Máquina de Vetores Suporte

André Ricardo Gonçalves
andreric [at] dca.fee.unicamp.br
www.dca.fee.unicamp.br/~andreric

Sumário

1 Máquina de Vetores Suporte

Fundamentada na Teoria da Aprendizagem Estatística, a Máquina de Vetores Suporte, do inglês *Support Vectors Machine - SVM*, foi desenvolvida por (VAPNIK, 1995), com o intuito de resolver problemas de classificação de padrões.

Segundo(HAYKIN, 1999) a máquina de vetores suporte é uma outra categoria das redes neurais alimentadas adiante, ou seja, redes cujas saídas dos neurônios de uma camada alimentam os neurônios da camada posterior, não ocorrendo a realimentação.

Esta técnica originalmente desenvolvida para classificação binária, busca a construção de um hiperplano como superfície de decisão, de tal forma que a separação entre exemplos seja máxima. Isso considerando padrões linearmente separáveis[1].

Já para padrões não-linearmente separáveis, busca-se uma função de mapeamento Φ apropriada para tornar o conjunto mapeado linearmente separável.

Devido a sua eficiência em trabalhar com dados de alta dimensionalidade é reportada na literatura como uma técnica altamente robusta, muitas vezes comparada as Redes Neurais (SUNG; MUKKAMALA, 2003) e (DING; DUBCHAK, 2001).

1.1 Teoria da Aprendizado Estatístico

A Teoria do Aprendizado Estatístico visa estabelecer condições matemáticas que permitem escolher um classificador, com bom desempenho, para o conjunto de dados disponíveis para treinamento e teste (LORENA; CARVALHO, 2003a). Em outras palavras esta teoria busca encontrar um bom classificador levando em consideração todo o conjunto de dados, porém se abstendo de casos particulares. Na próxima seção serão apresentados alguns conceitos básicos da TAE.

1.1.1 Conceitos Básicos

O desempenho desejado de um classificador f é que o mesmo obtenha o menor erro durante o treinamento, sendo o erro mensurado pelo número de predições incorretas de f. Sendo assim definimos como risco empírico $R_{emp}(f)$, como sendo a medida de perda entre a resposta desejada e a resposta real. A Eq. (1.1) mostra a definição do risco empírico.

[1] Os padrões devem estar suficientemente separados entre si para assegurar que a superfície de decisão consista de um hiperplano (HAYKIN, 1999).

$$R_{emp}(f) = \frac{1}{n}\sum_{i=1}^{n} c(f(\mathbf{x_i}), y_i)) \tag{1.1}$$

onde $c(\cdot)$ é a função de custo relacionada a previsão $f(\mathbf{x_i})$ com a saída desejada y_i (LORENA; CARVALHO, 2003b), onde um tipo de função de custo é a "perda 0/1" definida pela Eq. (1.2). O processo de busca por uma função f' que represente um menor valor de R_{emp} é denominado de *Minimização do Risco Empírico*.

$$c(f(\mathbf{x_i}), y_i) = \begin{cases} 1 & \text{, se } y_i \text{f}(x_i) < 0 \\ 0 & \text{, caso contrário} \end{cases} \tag{1.2}$$

Sobre a hipótese de que os padrões de treinamento ($\mathbf{x_i}$,y_i) são gerados por uma distribuição de probabilidade $P(x, y)$ em $\mathbb{R}^N$ x {-1,+1} sendo P desconhecida. A probabilidade de classificação incorreta do classificador f é denominada de Risco Funcional, que quantifica a capacidade de generalização, conforme é mostrado pela Eq. (1.3) (SMOLA et al., 1999a) (SMOLA et al., 1999b).

$$R(f) = \int c(f(\mathbf{x_i}), y_i) dP(\mathbf{x_i}, y_i) \tag{1.3}$$

Durante processo de treinamento, $R_{emp}(f)$, pode ser facilmente obtido, ao contrário de $R(f)$, pois em geral a distribuição de probabilidades P é desconhecida (LORENA; CARVALHO, 2003a).

A partir disto, dado um conjunto de dados de treinamento ($\mathbf{x_i}$,y_i) com $\mathbf{x_i} \in \mathbb{R}^N$ e $y_i \in \{-1, +1\}$, $i = \{1, 2, ..., n\}$, sendo $\mathbf{x_i}$ o vetor de entrada e y_i o rótulo da classe.

O objetivo então é estimar uma função f: $\mathbb{R}^N \rightarrow \{-1, +1\}$. Caso nenhuma restrição seja imposta na classe de funções em que se escolhe a estimativa f, pode ocorrer que a função obtenha um bom desempenho no conjunto de treinamento, porém não tendo o mesmo desempenho em padrões desconhecidos, sendo este fenômeno denominado de *"'overfitting"*. Em outras palavras, a minimização apenas do risco empírico $R_{emp}(f)$ não garante uma boa capacidade de generalização, sendo desejado um classificador f^* tal que $R(f^*) = \min_{f \in F} R(f)$, onde F é o conjunto de funções f possíveis.

A figura 1 mostra um exemplo onde uma classe de funções pode ser utilizada para separar padrões linearmente separáveis. É necessário determinar uma função que minimize o R_{emp}, representado na figura como a reta mais escura.

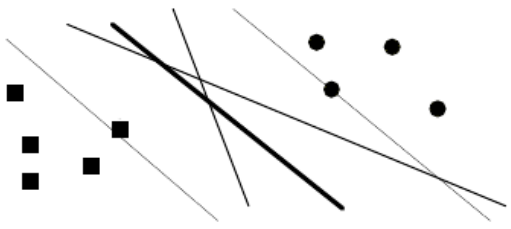

Figura 1: Classe de hiperplanos com um hiperplano ótimo

A TAE provê formas de limitar a classe de funções (hiperplanos), com o intuito de prevenir modelos ruins, ou seja, que levem ao *"overfitting"*, implementando uma função com a capacidade adequada para o conjunto de dados de treinamento (HEARST et al., 1998). Estas limitações são impostas ao risco funcional da função. Os limites utilizam o conceito de dimensão VC.

1.1.2 Dimensão VC

De acordo com (SMOLA et al., 1999b), dado um conjunto de funções sinal G, sua dimensão VC (Vapnik-Chervonenkis) é definida como o tamanho do maior conjunto de pontos que pode ser particionado arbitrariamente pelas funções contidas em G.

Em outras palavras a dimensão VC do conjunto de funções de classificação G é o número máximo de exemplos de treinamento que pode ser aprendido pela máquina sem erro, para todas as rotulações possíveis das funções de classificação (HAYKIN, 1999).

De forma genérica, para funções lineares no $\mathbb{R}^N$ para $n \geq 2$ a dimensão VC é dada pela expressão 1.4.

$$VC(n) = n + 1. \tag{1.4}$$

1.1.3 Conceito de margem e vetores suporte

Sendo $f(x) = (\mathbf{w} \cdot \mathbf{x}) + b$ um hiperplano, podemos definir como *margem* como a menor distância entre os exemplos do conjunto de treinamento e o hiperplano utilizado para separação destas classes (LORENA; CARVALHO, 2003b). A margem determina quão bem duas classes podem ser separadas (SMOLA et al., 1999b).

A margem máxima é obtida com a utilização do hiperplano ótimo, a definição de hiperplano ótimo é apresentada na seção 1.2.1. Podemos definir formalmente a margem conforme 1.1.1.

Definição 1.1.1 *(Margem) A margem ρ de um classificador f é definida por*

$$\rho = \min_i y_i f(\mathbf{x_i}). \tag{1.5}$$

A margem é obtida pela distância entre o hiperplano e os vetores que estão mais próximos a ele, sendo estes vetores denominados de *vetores suporte*. De acordo com (SMOLA et al., 1999b) os vetores suporte são padrões críticos, que sozinhos determinam o hiperplano ótimo, sendo os outros padrões (não-críticos) irrelevantes, podendo ser removidos do conjunto de treinamento sem afetar os resultados. Na figura 2 os vetores suporte são destacados por círculos externos nos padrões.

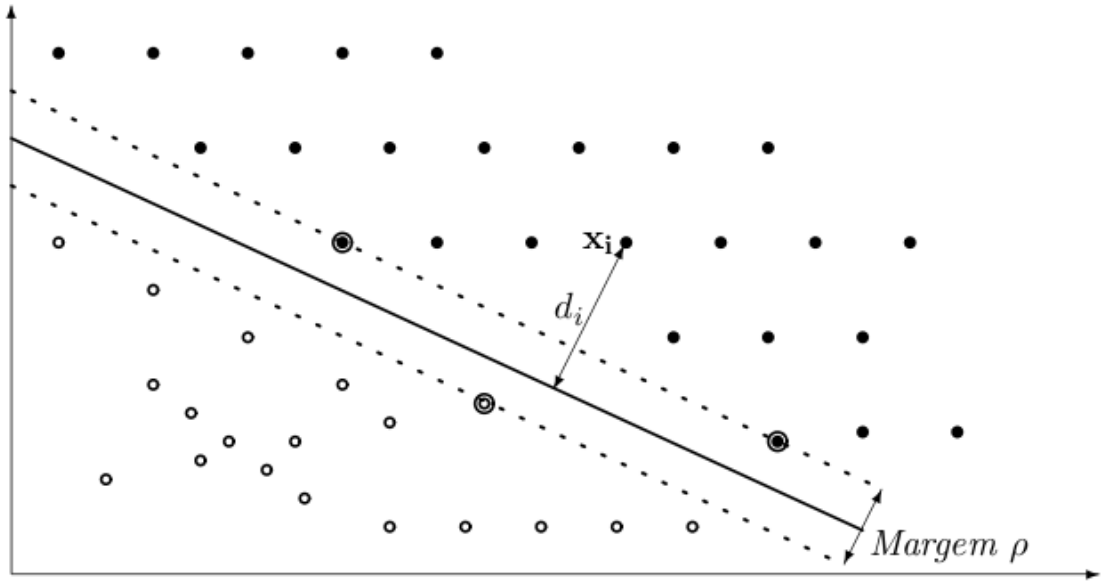

Figura 2: Identificação da margem ρ e dos vetores suporte sobre a linha pontilhada. [Fonte: (LIMA, 2002)]

1.2 Classificação de Padrões Linearmente Separáveis

Uma classificação linear consiste em determinar uma função $f : X \subseteq \mathbb{R}^N \rightarrow \mathbb{R}^N$, que atribui um rótulo (+1) se $f(x) \geqslant 0$ e (-1) caso contrário. Considerando uma função linear, podemos representá-la pela Eq. (1.7).

$$f(x) = \langle \mathbf{w} \cdot \mathbf{x} \rangle + b \tag{1.6}$$

$$= \sum_{i=1}^{n} w_i \mathbf{x_i} + b \tag{1.7}$$

onde $\mathbf{w}$ e $b \in \mathbb{R}^N$ x $\mathbb{R}^N$, são conhecidos como *vetor peso* e *bias*, sendo estes parâmetros responsáveis por controlar a função e a regra de decisão (LIMA, 2002). Os valores de $\mathbf{w}$ e b são obtidos pelo processo de aprendizagem a partir dos dados de entrada.

O vetor peso ($\mathbf{w}$) e o *bias* (b) podem ser interpretados geometricamente sobre um hiperplano. Um hiperplano é um subespaço afim, que divide um espaço em duas partes, correspondendo a dados de duas classes distintas (LIMA, 2002). O vetor peso ($\mathbf{w}$) define uma direção perpendicular ao hiperplano, como mostra a figura 3, e com a variação do *bias* o hiperplano é movido paralelamente a ele mesmo.

Sendo assim um SVM linear busca encontrar um hiperplano que separe perfeitamente os dados de cada classe e cuja margem de separação seja máxima, sendo este hiperplano denominado de *hiperplano ótimo*.

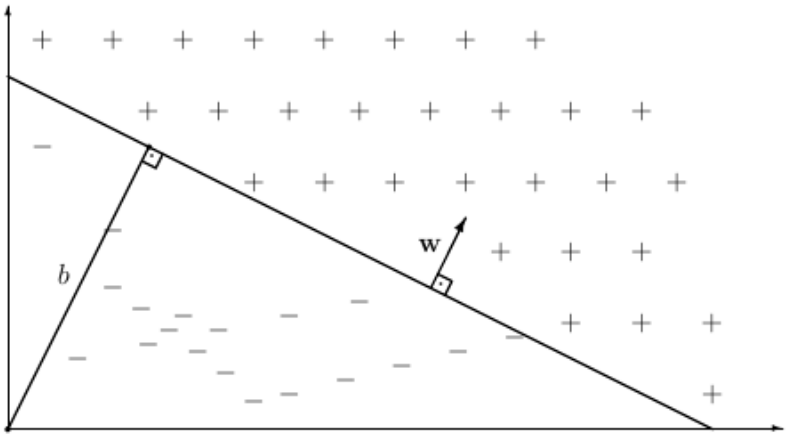

Figura 3: Interpretação geométrica de **w** e **b** sobre um hiperplano. [Fonte: (LIMA, 2002)]

1.2.1 Hiperplano Ótimo

Assumindo-se que o conjunto de treinamento é linearmente separável, o hiperplano ótimo é o hiperplano de separação com maior margem. O hiperplano ótimo é definido como:

$$\langle \mathbf{w} \cdot \mathbf{x} \rangle + b = 0 \tag{1.8}$$

sendo $\mathbf{w}$ e b, o vetor peso e o *bias* respectivamente.

Considerando a restrição imposta pela Eq. (1.9), os classificadores lineares que separam um conjunto de treinamento possuem margem positiva. Ou seja, esta restrição afirma que não há nenhum dado entre $\langle \mathbf{w} \cdot \mathbf{x} \rangle + b = 0$ e $\langle \mathbf{w} \cdot \mathbf{x} \rangle + b = \pm 1$, sendo a margem sempre maior que a distância entre os hiperplanos $\langle \mathbf{w} \cdot \mathbf{x} \rangle + b = 0$ e $|\langle \mathbf{w} \cdot \mathbf{x} \rangle + b = 1|$. Devido a estas suposições as SVMs obtidas são normalmente chamadas de SVMs com margens rígidas (ou largas).

$$\begin{array}{ll} \langle \mathbf{w} \cdot \mathbf{x_i} \rangle + b \geq +1 & \textit{, para } y_i = \textit{+1} \\ \langle \mathbf{w} \cdot \mathbf{x_i} \rangle + b \leq -1 & \textit{, para } y_i = \textit{-1} \end{array} \tag{1.9}$$

Estas equações podem ser combinadas em

$$y_i(\langle \mathbf{w} \cdot \mathbf{x_i} \rangle + b) \geq 1, \;\; i = \{1, 2, ..., n\} \tag{1.10}$$

Seja d_+ (d_-) a distância euclidiana entre os vetores suporte positivos (negativos) e o hiperplano, definimos como *margem* ρ de um hiperplano de separação como sendo a maior margem geométrica entre todos os hiperplanos, podemos representar por $\rho = (d_+ + d_-)$. Denotaremos por $d_i(\mathbf{w}, b; \boldsymbol{x_i})$, como a distância de um dado $\boldsymbol{x_i}$ ao hiperplano (**w**,b), sendo calculado pela Eq. (1.11) (LIMA, 2002)

$$d_i(\mathbf{w}, b; \mathbf{x_i}) = \frac{|\langle \mathbf{w} \cdot \mathbf{x_i} \rangle + b|}{\|\mathbf{w}\|} = \frac{y_i(\langle \mathbf{w} \cdot \mathbf{x_i} \rangle + b)}{\|\mathbf{w}\|} \tag{1.11}$$

levando em consideração a restrição imposta pela Eq. (1.10), podemos escrever

$$d_i(\mathbf{w}, b; \mathbf{x_i}) \geq \frac{1}{\|\mathbf{w}\|}. \tag{1.12}$$

Com isso podemos identificar $\frac{1}{\|\mathbf{w}\|}$ como o limite inferior da distância entre os vetores suporte $\boldsymbol{x_i}$ e o hiperplano de separação ($\mathbf{w}$,b), as distâncias d_+ e d_- ficam

$$d_+ = d_- = \frac{1}{\|\mathbf{w}\|}. \tag{1.13}$$

Como suposto anteriormente que a margem é sempre maior que a última instância, a minimização de $\|\mathbf{w}\|$ leva a maximização da margem. A partir disto podemos definir a margem ρ através da Eq. (1.14)

$$\rho = (d_+ + d_-) = \frac{\mathbf{2}}{\|\mathbf{w}\|}. \tag{1.14}$$

A figura 4 mostra a distância entre hiperplanos e os vetores suporte.

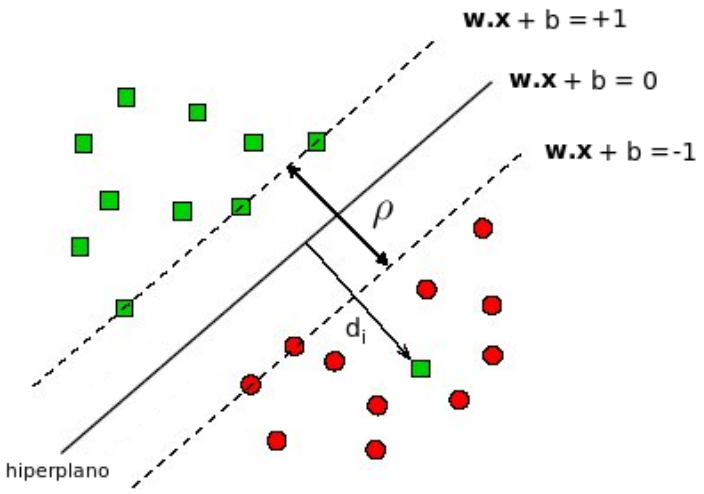

Figura 4: Distância entre hiperplanos e vetores suporte.

O hiperplano ótimo é dado pela minimização da norma $\|\mathbf{w}\|$, considerando a restrição da Eq. (1.10). Formalmente podemos reescrever (LORENA; CARVALHO, 2003b)

$$\begin{array}{ll} \textbf{Problema} & P1 \\ \textbf{Minimizar} & \|\mathbf{w}\| \\ \textbf{Sujeito a} & y_i(\langle \mathbf{w} \cdot \mathbf{x_i} \rangle + b) \geq 1, \text{ para i=\{1,2,...,n\}} \end{array} \tag{1.15}$$

Como pode ser observado, o problema acima trata-se de um problema clássico de otimização, denominado *programação quadrática* (HEARST et al., 1998). Este problema pode ser resolvido com o método clássico de *multiplicadores de Lagrange* (LIMA, 2002).

Utilizando a teoria dos multiplicadores de Lagrange, podemos representar o problema 1.15, como:

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i (y_i(\langle \mathbf{w} \cdot \mathbf{x_i} \rangle + b) - 1) \tag{1.16}$$

onde α_i são os *multiplicadores de Lagrange*. O problema então passa a ser a minimização de $L(\mathbf{w}, b, \alpha)$, em relação a $\mathbf{w}$ e b e a maximização dos α_i (LORENA; CARVALHO, 2003b). O método dos multiplicadores encontra os pontos ótimos igualando as derivadas parciais a zero. Sendo assim os pontos ótimos da Eq. (1.16) são obtidos por meio da resolução das igualdades:

$$\frac{\partial L}{\partial b} = 0 \tag{1.17}$$

$$\frac{\partial L}{\partial \mathbf{w}} = 0 \tag{1.18}$$

sendo

$$\frac{\partial L}{\partial \mathbf{w}} = \left(\frac{\partial L}{\partial w_1}, \frac{\partial L}{\partial w_2}, ..., \frac{\partial L}{\partial w_n} \right). \tag{1.19}$$

A partir das Equações 1.17 e 1.18 obtém-se

$$\sum_{i=1}^{n} \alpha_i y_i = 0 \tag{1.20}$$

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x_i}. \tag{1.21}$$

Substituindo as equações 1.20 e 1.21 no lado direito da Eq. (1.16), chegamos ao seguinte problema de otimização:

$$\begin{array}{ll} \textbf{Problema} & \textbf{P2} \\ \textbf{Maximizar} & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \langle \mathbf{x_i} \cdot x_j \rangle \\ \textbf{Sujeito a} & \begin{cases} \alpha_i \geq 0, \ \ i = \{1, ..., n\} \\ \sum_{i=1}^{n} \alpha_i y_i = 0 \end{cases} \end{array} \tag{1.22}$$

O objetivo então é determinar os valores ótimos de (**w**,b), que representaremos por ($\mathbf{w}^*$,b^*). Através da Eq. (1.21) podemos calcular $\mathbf{w}^*$ e b^* como segue:

$$\mathbf{w}^* = \sum_{i=1}^{n} \alpha_i^* y_i \mathbf{x_i} \tag{1.23}$$

o valor de b^* pode ser obtido utilizando as equações de Karush-Kuhn-Tucker (KKT)

$$\alpha_i^* (y_i (\langle \mathbf{w}^* \cdot \mathbf{x_i} \rangle + b^*) - 1) = 0, \ \ i = \{1, ..., n\} \tag{1.24}$$

É possível observar que os α_i^*'s assumem valores positivos para os exemplos de treinamento que estão a uma distância do hiperplano ótimo igual a largura da margem, ou seja, os vetores suporte. Para o restante dos exemplos, α_i^* é nulo. Portanto, conclui-se que o hiperplano ótimo é obtido unicamente pelos vetores suporte. Logo, se apenas o subconjunto de dados de treinamento formado pelos vetores suporte fossem utilizados, o hiperplano obtido seria o mesmo gerado pela utilização de todo o conjunto (LORENA; CARVALHO, 2003b).

Dado um vetor suporte $\mathbf{x_j}$, podemos obter b^* por meio da condição de KKT

$$b^* = y_j - \langle \mathbf{w}^* \cdot \mathbf{x_j} \rangle. \tag{1.25}$$

Com os valores dos parâmetros $\mathbf{w}^*$ e b^* calculados, podemos classificar de um novo padrão **z** apenas calculando

$$sgn(\langle \mathbf{w}^* \cdot \mathbf{z} \rangle) + b^*) \tag{1.26}$$

a classificação é dada apenas pelo cálculo do produto interno entre o novo padrão e todos os vetores suporte.

1.3 Classificação de Padrões Não-Linearmente Separáveis

As SVMs apresentadas até agora, trabalham apenas quando os padrões são linearmente separáveis. Em problemas reais esta característica é dificilmente encontrada, sendo a maioria deles complexos e não-lineares. Para estender a SVM linear a resolução de problemas não-lineares foram introduzidas funções reais, que mapeiam o conjunto de treinamento em um espaço linearmente separável, o *espaço de características*.

Um conjunto de dados é dito ser não-linearmente separável, caso não seja possível separar os dados com um hiperplano. A figura 5 mostra um conjunto linearmente e outro não-linearmente separável.

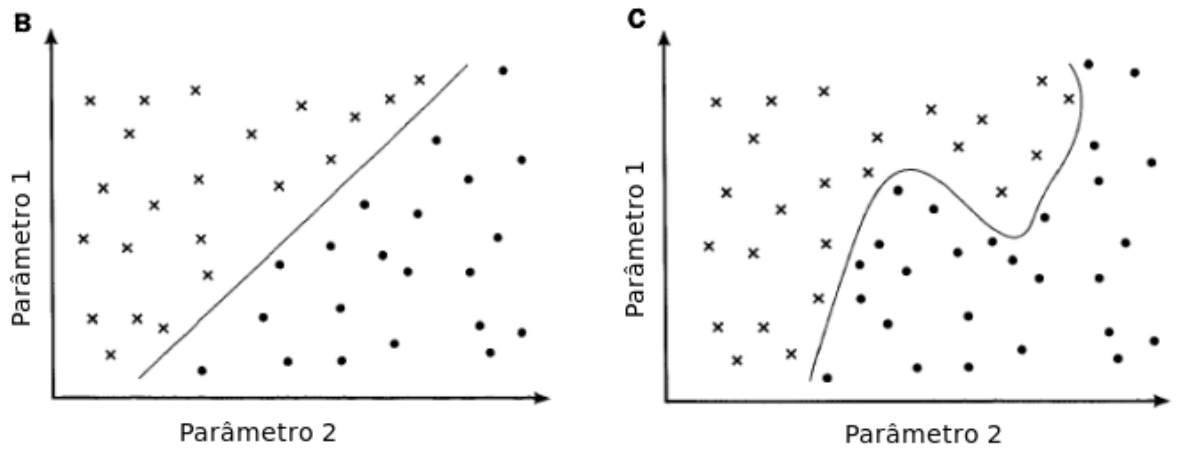

Figura 5: Exemplos de padrões linearmente e não-linearmente separável respectivamente.

O teorema de Cover afirma que um problema não-linear tem maior probabilidade de ser linearmente separável, em um espaço de mais alta dimensionalidade (SMOLA et al., 1999b). A partir disso, a SVM não-linear realiza uma mudança de dimensionalidade, por meio das funções *Kernel*, caindo então em um problema de classificação linear, podendo fazer uso do hiperplano ótimo.

1.3.1 Mapeamento no espaço de características

Seja o conjunto de entrada S representado pelos pares $\{(\mathbf{x_1}, y_1), ..., (\mathbf{x_n}, y_n)\}$, com $y_i, i = 1, 2, .., n$ o rótulo de cada padrão i, o conjunto de dados de treinamento.

O espaço de característica é um espaço de mais alta dimensionalidade no qual serão mapeados o conjunto de entrada S, por meio de uma função Φ, a fim de obter um novo conjunto de dados S' linearmente separável, representado por $\{(\mathbf{\Phi}(\mathbf{x_1}), y_1), ..., (\mathbf{\Phi}(\mathbf{x_n}), y_n)\}$. A figura 6 mostra o processo de mapeamento.

Muitas vezes é interessante diminuir o número de características do conjunto de entrada, em um subconjunto menor com apenas os atributos que contém as informações essenciais, este procedimento é denominado de *redução da dimensionalidade* (LIMA, 2002).

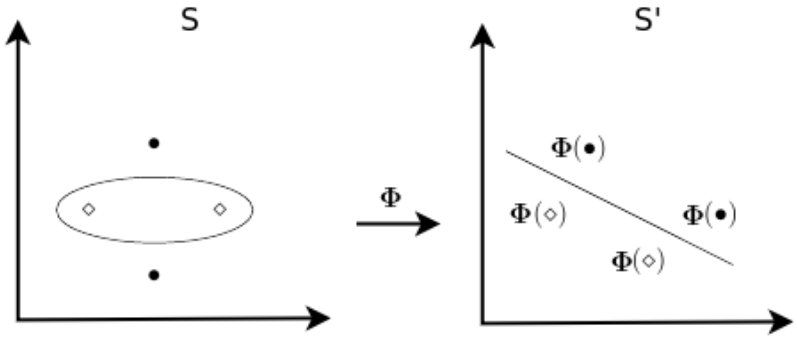

Figura 6: Mapeamento do conjunto de treinamento para o espaço de característica

Esta modificação de conjunto de atributos apresenta ganho no sistema, diminuindo o tempo computacional e aumentando a acurácia, pois geralmente estes parâmetros degradam o sistema com aumento do número de características, sendo este fenômeno referenciado de *maldição da dimensionalidade* (LIMA, 2002).

1.3.2 SVMs lineares no espaço de características

Com os dados de treinamento mapeados para o espaço de características, a única modificação a ser feita na SVM linear descrita na Eq. (1.22) é a utilização dos valores mapeados $\Phi(\mathbf{x})$ no lugar de $\mathbf{x}$, sendo assim o problema consiste em

$$\begin{array}{ll} \textbf{Problema} & \textbf{P3} \\ \textbf{Maximizar} & \displaystyle\sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \langle \mathbf{\Phi}(\mathbf{x_i})\cdot\mathbf{\Phi}(\mathbf{x_j})\rangle \\ \textbf{Sujeito a} & \begin{cases} \alpha_i \geq 0, \;\; i = \{1,...,n\} \\ \displaystyle\sum_{i=1}^{n}\alpha_i y_i = 0 \end{cases} \end{array} \tag{1.27}$$

valendo para classificação não-linear as mesmas considerações do Karush-Kuhn-Tucker, descrito no classificador linear. O hiperplano de decisão ótimo, agora é definido como

$$(\mathbf{w}\cdot\Phi(\mathbf{x})) + b = 0 \tag{1.28}$$

O problema de classificação não-linear de um novo padrão **z** é solucionado calculando

$$sgn(\langle \mathbf{w}^*\cdot\mathbf{\Phi}(\mathbf{z})\rangle + b^*) \tag{1.29}$$

1.3.3 Funções "Kernel"

Uma função *Kernel* recebe dois dados de entrada $\mathbf{x_i}$ e $\mathbf{x_j}$ e calcula o produto interno destes dados no espaço de características

$$\mathcal{K}(\mathbf{x_i}, \mathbf{x_j}) = \langle \Phi(\mathbf{x_i}) \cdot \Phi(\mathbf{x_j}) \rangle \tag{1.30}$$

sendo necessário que a função $\Phi(\cdot)$ pertença a um domínio, onde seja possível o cálculo do produto interno. Funções estas que satisfazem as condições do *Teorema de Mercer*.

Uma função é dita ser uma função *Kernel* se ela satisfaz as condições estabelecidas pelo *Teorema de Mercer*.

Teorema 1.3.1 *(Teorema de Mercer) Uma função é dita ser uma função Kernel, se a matriz K é positivamente definida, onde K é obtida por*

$$K = K_{ij} = \mathcal{K}(\mathbf{x_i}, \mathbf{x_j}). \tag{1.31}$$

Uma matriz é positivamente definida, se seus autovalores são maiores que zero. As funções *Kernel* que satisfazem as condições do Teorema de Mercer, são chamadas de *Kernels de Mercer* (SMOLA et al., 1999b).

Algumas das funções *Kernels* mais utilizadas estão descritas na tabela 1.

Tipo de Kernel	**Função $\mathcal{K}(\mathbf{x_i}, \mathbf{x_j})$**	**Tipo do Classificador**
Polinomial	$(\langle \mathbf{x_i} \cdot \mathbf{x_j} \rangle + 1)^p$	Máquina de aprendizagem polinomial
Gaussiano (ou RBF)	$\exp\left(-\frac{\|\mathbf{x_i} - \mathbf{x_j}\|^2}{2\sigma^2}\right)$	Rede RBF
Sigmoidal	$\tanh(\beta_0 \langle \mathbf{x_i} \cdot \mathbf{x_j} \rangle) + \beta_1$	Perceptron de duas camadas

Tabela 1: Resumo dos *Kernels* mais populares

Algumas considerações sobre as funções *Kernels* descritas na tabela 1, foram apresentadas por (HAYKIN, 1999):

- Na Máquina de aprendizagem polinomial a potência p é especificada *a priori* pelo usuário;
- Na Rede RBF a amplitude σ^2, comum a todos os *Kernels*, é especificada pelo usuário;
- No Perceptron de duas camadas o teorema de Mercer é satisfeito apenas para alguns valores de β_0 e β_1.

Algumas escolhas devem ser feitas para obtenção de uma SVM, como a função *Kernel*, os parâmetros da função, além da definição do algoritmo para determinação do hiperplano ótimo (LORENA; CARVALHO, 2003b).

1.4 Classificação Multiclasses

As máquinas de vetores suporte foram propostas inicialmente como ferramenta de classificação binária. Porém muitos dos problemas reais possuem características multiclasses. Para que fosse possível a utilização da SVM neste tipo de aplicação, foram propostos alguns procedimentos ara estender a SVM binária.

Em termos formais, em um sistema multiclasses o conjunto de treinamento é composto por pares $(\mathbf{x_i}, y_i)$, tal que $y_i \in \{1, ..., k\}$, com $k > 2$, onde k é o número de classes.

As principais abordagens utilizam como base a decomposição de um problema multiclasse com k classes, $k > 2$, em k problemas binários, destacando os métodos: decomposição "Um-Contra-Todos" e decomposição "Todos-Contra-Todos". Estes métodos são apresentados na seção 1.4.1 e 1.4.2 respectivamente.

1.4.1 Decomposição "Um-Contra-Todos"

O método "Um-Contra-Todos" (UCT) baseia-se na construção de k classificadores binários, sendo k o número de classes. Cada classificador f_i é responsável por classificar uma classe i das demais.

Dado um novo padrão $\mathbf{x}$, a classe a qual este novo padrão pertence é a classe representada pelo classificador que obteve o valor máximo entre o k classificadores. Formalmente é definido por

$$f(x) = \arg \max_{1 \leq i \leq k} (f_i(\mathbf{x})). \tag{1.32}$$

1.4.2 Decomposição "Todos-Contra-Todos"

Proposto por (FRIEDMAN, 1996) a abordagem "Todos-Contra-Todos" (TCT) consiste em comparar as classes duas a duas, sendo necessárias $k \cdot (k-1)/2$ SVMs, onde k é o número de classes. Para decidir a qual classe pertence um novo padrão $\mathbf{x}$, utiliza-se um esquema de votação por maioria, onde cada uma das SVMs fornecem um resultado. A solução final é dada pela classe com maior quantidade de votos.

Com um grande número de SVMs executadas, um alto tempo de processamento é necessário, comparado ao método de decomposição UCT. Por exemplo, sendo $k = 10$, no método UCT são necessárias 10 SVMs enquanto seriam necessárias 45 SVMs no método TCT. O método TCT não provê limites de generalização (PLATT; CRISTIANINI; SHAWE-TAYLOR, 2000).

Com o intuito de solucionar estes problemas, (PLATT; CRISTIANINI; SHAWE-

TAYLOR, 2000) propôs um modelo, denominado DAGSVM, que utiliza grafos direcionados acíclicos (DAG), sendo cada nó do grafo encarregado de classificar entre duas classes distintas. A figura 7 mostra o grafo de SVMs que representa a arquitetura do modelo DAGSVM.

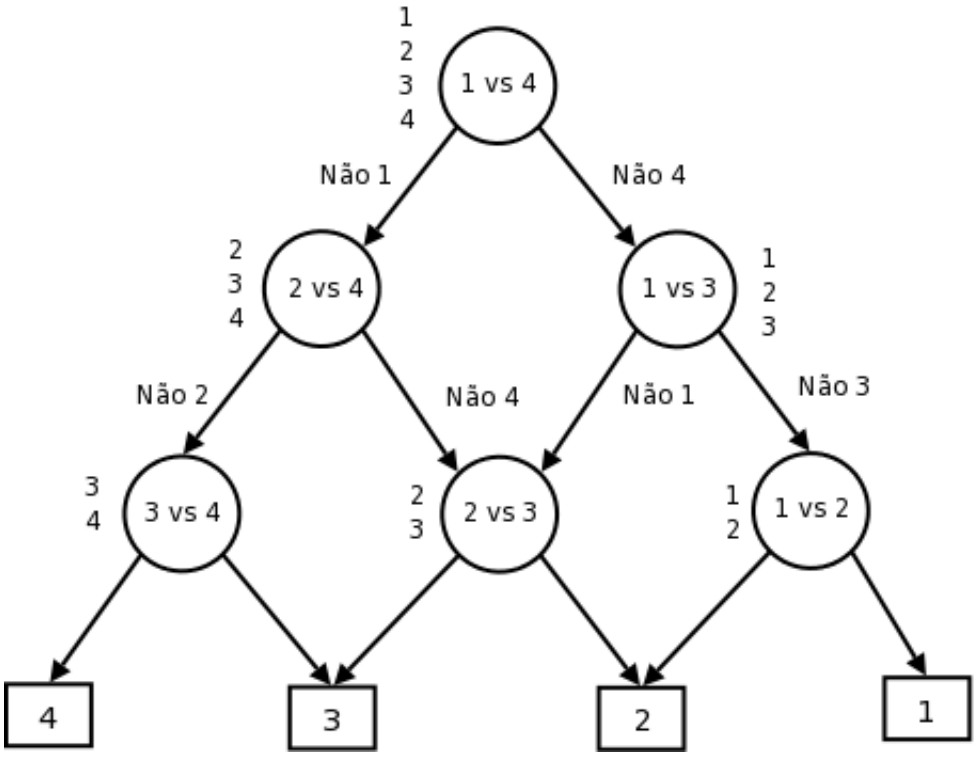

Figura 7: Arquitetura do método DAGSVM.

Segundo (PLATT; CRISTIANINI; SHAWE-TAYLOR, 2000) o algoritmo trabalho equivalentemente a operação sobre um lista, onde cada nó da árvore elimina uma classe da lista.

Inicialmente todas as classes estão na lista, sendo que cada nó do grafo decide, entre a primeira e última classe da lista, a classe com maior chance do padrão a pertencer. A classe não escolhida é eliminada da lista e continua o processo com a primeira e última classe da nova lista. O algoritmo termina quando há apenas uma classe na lista. Sendo assim o número de avaliações requeridas pelo DAGSVM é $k-1$, conseqüentemente possui um tempo de processamento bem menor do que o método TCT originalmente proposto.

(PLATT; CRISTIANINI; SHAWE-TAYLOR, 2000) mostrou que o limite do erro de generalização é determinado pela margem máxima da SVM de cada nó do grafo.

1.5 Aplicações

A proposta inicial da SVM foi a aplicação em problemas de classificação, sendo este o principal enfoque desta técnica. Pode ainda ser aplicado em problemas de regressão (HAYKIN, 1999).

Devido ao fato de sua eficiência em problemas de alta dimensionalidade, a SVM vem obtendo grande sucesso em aplicações de visão computacional, que busca ex-

trair informações a partir de imagens. Exemplos de aplicação em visão computacional são: classificação de impressões digitais (LIMA, 2002). Também são aplicadas em bioinformática (MA; HUANG, 2008) e classificação textual (TONG; KOLLER, 2000). A SVM também pode ser aplicada em regressão não-linear, como em (SUN; SUN, 2003).

Esta técnica tem seus resultados muitas vezes comparados com resultados de outras técnicas como redes neurais.

1.6 Conclusão

As Máquinas de Vetor Suporte destacam-se pela forte fundamentação teórica existente, possuindo como base a teoria da aprendizagem estatística, sendo esta característica um diferencial sobre outras técnicas como redes neurais, que não possui um modelo teórico.

A capacidade em trabalhar com padrões de alta dimensionalidade é outra característica interessante desta técnica, sendo ideal para aplicação em problemas de visão computacional, como reconhecimento de padrões e filtragem.

Mesmo com características atrativas, algumas ressalvas devem ser feitas, como descreve (LORENA; CARVALHO, 2003b):

- Velocidade de classificação pode ser menor do que outras técnicas como Redes Neurais;
- Alta complexidade computacional na busca de soluções, agravando ainda mais quando um grande número de dados estão disponíveis para treinamento;
- Conhecimento adquirido não é facilmente interpretável.

Diversos estudos foram realizados com o intuito de minimizar estas deficiências, o que juntamente com a robustez desta técnica, faz da SVM uma das técnicas mais exploradas atualmente.

Referências

DING, C. H.; DUBCHAK, I. Multi-class protein fold recognition using support vector machines and neural networks. *Bioinformatics*, v. 17, n. 4, p. 349–358, 2001.

FRIEDMAN, J. H. *Another approach to polychotomous classification*. [S.l.], 1996. Disponível em: <http://www-stat.stanford.edu/~jhf/ftp/poly.ps.Z>.

HAYKIN, S. *Redes Neurais, Princípios e prática*. 2. ed. [S.l.]: Bookman, 1999.

HEARST, M. A. et al. Support vector machines. *IEEE Intelligent Systems*, IEEE Computer Society, Los Alamitos, CA, USA, v. 13, n. 4, p. 18–28, 1998. ISSN 1094-7167.

LIMA, A. R. G. *Máquinas de Vetores Suporte na Classificação de Impressões Digitais*. Dissertação (Mestrado) — Universidade Federal do Ceará, Fortaleza, Ceará, 2002.

LORENA, A. C.; CARVALHO, A. C. P. L. de. *Introdução aos Classificadores de Margens Largas*. São Carlos - SP, Maio 2003.

LORENA, A. C.; CARVALHO, A. C. P. L. de. *Introdução às Máquinas de Vetores Suporte*. São Carlos - SP, Abril 2003.

MA, S.; HUANG, J. Penalized feature selection and classification in bioinformatics. *Brief Bioinform*, v. 9, n. 5, p. 392–403, September 2008. Disponível em: <http://dx.doi.org/10-.1093/bib/bbn027>.

PLATT, J.; CRISTIANINI, N.; SHAWE-TAYLOR, J. Large margin dags for multiclass classification. In: SOLLA, S.; LEEN, T.; MUELLER, K.-R. (Ed.). *Advances in Neural Information Processing Systems 12*. [S.l.: s.n.], 2000. p. 547–553.

SMOLA, A. J. et al. *Advances in Large Margin Classifiers*. [S.l.]: Morgan-Kauffman, 1999.

SMOLA, A. J. et al. Introduction to large margin classifiers. In: _____. [S.l.]: Morgan-Kauffman, 1999. cap. 1, p. 1–28.

SUN, Z.; SUN, Y. Fuzzy support vector machine for regression estimation. In: *Systems, Man and Cybernetics, 2003. IEEE International Conference on*. [S.l.: s.n.], 2003. p. 3336–3341.

SUNG, A. H.; MUKKAMALA, S. Identifying important features for intrusion detection using support vector machines and neural networks. *Applications and the Internet, 2003. Proceedings. 2003 Symposium on*, p. 209–216, January 2003.

TONG, S.; KOLLER, D. Support vector machine active learning with applications to text classification. In: *Proceedings of ICML-00, 17th International Conference on Machine Learning*. Stanford, US: Morgan Kaufmann Publishers, San Francisco, US, 2000. p. 999–1006. Disponível em: <http://citeseer.ist.psu.edu/tong00support% -.html>.

VAPNIK, V. N. *The nature of statistical learning theory*. New York, NY, USA: Springer-Verlag New York, Inc., 1995. ISBN 0387945598. Disponível em: <http://portal.acm.org/citation-.cfm?id=211359>.