2
Máquinas de Vetor Suporte

2.1.
Introdução

Os fundamentos das Máquinas de Vetor Suporte (SVM) foram desenvolvidos por Vapnik e colaboradores [2], [3], [4]. A formulação por ele apresentada se baseia no princípio de Minimização do Risco Estrutural (SRM), que tem um desempenho de generalização superior ao tradicional princípio de Minimização do Risco Empírico (ERM) [7], empregado em redes neurais convencionais.

O princípio SRM (ver apêndice 1) é baseado no fato de que a taxa de erro nos dados de teste (taxa de erro de generalização) é limitada pela soma da taxa de erro de treinamento e por um termo que depende da dimensão de Vapnik-Chervonenkis (dimensão VC) [2], [5], [8], [23]. A dimensão VC não tem conexão com a noção geométrica de dimensão e desempenha um papel central na teoria do aprendizado estatístico, como será mostrado no apêndice 1. É uma medida da capacidade ou poder de expressão de um conjunto de funções. No caso de padrões separáveis, a máquina de vetor suporte tem valor 0 para a taxa de erro de treinamento e minimiza a dimensão VC. Conseqüentemente, a máquina de vetor suporte pode ter um bom desempenho de generalização em problemas de classificação de padrões.

SVMs foram desenvolvidas para resolver problemas de classificação, tendo sido utilizadas com sucesso em aplicações de reconhecimento de padrões [24], tais como categorização de textos [25], categorização de SPAM [26], reconhecimento de caracteres manuscritos [3], [4], reconhecimento de textura [27], análise de expressões de genes [28], reconhecimento de objetos em 3 dimensões [29], etc. Recentemente as máquinas de vetor suporte foram estendidas ao domínio de problemas de regressão [1], [5], [7], [8], [23], [30].

Basicamente o funcionamento de uma SVM pode ser descrito da seguinte forma: dadas duas classes e um conjunto de pontos que pertencem a essas classes, uma SVM determina o hiperplano que separa os pontos de forma a colocar o maior número de pontos da mesma classe do mesmo lado, enquanto

maximiza a distância de cada classe a esse hiperplano. A distância de uma classe a um hiperplano é a menor distância entre ele e os pontos dessa classe e é chamada de margem de separação. O hiperplano gerado pela SVM é determinado por um subconjunto dos pontos das duas classes, chamado vetores suporte.

Os conceitos acima serão mais detalhados nas seções seguintes.

2.2. Classificação Binária

Uma máquina de vetor suporte constrói um classificador binário a partir de um conjunto de padrões, chamados de exemplos de treinamento, em que a classificação é conhecida.

Seja (x_i,y_i), com $x_i \in \mathbb{R}^n$ e $y_i \in \{-1,1\}$, i = 1, ..., N o conjunto de exemplos de treinamento, onde x_i é o vetor de entrada e y_i é a classificação desejada.

O objetivo é estimar uma função f: $\mathbb{R}^n \rightarrow \{\pm 1\}$, usando os exemplos de treinamento, que classifique corretamente os exemplos de teste (x,y), não utilizados no treinamento. Se nenhuma restrição for imposta na classe de funções em que se escolhe a estimativa f, mesmo uma função que tenha um bom desempenho nos pontos de treinamento pode não generalizar bem nos exemplos não utilizados no treinamento. Assim, minimizar somente o erro de treinamento não implica em um erro de teste pequeno. Esse fenômeno, chamado de "overfitting", ocorre com mais freqüência quando a dimensionalidade do espaço de entrada aumenta.

A teoria do aprendizado estatístico mostra que é preciso restringir a classe de funções em que f é escolhida. Essa teoria fornece limites para o erro de teste. A minimização desses limites, que dependem do risco empírico e da capacidade da classe de funções especificada, leva ao princípio de minimização do risco estrutural (SRM). O conceito de capacidade mais conhecido da teoria do aprendizado estatístico é a dimensão VC, definida como o maior número de pontos que podem ser separados de todas as maneiras possíveis, usando-se funções da classe escolhida [2], [3] (ver apêndice 1).

A máquina de vetor suporte constrói um conjunto de hiperplanos cujos limites da dimensão VC possam ser computados e usa, então, o princípio de minimização do risco estrutural para identificar o hiperplano ótimo que maximize a margem dos exemplos mais próximos [2], [3]. Isso é equivalente a minimizar o limite da dimensão VC.

As primeiras SVMs não eram capazes de lidar com erros de classificação; somente padrões de treinamento linearmente separáveis no espaço de características podiam ser usados. SVMs baseadas nesse algoritmo de treinamento são conhecidas como SVMs com "margem maximal". Como exemplo, considere a Figura 1. Existem vários hiperplanos que separam os dados, mas há somente um que maximiza a margem (em negrito), ou seja, que

maximiza a distância entre os pontos mais próximos de cada classe e o hiperplano de separação.

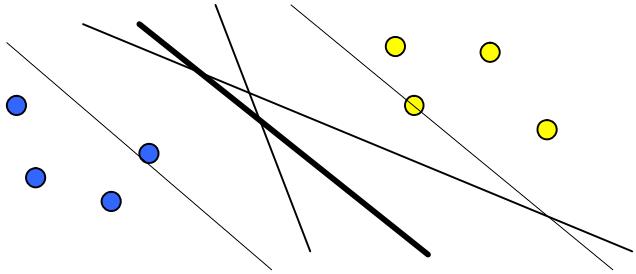

Figura 1 - Hiperplano de Margem Maximal

Com a adoção das "variáveis soltas" (ou “variáveis de folga”) no processo de treinamento, padrões de treinamento não linearmente separáveis no espaço de características puderam ser tratados. Conforme será descrito na próxima seção, a utilização de variáveis soltas diminui o erro de classificação e permite um equilíbrio entre a topologia da SVM (risco estrutural) e o erro de treinamento (risco empírico) [31]. Como resultado, hiperplanos de separação com margem maximal e erros de classificação mínimos são obtidos com "margem suave".

2.2.1.
SVMs com margens maximais

Para padrões linearmente separáveis, a solução do problema de treinamento de SVMs consiste em achar um hiperplano que separe perfeitamente os pontos de cada classe e cuja margem de separação seja máxima. Esse hiperplano é chamado de hiperplano ótimo e é definido por:

$$(w.x) + b = 0, \quad (1)$$

onde $w \in \mathbb{R}^n$ é o vetor de pesos e o escalar b é o *bias*. Um vetor x_j da classe y_j, $y_j \in \{-1,1\}$, deve satisfazer a equação

$$y_j((w.x_j) + b) \geq 1 \text{ para } j = 1,..., N. \quad (2)$$

A distância euclidiana entre esse hiperplano e os pontos que estão sobre a margem, isto é, $y_j((w.x_j) + b) = 1$, é determinada pela equação abaixo:

$$\frac{y_j\left((w.x_j)+b\right)}{\|w\|} = \frac{1}{\|w\|}. \quad (3)$$

Assim, minimizar $||w||$ é equivalente a maximizar a margem do hiperplano de separação.

A construção de SVMs com margem maximal pode ser descrita como:

Dado um conjunto de treinamento $\{(x_1, y_1), ..., (x_N, y_N)\}$, determine os valores ótimos para o vetor de pesos w e o *bias* b para que a seguinte restrição

$$y_j(w.x_j + b) \geq 1, \text{ para } j = 1,..., N, \quad (4)$$

seja satisfeita quando a função custo

$$\Phi(w) = \frac{1}{2}(w.w) \quad (5)$$

for minimizada.

A formulação acima é chamada de problema primal. A função custo é convexa e todas as restrições são lineares. Usando-se a teoria dos multiplicadores de Lagrange [1], [5], [8], esse problema pode ser representado como:

$$J(w,b,a) = \frac{1}{2}(w.w) - \sum_{i=1}^{N} a_i \left[y_i\left((w.x_i)+b\right)-1\right], \quad (6)$$

onde a_i são chamados multiplicadores de Lagrange. A solução para o problema de otimização é determinada minimizando-se $J(w,b,a)$ em relação a w e b e maximizando-se $J(w,b,a)$ em relação a a. Diferenciando-se essa equação em

[1] $(w.x)$ é o produto interno dos vetores w e x.

relação às componentes w_j de w, j = 1,...,n, e b e igualando-se as derivadas resultantes a zero, tem-se:

$$\left(\frac{\partial J}{\partial w_1},...,\frac{\partial J}{\partial w_n}\right) = w - \sum_{i=1}^{N}\alpha_i y_i x_i = 0 \Rightarrow w = \sum_{i=1}^{N} a_i y_i x_i \ , \tag{7}$$

$$\frac{\partial J}{\partial b} = \sum_{i=1}^{N} a_i y_i = 0 . \tag{8}$$

Substituindo-se as expressões obtidas acima na expressão do Lagrangeano:

$$J(w,b,a) = \frac{1}{2}(w.w) - \sum_{i=1}^{N} a_i \left[y_i\left((w.x_i)+b\right)-1\right]$$

$$= \frac{1}{2}\left(\sum_{i=1}^{N} a_i y_i x_i\right)\left(\sum_{j=1}^{N} a_j y_j x_j\right) - \sum_{i=1}^{N} a_i \left[y_i\left(\left(\left(\sum_{j=1}^{N} a_j y_j x_j\right)x_i\right)+b\right)-1\right]$$

$$= \frac{1}{2}\sum_{i,j=1}^{N} a_i a_j y_i y_j \left(x_i.x_j\right) - \sum_{i,j=1}^{N} a_i a_j y_i y_j \left(x_i.x_j\right) - b\sum_{i=1}^{N} a_i y_i + \sum_{i=1}^{N} a_i \ .$$

Utilizando-se, agora, a equação 8, chega-se a

$$J(w,b,a) = -\frac{1}{2}\sum_{i,j=1}^{N} a_i a_j y_i y_j \left(x_i.x_i\right) + \sum_{i=1}^{N} a_i \ . \tag{9}$$

Desse modo, o problema dual a ser resolvido é, considerando um conjunto de treinamento $\{(x_1, y_1),..., (x_N, y_N)\}$, determinar os multiplicadores de Lagrange ótimos a_i para que J seja maximizado, e que satisfaçam a

$$\sum_{i=1}^{N} a_i y_i = 0 \text{ e } a_i \geq 0. \tag{10}$$

As condições de Karush-Kuhn-Tucker [1], [5], [8], que são condições necessárias e suficientes, estabelecem que as soluções ótimas a^*,w^* e b^* devem satisfazer à seguinte igualdade:

$$a^*[y_j((w^*.x_j) + b^*) - 1] = 0, \text{ para j} = 1,..., N. \tag{11}$$

Assim, pela equação, vê-se que os pontos em que $y_j((w.x_j) + b) \neq 1$ devem necessariamente ter $a = 0$. Só os pontos com margem 1 podem ter os correspondentes $a \neq 0$. Esses pontos são chamados de vetores suporte. Na expressão de w somente esses pontos estão envolvidos.

2.2.2.
SVMs com margens suaves

Nesta seção, considera-se o caso mais difícil em que os padrões de treinamento não são linearmente separáveis. Para esse conjunto de padrões não é possível construir um hiperplano de separação que classifique corretamente todos os pontos.

Como já mencionado, para a formulação da SVM da seção anterior permitir erros de classificação, introduzem-se variáveis soltas. Essas variáveis permitem que a equação 4 seja violada. Assim, um vetor x_j é classificado corretamente como da classe y_j, $y_j \in \{-1,1\}$, quando a seguinte expressão é verdadeira

$$y_j((w.x_j) + b) + x_j \geq 1 \text{ para j = 1,..., N,} \quad (12)$$

onde w é o vetor de pesos em $\mathbb{R}^n$, o escalar b é o *bias* e x_j são variáveis soltas não negativas associadas a cada vetor de treinamento x_j. Se $0 \leq x_j \leq 1$, então o ponto x_j se encontra do lado correto do hiperplano de separação, ou seja, o padrão é classificado corretamente. No caso de $x_j > 1$, o ponto x_j se encontra do lado errado do hiperplano de separação, já que, nesse caso, $y_j((w.x_j) + b) < 0$. Dessa forma, os pontos x_j em que x_j é maior do que 1 são pontos classificados incorretamente. Assim, Σx_j é um limite superior para o número de erros de treinamento. A Figura 2 mostra a variável solta para dois pontos classificados incorretamente pelo hiperplano de separação, em que x_j é a variável solta associada a x_j, j = 1, 2.

Da mesma forma que no caso dos padrões linearmente separáveis, substituindo-se a restrição (4) pela (12), a construção de SVMs com margem suave pode ser descrita como:

Dado um conjunto de treinamento $\{(x_1, y_1), ...(x_N, y_N)\}$, encontre os valores ótimos para o vetor de pesos w e o *bias* b para que as seguintes restrições

$$y_j(w.x_j + b) + x_j \geq 1, \text{ para j = 1,..., N,}$$
$$x_j \geq 0, \text{ para j = 1,..., N,} \quad (13)$$

sejam satisfeitas quando a função custo

$$\Phi(w,x) = \frac{1}{2}(w.w) + C\sum_{j=1}^{N} x_j \quad (14)$$

é minimizada, onde C é um parâmetro de treinamento que estabelece o equilíbrio entre a complexidade do modelo e o erro de treinamento. Este

parâmetro, conhecido como constante de regularização, controla o peso do número de erros, que, como foi dito na página anterior, é limitado pelo somatório das variáveis soltas, e do tamanho da margem, que é inversamente proporcional à norma de *w* [1], [5], [8]. Valores grandes de *C*, por exemplo, atribuem maior peso ao número de erros (permitindo poucos erros) e menor peso à margem do hiperplano (gerando uma margem pequena).

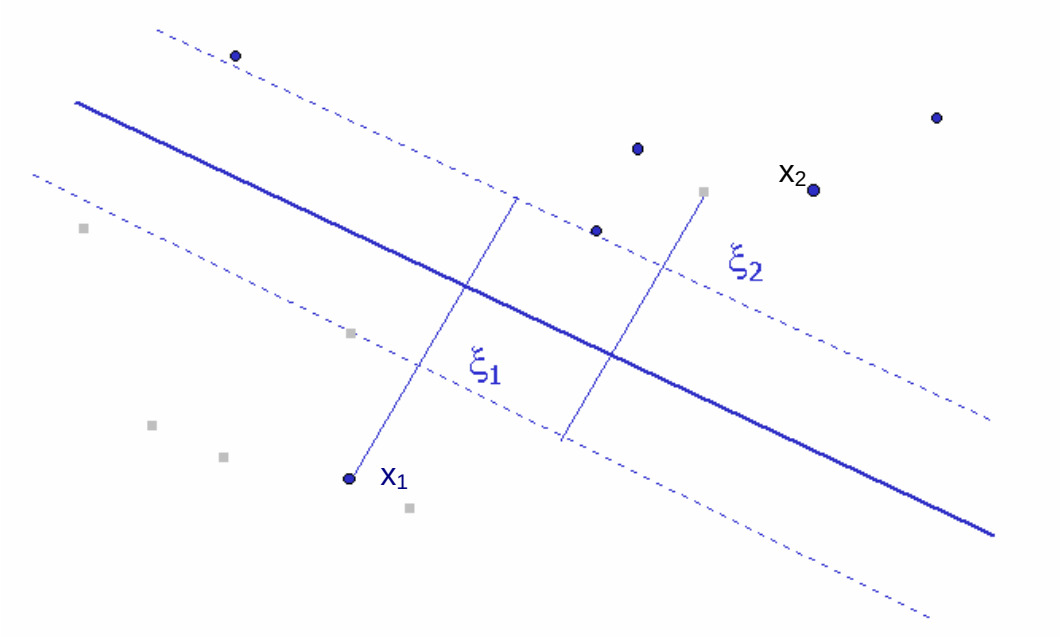

Figura 2 - Variáveis soltas

A formulação acima é chamada de problema primal. A função custo é convexa e todas as restrições são lineares. Da mesma forma que para a margem maximal, pode-se usar a teoria dos multiplicadores de Lagrange e, resolvendo o problema de forma similar, chegar ao problema dual abaixo. Note que as variáveis soltas não aparecem no problema dual.

Considerando um conjunto de treinamento $\{(x_1, y_1),..., (x_N, y_N)\}$, determinar os multiplicadores de Lagrange ótimos a_i para que

$$W(a) = \sum_{i=1}^{N} a_i - \frac{1}{2}\sum_{i,j=1}^{N} a_i a_j y_i y_j (x_i . x_j), \qquad (15)$$

seja máximo, e que satisfaçam a [1], [7]

$$\sum_{i=1}^{N} a_i y_i = 0, \quad 0 \le a_i \le C. \qquad (16)$$

De maneira similar ao caso dos padrões linearmente separáveis, pelas condições de Karush-Kuhn-Tucker [1], [5], [8], apenas os pontos em que $y_j((w.x_j) + b) = 1$ podem ter os correspondentes $a \neq 0$. Esses pontos, analogamente, são chamados de vetores suporte.

2.2.3.
SVMs não lineares

Nas seções anteriores, considera-se apenas o caso em que as SVMs são lineares, isto é, os hiperplanos de separação obtidos são combinações lineares dos atributos dados. No entanto, essa restrição pode representar uma grande desvantagem quando se considera um problema em que os padrões de treinamento não são linearmente separáveis. Para se superar essa limitação, utilizam-se máquinas não-lineares que projetam os dados de entrada em um espaço de características de dimensão maior, o que aumenta o poder computacional das máquinas lineares. O teorema de Cover garante que um espaço de entrada com padrões não linearmente separáveis pode ser transformado em um novo espaço de características em que os padrões são linearmente separáveis, desde que duas condições sejam satisfeitas: a transformação seja não linear e a dimensão do espaço de características seja suficientemente grande. Assim, é possível construir-se um hiperplano ótimo nesse espaço de características.

Na prática, a modificação necessária para se implementar as SVMs não lineares em um espaço de características de dimensão maior é mínima, bastando substituir nas equações da seção anterior x por $j(x)$, onde j é um mapeamento não linear do espaço de entrada no espaço de características.

Assim, o problema dual é

Considerando um conjunto de treinamento $\{(x_1, y_1),..., (x_N, y_N)\}$, determinar os multiplicadores de Lagrange ótimos a_i para que

$$W(a) = \sum_{i=1}^{N} a_i - \frac{1}{2}\sum_{i,j=1}^{N} a_i a_j y_i y_j (j(x_i).j(x_j)), \qquad (17)$$

seja máximo, e que satisfaçam a [1], [7]

$$\sum_{i=1}^{N} a_i y_i = 0, \quad 0 \le a_i \le C. \qquad (18)$$

De maneira similar aos casos anteriores, pelas condições de Karush-Kuhn-Tucker [1], [5], [8], somente os pontos em que $y_j((w.j(x_j)) + b) = 1$ podem ter os correspondentes $a \neq 0$. Como nos casos anteriores, esses pontos são chamados de vetores suporte.

2.2.4.
Mapeamento Implícito usando funções Kernel

A formulação apresentada por SVMs não lineares tem uma característica singular: um produto interno realizado no espaço de características, representado como $j(x_i).j(x_j)$ na equação 17. Se existe uma função simétrica *K* tal que

$$K(x_i,x_j) = j(x_i).j(x_j) = K(x_j,x_i), \quad (19)$$

o custo computacional da avaliação do operador $j(.)$ pode ser evitado, já que a função *K* só depende de variáveis do espaço de entradas. Tais funções *K* são conhecidas como kernel de produto interno ou, simplesmente, funções kernel. A caracterização de quando uma função pode ser um kernel é dada pelo teorema de Mercer [1]. Alguns kernels conhecidos são dados na Tabela 1. Para exemplos de outros tipos de kernel, indica-se [32].

Tabela 1 - Exemplos de Kernels

Kernel	Expressão	Parâmetros
Polinomial	$K(x_i, x_j) = ((x_i.x_j) + a)^p$	a, p
RBF	$K(x_i, x_j) = exp\left(-\frac{1}{2s^2}\|x_i - x_j\|^2\right)$	s^2
Perceptron	$K(x_i,x_j) = \tanh(b_0(x_i.x_j) + b_1)$	b_0, b_1

Na máquina de vetor suporte RBF, o número de funções de base radial e seus centros são determinados automaticamente pelo número de vetores suporte e seus valores, respectivamente.

Na máquina de vetor suporte do tipo perceptron de duas camadas, o número de neurônios ocultos e seus vetores de peso são também determinados automaticamente pelo número de vetores suporte e seus pesos, respectivamente.

No caso de padrões não linearmente separáveis, a equação do hiperplano de separação é:

$$(w.j(x)) + b = 0,$$

onde o vetor $j(x)$ representa o mapeamento do vetor de entrada x no espaço de características, w é o vetor de pesos e o escalar b é o *bias*. No entanto, o mapeamento $j(.)$ é desconhecido na maioria dos casos. Para se escrever a equação do hiperplano em função do kernel K, pode-se usar a equação 20 abaixo.

A expressão do vetor de pesos, que pode ser obtida de forma similar à da equação 7, é:

$$w = \sum_{i=1}^{N} a_i y_i j(x_i). \tag{20}$$

Assim, a equação do hiperplano de separação também pode ser representada usando-se kernel:

$$(w.j(x)) + b = 0 \Rightarrow \sum_{i=1}^{N} a_i y_i (j(x_i).j(x)) + b = 0$$

$$\Rightarrow \sum_{i=1}^{N} a_i y_i K(x_i, x) + b = 0. \tag{21}$$

Substituindo a função kernel na equação 17, chega-se ao problema dual abaixo.

Dado um conjunto de entrada $\{(x_1, y_1),..., (x_N, y_N)\}$, o treinamento da SVM consiste em determinar os multiplicadores de Lagrange ótimos a_i para que

$$W(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \left(\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \right), \tag{22}$$

seja máximo, e que satisfaçam a [1], [7]

$$\sum_{i=1}^{N} a_i y_i = 0, \quad 0 \le a_i \le C. \tag{23}$$

SVMs foram originalmente desenvolvidas para classificação binária. No caso de classificação em múltiplas classes [17], [19], [33], [34], [35], [36], [37], [38], [39], [40], [41], [42], é necessária a utilização de algum método para estender a SVM binária (método de Crammer e Singer) ou para combinar os resultados das SVMs binárias ("decomposição um por classe" ("one-against-all") e "separação das classes duas a duas" ("one-against-one)). Na seção seguinte, são apresentados estes três métodos.

2.3.
Classificação em Múltiplas Classes

2.3.1.
Introdução

SVMs foram originalmente desenvolvidas para classificação binária. No caso de classificação em k classes, k > 2, existem duas abordagens básicas para estender seu esquema (ver Figura 3). A primeira é a redução do problema de múltiplas classes a um conjunto de problemas binários. Dois métodos usam essa abordagem: "decomposição um por classe" ("one-against-all") [5], [17], [18], [21]; e "separação das classes duas a duas" ("one-against-one") [5], [17], [18], [19], [20]. A segunda abordagem é a generalização de SVMs binárias para mais de duas classes. O método que utiliza essa abordagem é o método de Crammer e Singer [17], [22].

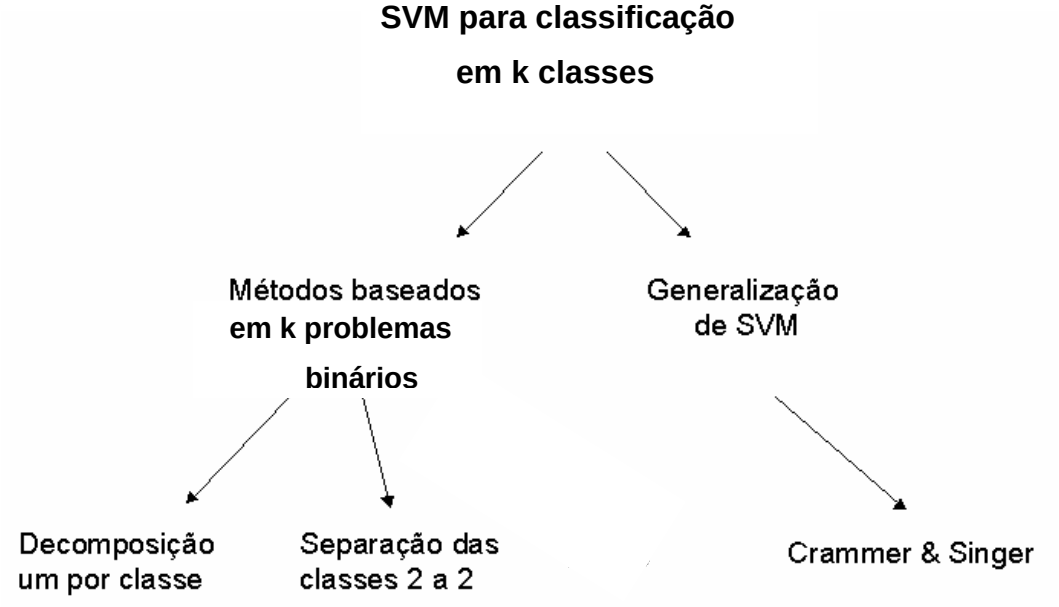

Figura 3 - Métodos para Classificação em Múltiplas Classes

O método de decomposição um por classe baseia-se na construção de k SVMs de classificação binária para separar uma classe de todas as outras. Em seguida, os resultados de todas as SVMs são agrupados, fazendo-se a classificação desejada nas k classes.

Já o método de separação das classes duas a duas usa uma SVM binária para distinguir cada par de classes. Assim, são construídas k(k-1)/2 SVMs. A classificação final é obtida a partir do resultado de todas as SVMs.

O método de Crammer e Singer usa uma maneira mais natural de resolver o problema de classificação em k classes, k > 2, que é construir uma função de decisão considerando todas as classes de uma vez. Nesse método, todos os exemplos de treinamento são usados ao mesmo tempo.

Em todos os métodos, supõe-se que sejam dados N pontos de treinamento (x_i,y_i), onde $x_i \in \mathbb{R}^n$ e $y_i \in \{1,...,k\}$ é a classe de x_i. Além disso, como no caso de SVM binária, *C* é a constante de regularização, que estabelece o equilíbrio entre a complexidade do modelo e o erro de treinamento.

As seções seguintes descrevem em mais detalhes os diferentes métodos existentes para classificação em múltiplas classes.

2.3.2.
Decomposição um por classe

Esse método é o mais antigo e o mais empregado. Para uma classificação em k classes, são resolvidos k problemas binários, isto é, é construída uma SVM para cada uma das k classes. A construção da *i*-ésima SVM, $i \in \{1,...,k\}$, é feita usando todos os padrões de treinamento, os exemplos da classe *i* com saída $y = 1$ e os outros exemplos com saída $y = -1$.

Nesse caso, resolve-se o seguinte problema: determine os valores ótimos para o vetor de pesos w^i e o *bias* b^i, considerando o conjunto de treinamento $\{(x_1, y_1), ...,(x_N\ y_N)\}$, para que as restrições

$$(w^i.j(x_t)) + b^i + (x^i)_t \geq 1, \text{ se } y_t = i, \tag{24}$$

$$(w^i.j(x_t)) + b^i + (x^i)_t \leq -1, \text{ se } y_t \neq i, \tag{25}$$

$$(x^i)_t \geq 0,\ t = 1,...,N, \tag{26}$$

sejam satisfeitas quando a função custo

$$\Phi(w^i, x^i) = \frac{1}{2}(w^i.w^i) + C\sum_{t=1}^{N}(x^i)_t \tag{27}$$

for minimizada.

A saída da *i*-ésima SVM é a *i*-ésima função de decisão, a qual é dada por $(w^i.j(x)) + b^i$.

Cada uma das k SVMs tem, a princípio, um valor diferente como saída. A classe de um dado ponto x é resultado da combinação das k saídas, o que pode ser feito de várias maneiras, como, por exemplo, efetuar uma combinação linear das saídas de todas as k SVMs ou usar um outro método de classificação para decidir a classe final [43]. A maneira de se combinar as k saídas das SVMs que será aqui usada é a mais comum. Um ponto x pertence à classe que tem maior função de decisão $(\text{argmax}_{i=1,...,k}((w^i.j(x)) + b^i))$.

2.3.3.
Separação das classes duas a duas

Este método constrói k(k-1)/2 classificadores binários, em que cada um é treinado com dados de 2 classes. Para os exemplos de treinamento da classe *i* e da classe *j*, $i \neq j$, o problema de otimização é minimizar:

$$\Phi(w^{ij}, x^{ij}) = \frac{1}{2}(w^{ij}.w^{ij}) + C\sum_{t=1}^{N}(x^{ij})_t \ , \qquad (28)$$

com as restrições

$$(w^{ij}.j(x_t)) + b^{ij} + (x^{ij})_t \geq 1, \text{ se } y_t = i, \qquad (29)$$

$$(w^{ij}.j(x_t)) + b^{ij} + (x^{ij})_t \leq -1, \text{ se } y_t = j, \qquad (30)$$

$$(x^{ij})_t \geq 0,\ t = 1,..., N. \qquad (31)$$

Como para a resolução de um problema com k classes são construídas k(k-1)/2 SVMs, esse número, para k > 2, é maior do que o número de SVMs construídas no método de decomposição um por classe. No entanto, os problemas resolvidos são menores. O método de decomposição um por classe usa todos os pontos de treinamento na construção das SVMs. Já o método de separação das classes duas a duas usa, em média, 2N/k pontos de treinamento para cada SVM.

Há duas maneiras de se combinar os resultados das k(k-1)/2 SVMs: estratégia do voto e uso de um grafo acíclico dirigido.

A decisão da classe utilizando-se a estratégia do voto [17] é feita do seguinte modo: se o sinal de $((w^{ij}.j(x)) + b^{ij})$ é positivo, então soma-se um voto para a classe *i*; se não, soma-se um voto à classe *j*. Assim, *x* pertence à classe com maior votação.

O método com o uso do grafo acíclico dirigido [17] é chamado de DAGSVM (Directed Acyclic Graph SVM). A decisão da classe é feita usando-se um grafo acíclico dirigido com um nó externo (raiz), k(k-1)/2 nós internos e k folhas. Cada nó é uma SVM binária das classes *i* e *j*, representada por SVM_{ij}. Dado um ponto de teste *x*, começando-se na raiz, a função binária de decisão é avaliada. Caminha-se no grafo para a direita ou para a esquerda, dependendo do valor de saída. Assim, percorre-se um caminho antes de se chegar à folha com a classe prevista. Em outras palavras, avalia-se o ponto de teste *x* com a SVM_{ij}, em uma dada ordem, determinando-se a que classe *x* não pertence. Assim, na primeira etapa já se sabe que *x* não é da classe *i* ou não é da classe *j*. Em seguida, avalia-se a próxima SVM de maneira análoga, até que só uma classe não tenha sido eliminada, a qual é estabelecida como a classe de *x*. Um

exemplo para classificação em 3 classes com o método DAGSVM é exposto na Figura 4. Como são 3 classes, definem-se 3 SVMs: SVM12, que separa a classe 1 da 2; SVM13, que separa as classes 1 e 3; e SVM23, que distingue as classes 2 e 3. Na Figura 4, a primeira SVM utilizada é a SVM12. Essa SVM classifica os pontos como da classe 1 ou da classe 2. Como o método DAGSVM elimina as classes até chegar à classe final, na primeira etapa, a classe 1 é eliminada se *x* é classificado como da classe 2 pelo SVM12 e caso contrário a classe 2 é eliminada. Em seguida, o ponto *x* é classificado pela SVM13 independente do resultado da classificação da SVM12. Assim, mesmo se *x* for classificado como da classe 1 por SVM12, a SVM13 pode classificá-lo como da classe 3 (Figura 4(e)). Após a classificação com a SVM13, uma classe é, novamente, eliminada. A classe eliminada pode ser a mesma da etapa anterior (classe 1), o que não acrescenta nenhuma informação. Nesse caso, e somente nele, o ponto *x* é classificado pela SVM23 para eliminar outra classe. A classe 3 é eliminada se o ponto *x* é classificado pela SVM23 como da classe 2 (Figura 4(b)). Caso contrário, a classe 2 é eliminada (Figura 4(c)). Nos casos em que classes diferentes são descartadas pelas SVM12 e SVM13, o ponto *x* não é classificado pela SVM23 (Figuras 4(a), (d) e (e)).

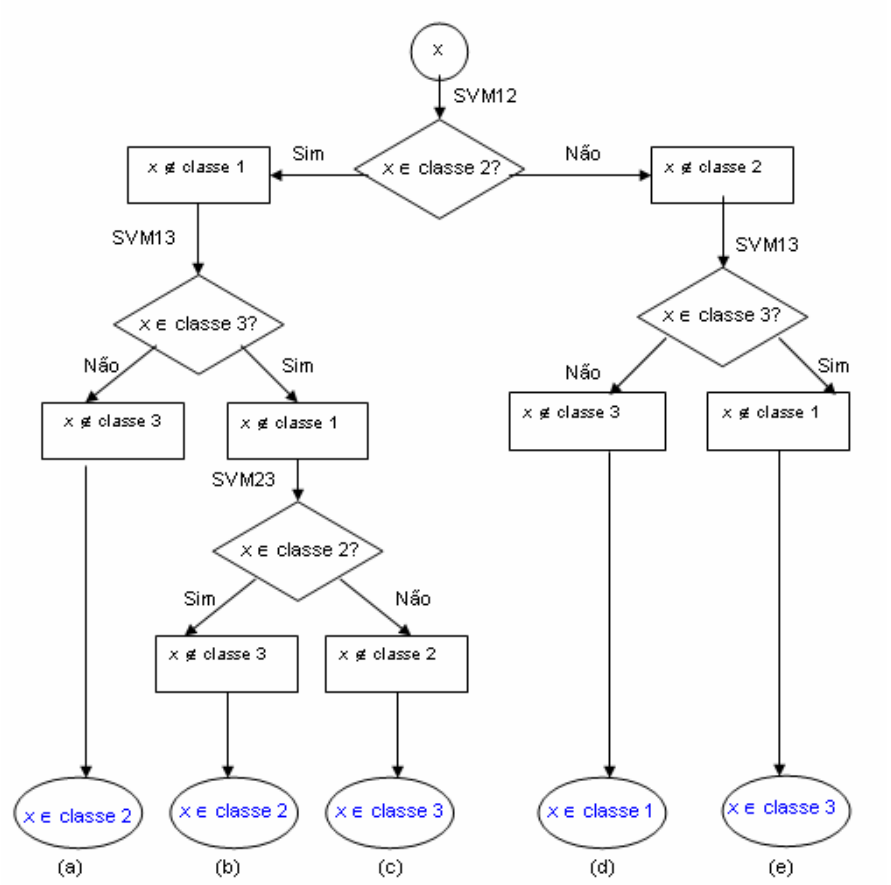

Figura 4 - DAGSVM

2.3.4.
Método de Crammer e Singer

No método de Crammer e Singer, o problema de classificação em k classes, k > 2, é resolvido com um único problema de otimização (generalização do problema de otimização da SVM binária), com a diferença de que, na construção desse método, os coeficientes b^i não aparecem [22].

Na solução do problema, são considerados um peso para cada uma das classes (w^i é o peso para classe i, $i \in \{1,...,k\}$) e uma variável solta para cada um dos exemplos de treinamento (x_t é a variável solta associada ao padrão (x_t,y_t), t = 1,...,N). Em seguida, é feito um somatório que define a função a ser minimizada.

Assim, o método minimiza a função

$$\Phi((w^1,...,w^k),(x_1,...,x_N)) = \frac{1}{2}\sum_{i=1}^{k}(w^i.w^i) + C\sum_{t=1}^{N}x_t \quad (32)$$

sujeito a

$$(w^{yt}.j(x_t)) - (w^i.j(x_t)) + x_t \geq 1,\ t = 1,...,N,\ i \in \{1,...,k\} \text{ e } i \neq y_t \quad (33)$$

$$x_t \geq 0,\ t = 1,...,N. \quad (34)$$

A função de decisão é

$$f(x) = \text{argmax}_{i=1,...,k}(w^i.j(x)). \quad (35)$$