbf{w}\|\,\|\mathbf{x}_1 - \mathbf{x}_2\|} \tag{13}$$

Como deseja-se obter o comprimento do vetor projetado, toma-se a norma da Equação 13, obtendo:

$$\frac{2}{\|\mathbf{w}\|} \tag{14}$$

Essa é a distância d, ilustrada na Figura 6, entre os hiperplanos H_1 e H_2, paralelos ao hiperplano separador. Como $\mathbf{w}$ e b foram escalados de forma a não haver exemplos entre H_1

e H_2, $\frac{1}{\|\mathbf{w}\|}$ é a distância mínima entre o hiperplano separador e os dados de treinamento. Essa distância é definida como a margem geométrica do classificador linear [6].

A partir das considerações anteriores, verifica-se que a maximização da margem de separação dos dados em relação a $\mathbf{w} \cdot \mathbf{x} + b = 0$ pode ser obtida pela minimização de $\|\mathbf{w}\|$ [5]. Dessa forma, recorre-se ao seguinte problema de otimização [38]:

$$\underset{\mathbf{w},b}{\text{Minimizar}} \quad \frac{1}{2} \|\mathbf{w}\|^2 \tag{15}$$

$$\text{Com as restrições: } y_i \left(\mathbf{w} \cdot \mathbf{x}_i + b\right) - 1 \geqslant 0, \quad \forall i = 1, \ldots, n \tag{16}$$

As restrições são impostas de maneira a assegurar que não haja dados de treinamento entre as margens de separação das classes. Por esse motivo, a SVM obtida possui também a nomenclatura de SVM com margens rígidas.

O problema de otimização obtido é quadrático, cuja solução possui uma ampla e estabelecida teoria matemática [38]. Como a função objetivo sendo minimizada é convexa e os pontos que satisfazem as restrições formam um conjunto convexo, esse problema possui um único mínimo global [31]. Problemas desse tipo podem ser solucionados com a introdução de uma função Lagrangiana, que engloba as restrições à função objetivo, associadas a parâmetros denominados multiplicadores de Lagrange α_i (Equação 17) [38].

$$L\left(\mathbf{w}, b, \boldsymbol{\alpha}\right) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i \left(y_i \left(\mathbf{w} \cdot \mathbf{x}_i + b\right) - 1\right) \tag{17}$$

A função Lagrangiana deve ser minimizada, o que implica em maximizar as variáveis α_i e minimizar $\mathbf{w}$ e b [28]. Tem-se então um ponto de sela, no qual:

$$\frac{\partial L}{\partial b} = 0 \quad \text{e} \quad \frac{\partial L}{\partial \mathbf{w}} = 0 \tag{18}$$

A resolução dessas equações leva aos resultados representados nas expressões 19 e 20.

$$\sum_{i=1}^{n} \alpha_i y_i = 0 \tag{19}$$

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i \tag{20}$$

Substituindo as equações 19 e 20 na Equação 17, obtém-se o seguinte problema de otimização:

$$\underset{\boldsymbol{\alpha}}{\text{Maximizar}} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \left(\mathbf{x}_i \cdot \mathbf{x}_j\right) \tag{21}$$

$$\text{Com as restrições:} \quad \begin{cases} \alpha_i \geqslant 0, \ \ \forall i = 1, \ldots, n \\ \sum\limits_{i=1}^{n} \alpha_i y_i = 0 \end{cases} \tag{22}$$

Essa formulação é denominada forma dual, enquanto o problema original é referenciado como forma primal. A forma dual possui os atrativos de apresentar restrições mais simples e permitir a representação do problema de otimização em termos de produtos internos entre dados, o que será útil na posterior não-linearização das SVMs (Seção 5). É interessante observar também que o problema dual é formulado utilizando apenas os dados de treinamento e os seus rótulos.

Seja $\boldsymbol{\alpha}^*$ a solução do problema dual e $\mathbf{w}^*$ e b^* as soluções da forma primal. Obtido o valor de $\boldsymbol{\alpha}^*$, $\mathbf{w}^*$ pode ser determinado pela Equação 20. O parâmetro b^* é definido por $\boldsymbol{\alpha}^*$ e por condições de Kühn-Tucker, provenientes da teoria de otimização com restrições e que devem ser satisfeitas no ponto ótimo. Para o problema dual formulado, tem-se [33]:

$$\alpha_i^* \left(y_i \left(\mathbf{w}^* \cdot \mathbf{x}_i + b^*\right) - 1\right) = 0, \ \forall i = 1, \ldots, n \tag{23}$$

Observa-se nessa equação que α_i^* pode ser diferente de 0 somente para os dados que se encontram sobre os hiperplanos H_1 e H_2. Estes são os exemplos que se situam mais próximos ao hiperplano separador, exatamente sobre as margens. Para os outros casos, a condição apresentada na Equação 23 é obedecida apenas com $\alpha_i^* = 0$. Esses pontos não participam então do cálculo de $\mathbf{w}^*$ (Equação 20). Os dados que possuem $\alpha_i^* > 0$ são denominados vetores de suporte (SVs, do Inglês *Support Vectors*) e podem ser considerados os dados mais informativos do conjunto de treinamento, pois somente eles participam na determinação da equação do hiperplano separador (Equação 26) [5].

O valor de b^* é calculado a partir dos SVs e das condições representadas na Equação 23 [38]. Computa-se a média apresentada na Equação 24 sobre todos $\mathbf{x}_j$ tal que $\alpha_j^* > 0$, ou seja, todos os SVs. Nessa equação, n_{SV} denota o número de SVs e SV representa o conjunto dos SVs.

$$b^* = \frac{1}{n_{\text{SV}}} \sum_{\mathbf{x}_j \in \text{ SV}} \frac{1}{y_j} - \mathbf{w}^* \cdot \mathbf{x}_j \tag{24}$$

Substituindo $\mathbf{w}^*$ pela expressão na Equação 20, tem-se:

$$b^* = \frac{1}{n_{\text{SV}}} \sum_{\mathbf{x}_j \in \text{SV}} \left(\frac{1}{y_j} - \sum_{\mathbf{x}_i \in \text{SV}} \alpha_i^* y_i \mathbf{x}_i \cdot \mathbf{x}_j \right) \tag{25}$$

Como resultado final, tem-se o classificador $g(\mathbf{x})$ apresentado na Equação 26, em que sgn representa a função sinal, $\mathbf{w}^*$ é fornecido pela Equação 20 e b^* pela Equação 25.

$$g\left(\mathbf{x}\right) = \text{sgn}\left(f\left(\mathbf{x}\right)\right) = \text{sgn}\left(\sum_{\mathbf{x}_i \in \text{SV}} y_i \alpha_i^* \mathbf{x}_i \cdot \mathbf{x} + b^* \right) \tag{26}$$

Esta função linear representa o hiperplano que separa os dados com maior margem, considerado aquele com melhor capacidade de generalização de acordo com a TAE. Essa característica difere as SVMs lineares de margens rígidas das Redes Neurais Perceptron, em que o hiperplano obtido na separação dos dados pode não corresponder ao de maior margem de separação.

4.2 SVMs com Margens Suaves

Em situações reais, é difícil encontrar aplicações cujos dados sejam linearmente separáveis. Isso se deve a diversos fatores, entre eles a presença de ruídos e *outliers* nos dados ou à própria natureza do problema, que pode ser não linear. Nesta seção as SVMs lineares de margens rígidas são estendidas para lidar com conjuntos de treinamento mais gerais. Para realizar essa tarefa, permite-se que alguns dados possam violar a restrição da Equação 16. Isso é feito com a introdução de variáveis de folga ξ_i, para todo $i = 1, \ldots, n$ [37]. Essas variáveis relaxam as restrições impostas ao problema de otimização primal, que se tornam [38]:

$$y_i \left(\mathbf{w} \cdot \mathbf{x}_i + b\right) \geqslant 1 - \xi_i, \quad \xi_i \geqslant 0, \ \ \forall i = 1, \ldots, n \tag{27}$$

A aplicação desse procedimento suaviza as margens do classificador linear, permitindo que alguns dados permaneçam entre os hiperplanos H_1 e H_2 e também a ocorrência de alguns erros de classificação. Por esse motivo, as SVMs obtidas neste caso também podem ser referenciadas como SVMs com margens suaves.

Um erro no conjunto de treinamento é indicado por um valor de ξ_i maior que 1. Logo, a soma dos ξ_i representa um limite no número de erros de treinamento [5]. Para levar em

consideração esse termo, minimizando assim o erro sobre os dados de treinamento, a função objetivo da Equação 15 é reformulada como [5]:

$$\underset{\mathbf{w},b,\boldsymbol{\xi}}{\text{Minimizar}} \quad \frac{1}{2}\left\|\mathbf{w}\right\|^2 + C\left(\sum_{i=1}^{n}\xi_i\right) \tag{28}$$

A constante C é um termo de regularização que impõe um peso à minimização dos erros no conjunto de treinamento em relação à minimização da complexidade do modelo [31]. A presença do termo $\sum_{i=1}^{n}\xi_i$ no problema de otimização também pode ser vista como uma minimização de erros marginais, pois um valor de $\xi_i \in (0,1]$ indica um dado entre as margens. Tem-se então uma formulação de acordo com os princípios da TAE discutidos na Seção 3.

Novamente o problema de otimização gerado é quadrático, com as restrições lineares apresentadas na Equação 27. A sua solução envolve passos matemáticos semelhantes aos apresentados anteriormente, com a introdução de uma função Lagrangiana e tornando suas derivadas parciais nulas. Tem-se como resultado o seguinte problema dual:

$$\underset{\boldsymbol{\alpha}}{\text{Maximizar}} \quad \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j\left(\mathbf{x}_i\cdot\mathbf{x}_j\right) \tag{29}$$

$$\text{Com as restrições: } \begin{cases} 0 \leqslant \alpha_i \leqslant C, \quad \forall i = 1,\ldots,n \\ \sum_{i=1}^{n}\alpha_i y_i = 0 \end{cases} \tag{30}$$

Pode-se observar que essa formulação é igual à apresentada para as SVMs de margens rígidas, a não ser pela restrição nos α_i, que agora são limitados pelo valor de C.

Seja $\boldsymbol{\alpha}^*$ a solução do problema dual, enquanto $\mathbf{w}^*$, b^* e $\boldsymbol{\xi}^*$ denotam as soluções da forma primal. O vetor $\mathbf{w}^*$ continua sendo determinado pela Equação 20. As variáveis ξ_i^* podem ser calculadas pela Equação 31 [11].

$$\xi_i^* = \max\left\{0, 1 - y_i\sum_{j=1}^{n} y_j\alpha_j^*\mathbf{x}_j\cdot\mathbf{x}_i + b^*\right\} \tag{31}$$

A variável b^* provém novamente de $\boldsymbol{\alpha}^*$ e de condições de Kühn-Tucker, que neste caso são [33]:

$$\alpha_i^* \left(y_i \left(\mathbf{w}^* \cdot \mathbf{x}_i + b^* \right) - 1 + \xi_i^* \right) = 0 \tag{32}$$

$$\left(C - \alpha_i^* \right) \xi_i^* = 0 \tag{33}$$

Como nas SVMs de margens rígidas, os pontos $\mathbf{x}_i$ para os quais $\alpha_i^* > 0$ são denominados vetores de suporte (SVs), sendo os dados que participam da formação do hiperplano separador. Porém, neste caso, pode-se distinguir tipos distintos de SVs [33]. Se $\alpha_i^* < C$, pela Equação 33, $\xi_i^* = 0$ e então, da Equação 32, estes SVs encontram-se sobre as margens e também são denominados livres. Os SVs para os quais $\alpha_i^* = C$ podem representar três casos [33]: erros, se $\xi_i^* > 1$; pontos corretamente classificados, porém entre as margens, se $0 < \xi_i^* \leqslant 1$; ou pontos sobre as margens, se $\xi_i^* = 0$. O último caso ocorre raramente e os SVs anteriores são denominados limitados. Na Figura 7 são ilustrados os possíveis tipos de SVs. Pontos na cor cinza representam SVs livres. SVs limitados são ilustrados em preto. Pontos pretos com bordas extras correspondem a SVs limitados que são erros de treinamento. Todos os outros dados, em branco, são corretamente classificados e encontram-se fora das margens, possuindo $\xi_i^* = 0$ e $\alpha_i^* = 0$.

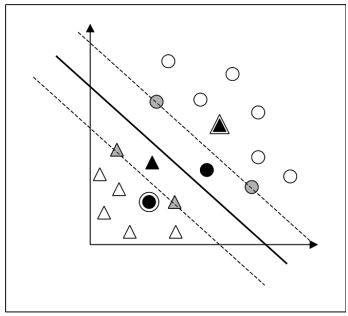

Figura 7. Tipos de SVs: livres (cor cinza) e limitados (cor preta) [31]

Para calcular b^*, computa-se a média da Equação 24 sobre todos SVs $\mathbf{x}_j$ entre as margens, ou seja, com $\alpha_i^* < C$ [38].

Tem-se como resultado final a mesma função de classificação representada na Equação 26, porém neste caso as variáveis α_i^* são determinadas pela solução da Expressão 29 com as restrições da Equação 30.

5 SVMs Não Lineares

As SVMs lineares são eficazes na classificação de conjuntos de dados linearmente separáveis ou que possuam uma distribuição aproximadamente linear, sendo que a versão de margens suaves tolera a presença de alguns ruídos e *outliers*. Porém, há muitos casos em que não é possível dividir satisfatoriamente os dados de treinamento por um hiperplano. Um exemplo é apresentado na Figura 8a, em que o uso de uma fronteira curva seria mais adequada na separação das classes.

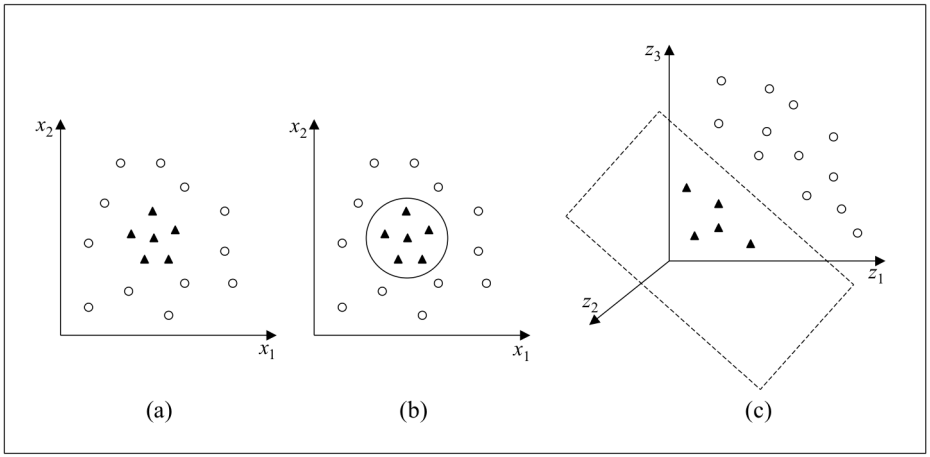

Figura 8. (a) Conjunto de dados não linear; (b) Fronteira não linear no espaço de entradas; (c) Fronteira linear no espaço de características [28]

As SVMs lidam com problemas não lineares mapeando o conjunto de treinamento de seu espaço original, referenciado como de entradas, para um novo espaço de maior dimensão, denominado espaço de características (*feature space*) [15]. Seja $\mathbf{\Phi} : \mathrm{X} \rightarrow \Im$ um mapeamento, em que X é o espaço de entradas e $\Im$ denota o espaço de características. A escolha apropriada de $\mathbf{\Phi}$ faz com que o conjunto de treinamento mapeado em $\Im$ possa ser separado por uma SVM linear.

O uso desse procedimento é motivado pelo teorema de Cover [14]. Dado um conjunto de dados não linear no espaço de entradas X, esse teorema afirma que X pode ser transformado em um espaço de características $\Im$ no qual com alta probabilidade os dados são linearmente separáveis. Para isso duas condições devem ser satisfeitas. A primeira é que a transformação seja não linear, enquanto a segunda é que a dimensão do espaço de características seja suficientemente alta.

Para ilustrar esses conceitos, considere o conjunto de dados apresentado na Figura 8a [28]. Transformando os dados de $\Re^2$ para $\Re^3$ com o mapeamento representado na Equação 34, o conjunto de dados não linear em $\Re^2$ torna-se linearmente separável em $\Re^3$ (Figura 8c). É possível então encontrar um hiperplano capaz de separar esses dados, descrito na Equação 35. Pode-se verificar que a função apresentada, embora linear em $\Re^3$ (Figura 8c), corresponde a uma fronteira não linear em $\Re^2$ (Figura 8b).

$$\mathbf{\Phi}\left(\mathbf{x}\right) = \Phi\left(x_1, x_2\right) = \left(x_1^2, \sqrt{2}x_1x_2, x_2^2\right) \tag{34}$$

$$f\left(\mathbf{x}\right) = \mathbf{w} \cdot \mathbf{\Phi}\left(\mathbf{x}\right) + b = w_1x_1^2 + w_2\sqrt{2}x_1x_2 + w_3x_2^2 + b = 0 \tag{35}$$

Logo, mapea-se inicialmente os dados para um espaço de maior dimensão utilizando $\mathbf{\Phi}$ e aplica-se a SVM linear sobre este espaço. Essa encontra o hiperplano com maior margem de separação, garantindo assim uma boa generalização. Utiliza-se a versão de SVM linear com margens suaves, que permite lidar com ruídos e *outliers* presentes nos dados. Para realizar o mapeamento, aplica-se $\mathbf{\Phi}$ aos exemplos presentes no problema de otimização representado na Equação 29, conforme ilustrado a seguir:

$$\underset{\boldsymbol{\alpha}}{\text{Maximizar}} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i\alpha_j y_i y_j \left(\mathbf{\Phi}(\mathbf{x}_i) \cdot \mathbf{\Phi}(\mathbf{x}_j)\right) \tag{36}$$

Sob as restrições da Equação 30. De forma semelhante, o classificador extraído se torna:

$$g\left(\mathbf{x}\right) = \operatorname{sgn}\left(f\left(\mathbf{x}\right)\right) = \operatorname{sgn}\left(\sum_{\mathbf{x}_i \in \text{ SV}} \alpha_i^* y_i \mathbf{\Phi}\left(\mathbf{x}_i\right) \cdot \mathbf{\Phi}\left(\mathbf{x}\right) + b^*\right) \tag{37}$$

Em que b^* é adaptado da Equação 24 também aplicando o mapeamento aos dados:

$$b^* = \frac{1}{n_{\text{SV}:\alpha^* < C}} \sum_{\mathbf{x}_j \in \text{ SV} \,:\, \alpha_j^* < C} \left(\frac{1}{y_j} - \sum_{\mathbf{x}_i \in \text{ SV}} \alpha_i^* y_i \mathbf{\Phi}\left(\mathbf{x}_i\right) \cdot \mathbf{\Phi}\left(\mathbf{x}_j\right)\right) \tag{38}$$

Como $\Im$ pode ter dimensão muito alta (até mesmo infinita), a computação de $\mathbf{\Phi}$ pode ser extremamente custosa ou inviável. Porém, percebe-se pelas equações 36, 37 e 38 que a única informação necessária sobre o mapeamento é de como realizar o cálculo de produtos

escalares entre os dados no espaço de características, pois tem-se sempre $\mathbf{\Phi}(\mathbf{x}_i) \cdot \mathbf{\Phi}(\mathbf{x}_j)$, para dois dados $\mathbf{x}_i$ e $\mathbf{x}_j$, em conjunto. Isso é obtido com o uso de funções denominadas Kernels.

Um Kernel K é uma função que recebe dois pontos $\mathbf{x}_i$ e $\mathbf{x}_j$ do espaço de entradas e computa o produto escalar desses dados no espaço de características [16]. Tem-se então:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{\Phi}(\mathbf{x}_i) \cdot \mathbf{\Phi}(\mathbf{x}_j) \tag{39}$$

Para o mapeamento apresentado na Equação 34 e dois dados $\mathbf{x}_i = (x_{1i}, x_{2i})$ e $\mathbf{x}_j = (x_{1j}, x_{2j})$ em $\Re^2$, por exemplo, o Kernel é dado por [15]:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \left(x_{1i}^2,\ \sqrt{2}\ x_{1i}\ x_{2i},\ x_{2i}^2\right) \cdot \left(x_{1j}^2,\ \sqrt{2}\ x_{1j}\ x_{2j},\ x_{2j}^2\right) = (\mathbf{x}_i \cdot \mathbf{x}_j)^2 \tag{40}$$

É comum empregar a função Kernel sem conhecer o mapeamento $\mathbf{\Phi}$, que é gerado implicitamente. A utilidade dos Kernels está, portanto, na simplicidade de seu cálculo e em sua capacidade de representar espaços abstratos.

Para garantir a convexidade do problema de otimização formulado na Equação 36 e também que o Kernel represente mapeamentos nos quais seja possível o cálculo de produtos escalares conforme a Equação 39, utiliza-se funções Kernel que seguem as condições estabelecidas pelo teorema de Mercer [26, 37]. De forma simplificada, um Kernel que satisfaz as condições de Mercer é caracterizado por dar origem a matrizes positivas semi-definidas $\mathbf{K}$, em que cada elemento K_{ij} é definido por $K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$, para todo $i, j = 1, \ldots, n$ [16].

Alguns dos Kernels mais utilizados na prática são os Polinomiais, os Gaussianos ou RBF (*Radial-Basis Function*) e os Sigmoidais, listados na Tabela 1. Cada um deles apresenta parâmetros que devem ser determinados pelo usuário, indicados também na tabela. O Kernel Sigmoidal, em particular, satisfaz as condições de Mercer apenas para alguns valores de δ e κ. Os Kernels Polinomiais com $d = 1$ também são denominados lineares.

Tipo de Kernel	**Função** $K(\mathbf{x}_i, \mathbf{x}_j)$	**Parâmetros**
Polinomial	$(\delta\,(\mathbf{x}_i \cdot \mathbf{x}_j) + \kappa)^d$	δ, κ e d
Gaussiano	$\exp\left(-\sigma \left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2\right)$	σ
Sigmoidal	$\tanh\left(\delta\,(\mathbf{x}_i \cdot \mathbf{x}_j) + \kappa\right)$	δ e κ

Tabela 1. Funções Kernel mais comuns [8]

6 Considerações Finais

Neste texto foram descritos os conceitos básicos a respeito das SVMs para problemas de classificação, os quais também podem ser consultados em [21, 22]. Com princípios embasados na teoria de aprendizado estatístico, essa técnica de AM se caracteriza por apresentar uma boa capacidade de generalização.

As SVMs também são robustas diante de dados de grande dimensão, sobre os quais outras técnicas de aprendizado comumente obtêm classificadores super ou sub ajustados. Outra característica atrativa é a convexidade do problema de otimização formulado em seu treinamento, que implica na existência de um único mínimo global. Essa é uma vantagem das SVMs sobre, por exemplo, as Redes Neurais Artificiais (RNAs) Perceptron Multicamadas (*Multilayer Perceptron*) [4, 14], em que há mínimos locais na função objetivo minimizada. Além disso, o uso de funções Kernel na não-linearização das SVMs torna o algoritmo eficiente, pois permite a construção de simples hiperplanos em um espaço de alta dimensão de forma tratável do ponto de vista computacional [5].

Entre as principais limitações das SVMs encontram-se a sua sensibilidade a escolhas de valores de parâmetros e a dificuldade de interpretação do modelo gerado por essa técnica, problemas que têm sido abordados em diversos trabalhos recentes, como [9, 12, 32, 18, 24, 40] e [13, 7, 44], respectivamente.

Observou-se no decorrer deste tutorial que o raciocínio empregado pelas SVMs na obtenção do classificador final leva a um problema de otimização dual em termos dos dados de treinamento. Porém, a forma de solução desse problema não foi apresentada. Existem diversos pacotes matemáticos capazes de solucionar problemas quadráticos com restrições. Contudo, eles geralmente não são adequados a aplicações de AM, que em geral se caracterizam pela necessidade de lidar com um grande volume de dados. Diversas técnicas e estratégias foram então propostas para adaptar a solução do problema de otimização das SVMs a aplicações de larga escala. Em geral, recorre-se a alguma estratégia decomposicional, em que subproblemas menores são otimizados a cada passo do algoritmo. Uma discussão mais detalhada a respeito dos métodos e algoritmos comumente empregados nesse processo pode ser encontrada em [11].

O presente artigo também se limitou a apresentar a formulação original das SVMs, a qual é capaz de lidar apenas com problemas de classificação binários. Existe uma série de técnicas que podem ser empregadas na generalização das SVMs para a solução de problemas multiclasses. Pode-se recorrer à decomposição do problema multiclasses em vários subproblemas binários ou reformular o algoritmo de treinamento das SVMs em versões multiclasses. Em geral, esse último procedimento leva a algoritmos computacionalmente custosos [17]. Por esse motivo, a estratégia decomposicional é empregada mais frequentemente. Revisões a respeito da obtenção de previsões multiclasses com SVMs podem ser consultadas

em [22, 23, 25].

As SVMs também podem ser aplicadas na solução de problemas de regressão e no agrupamento de dados (aprendizado não supervisionado). Contudo, o problema de otimização para o seu treinamento deve ser reformulado para lidar com as características e objetivos desses problemas. Mais detalhes podem ser consultados em [2, 36].

7 Agradecimentos

À FAPESP e ao CNPq pelo apoio financeiro.

8 Referências

[1] J. A. Baranauskas and M. C. Monard. Reviewing some machine learning concepts and methods. Technical Report 102, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos. Disponível em: `ftp://ftp.icmc.usp.br/pub/BIBLIOTECA/rel_tec/RT_102.ps.zip`, Fevereiro 2000.

[2] A. Ben-Hur, D. Horn, H. T. Siegelmann, and V. N. Vapnik. A support vector clustering method. In *Proceedings of the International Conference on Pattern Recognition (ICPR'00)*, volume 2, pages 724–727, 2000.

[3] B. E. Boser, I. L. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the 5th Annual Workshop on Computational Learning Theory*, pages 144–152, Pittsburg, Pennsylvania, US, 1992.

[4] A. Braga, A. C. P. L. F. Carvalho, and T. B. Ludermir. *Redes Neurais Artificiais: Teoria e Aplicações*. Editora LTC, 2000.

[5] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Knowledge Discovery and Data Mining*, 2(2):1–43, 1998.

[6] C. Campbell. An introduction to kernel methods. In R. J. Howlett and L. C. Jain, editors, *Radial Basis Function Networks: Design and Applications*, pages 155–192, Berlin, 2000. Springer Verlag.

[7] J. L. Castro, L. D. Flores-Hidalgo, C. J. Mantas, and J. M. Puche. Extraction of fuzzy rules from support vector machines. *Fuzzy Sets and Systems archive*, 158(18):2057–2077, 2007.

[8] C.-C. Chang and C.-J. Lin. LIBSVM: a library for support vector machines. Disponível em: `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`. Acessado em: 09/2003, 2004.

[9] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee. Choosing multiple parameters for support vector machines. *Machine Learning*, 46(1-3):131–159, 2002.

[10] C. Cortes and V. N. Vapnik. Support vector networks. *Machine Learning*, 20(3):273–296, 1995.

[11] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, 2000.

[12] K. Duan, S. S. Keerthi, and A. N. Poo. Evaluation of simple performance measures for tuning SVM hyperparameters. *Neurocomputing*, 51:41–59, 2003.

[13] Xiuju Fu, ChongJin Ong, S. Keerthi, Gih Guang Hung, and Liping Goh. Extracting the knowledge embedded in support vector machines. In *Proceedings IEEE International Joint Conference on Neural Networks*, volume 1, page 296, 2004.

[14] S. Haykin. *Neural Networks - A Compreensive Foundation*. Prentice-Hall, New Jersey, 2nd edition, 1999.

[15] M. A. Hearst, B. Schölkopf, S. Dumais, E. Osuna, and J. Platt. Trends and controversies - support vector machines. *IEEE Intelligent Systems*, 13(4):18–28, 1998.

[16] R. Herbrich. *Learning Kernel Classifiers: Theory and Algorithms*. MIT Press, 2001.

[17] C.-W. Hsu and C.-J. Lin. A comparison of methods for multi-class support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425, March 2002.

[18] F. Imbault and K. Lebart. A stochastic optimization approach for parameter tuning of support vector machines. In *Proceedings of the 17th International Conference on Pattern Recognition*, volume 4, pages 597–600, 2004.

[19] T. Joachims. *Learning to classify texts using support vector machines: methods, theory and algorithms*. Kluwer Academic Publishers, 2002.

[20] K. I. Kim, K. Jung, S. H. Park, and H. J. Kim. Support vector machines for texture classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(11):1542–1550, 2002.

[21] A. C. Lorena and A. C. P. L. F. Carvalho. Classificadores de margens largas (*Large Margin Classifiers*). Technical Report 195, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos. Disponível em: `ftp://ftp.icmc.usp.br/pub/BIBLIOTECA/rel_tec/RT_195.ps.zip`, 2003.

[22] A. C. Lorena and A. C. P. L. F. Carvalho. Introdução às máquinas de vetores suporte (*Support Vector Machines*). Technical Report 192, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos. Disponível em: `ftp://ftp.icmc.usp.br/pub/BIBLIOTECA/rel_tec/RT_192.ps.zip`, Abril 2003.

[23] A. C. Lorena and A. C. P. L. F. Carvalho. Revisão de técnicas para geração de classificadores de margens largas multiclasses. Technical Report 221, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos. Disponível em: `ftp://ftp.icmc.usp.br/pub/BIBLIOTECA/rel_tec/RT_221.ps`, Novembro 2003.

[24] A. C. Lorena and A. C. P. L. F. Carvalho. Multiclass SVM design and parameter selection with genetic algorithms. In *IEEE Proceedings of the IX Brazilian Symposium on Neural Networks (SBRN)*, 2006.

[25] Ana Carolina Lorena. *Investigação de estratégias para a geração de máquinas de vetores de suporte multiclasses*. PhD thesis, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos. Disponível em: `http://www.teses.usp.br/teses/disponiveis/55/55134/tde-26052006-111406/`, 2006.

[26] J. Mercer. Functions of positive and negative type and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society*, A 209:415–446, 1909.

[27] T. Mitchell. *Machine Learning*. McGraw Hill, 1997.

[28] K. R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf. An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, 12(2):181–201, Março 2001.

[29] M. C. Monard and J. A. Baranauskas. Conceitos de aprendizado de máquina. In S. O. Rezende, editor, *Sistemas Inteligentes - Fundamentos e Aplicações*, pages 89–114. Editora Manole, 2003.

[30] W. S. Noble. Support vector machine applications in computational biology. In B. Schölkopf, K. Tsuda, and J.-P. Vert, editors, *Kernel Methods in computational biology*, pages 71–92. MIT Press, 2004.

[31] A. Passerini. *Kernel Methods, multiclass classification and applications to computational molecular biology*. PhD thesis, Università Degli Studi di Firenze, 2004.

[32] A. Passerini, M. Pontil, and P. Frasconi. On tuning hyper-parameters of multiclass margin classifiers. In Proceedings of 8th Congress of Associazione Italiana per

l'Intelligenza Artificiale (AIIA). Disponível em: `http://www-dii.ing.unisi.it/aiia2002/paper/APAUT/passerini-aiia02.pdf`, 2002.

[33] M. Pontil and A. Verri. Support vector machines for 3-D object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(6):637–646, 1998.

[34] B. Schölkopf, I. Guyon, and J. Weston. Statistical learning and kernel methods in bioinformatics. In P. Frasconi and R. Shamir, editors, *Artificial Intelligence and Heuristic Methods in Bioinformatics*, pages 1–21. IOS Press, 2003.

[35] J. Shawe-Taylor, P. L. Barlett, R. C. Williamson, and M. Anthony. Structural risk minimization over data-dependent hierarquies. *IEEE Transactions on Information Theory*, 44(5):1926–1940, 1998.

[36] A. Smola and B. Schölkopf. A tutorial on support vector regression. Technical Report NC2-TR-1998-030, NeuroCOLT2, 1998.

[37] A. J. Smola, P. Barlett, B. Schölkopf, and D. Schuurmans. Introduction to large margin classifiers. In A. J. Smola, P. Barlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 1–28. MIT Press, 1999.

[38] A. J. Smola and B. Schölkopf. *Learning with Kernels*. The MIT Press, Cambridge, MA, 2002.

[39] M. C. P. Souto, A. C. Lorena, A. C. B. Delbem, and A. C. P. L. F. Carvalho. *Técnicas de Aprendizado de Máquina para problemas de Biologia Molecular*, pages 103–152. Minicursos de Inteligência Artificial, Jornada de Atualização Científica em Inteligência Artificial, XXIII Congresso da Sociedade Brasileira de Computação, 2003.

[40] B. F. Souza, A. C. P. L. F. Carvalho, R. Calvo, and R. P. Ishii. Multiclass svm model selection using particle swarm optimization. In *Proceedings of 6th HIS*, pages 31–36, 2006.

[41] V. N. Vapnik. *The nature of Statistical learning theory*. Springer-Verlag, New York, 1995.

[42] V. N. Vapnik. *Statistical Learning Theory*. John Wiley and Sons, 1998.

[43] V. N. Vapnik and A. Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16(2):283–305, 1971.

[44] Dexian Zhang, Zhixiao Yang, Yanfeng Fan, and Ziqiang Wang. Extracting symbolic rules from trained support vector machines based on the derivative heuristic information. In *Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007)*, volume 1, pages 592–597, 2007.