

# Tosco estudo a respeito de Support Vector Machines

Bogdano Arendartchuk

17 de agosto de 2010

## Resumo

Este artigo apresenta um estudo a respeito de Support Vector Machines (SVMs) feito durante o desenvolvimento do trabalho de conclusão do curso de Ciência da Computação da Universidade XXXXXXXXXX, que foi reprovado. É feita uma introdução a aprendizado supervisionado, aprendizado estatístico e então são apresentados os conceitos básicos. Finalmente são brevemente descritas as técnicas para a implementação de SVMs multiclases, com margens suaves e também não-lineares.

## 1 Introdução

Como requisito para o desenvolvimento do trabalho de conclusão de curso da Universidade XXXXXXXXXX, foi necessário desenvolver um estudo para compreender o funcionamento das *Support Vector Machines* – Máquinas de Vetores de Suporte (SVMs). Portanto, foram estudadas as principais referências disponíveis do assunto e este material foi desenvolvido como parte da “Revisão bibliográfica”. Infelizmente o trabalho foi reprovado (não por este estudo, mas ele certamente seria escrutinado se a *metodologia* da nossa proposta não estivesse suficientemente clara para os queridos avaliadores.

O texto segue a linha de raciocínio que foi usada por Lorena e Carvalho (2003), com algumas descrições de Cherkassky e Ma (2004), obviamente devidamente citados.

Antes de apresentar *Support Vector Machines*, é necessário conhecer alguns conceitos que são utilizadas por praticamente toda a literatura que aborda este tema. A organização do texto consiste em uma apresentação do que é aprendizado supervisionado na seção 2, descrevendo então o aprendizado estatístico na seção 3 e então detalha-se SVMs na seção 4.

Este texto foi desenvolvido com visando a aplicação de SVMs para a área de categorização de textos aonde, segundo Joachims, Nédellec e Rouveirol (1998),

o uso de SVMs lineares tem bom desempenho. Assim, o SVMs não-lineares são apresentadas de maneira muito superficial. Mesmo assim, isso não justifica a pobre descrição dada para a classificação multiclases da seção 4.3.

## 2 Formalização do aprendizado supervisionado

No problema de aprendizado supervisionado existe a figura de um professor que indica qual é o rótulo correto para cada exemplo (HAYKIN, 1994 apud LORENA; CARVALHO, 2003). Sendo assim, cada exemplo pode ser descrito na forma  $(\mathbf{x}_i, y_i)$ , aonde  $\mathbf{x}_i$  denota um exemplo e  $y_i$  seria a classe ou rótulo. Também, após o processo de treinamento, pode-se descrever o classificador como uma função  $f(\mathbf{x}) = y$ , sendo que  $\mathbf{x}$  não é necessariamente um dos valores de  $\mathbf{x}_i$ .

Os valores dos rótulos que os exemplos podem assumir podem ser discretos ou contínuos. Para o caso dos contínuos assume-se que é possível obter  $1, \dots, k$  valores. Quando  $k = 2$ , o problema é denominado como sendo de *classificação binária*. Quando  $k > 2$  denomina-se como sendo um problemas de *classificação multiclases*.

Os exemplos  $\mathbf{x}_i$ , são representados por vetores com as *características* (também denominadas *atributos*) de cada exemplo. Cada exemplo  $\mathbf{x}_i$  possui  $m$  atributos e também pode ser representado como  $\mathbf{x}_i = (x_{i1}, \dots, x_{im})$ . Os atributos podem assumir dois tipos de valores: nominais ou contínuos. Os atributos nominais assumem valores que não possuem ordem entre si e sua representação tem função simbólica (por exemplo: segunda-feira, azul, não). Os atributos contínuos possuem ordem entre si e comumente representam valores dos domínios  $\mathbb{Z}$  e  $\mathbb{R}$ .

O objetivo de uma técnica de aprendizado de máquina é obter uma função  $f(\mathbf{x}) = y$  que obtenha um  $y$  adequado aos exemplos que foram apresentados pelo *professor* por meio de indução (OSUNA; FREUND; GIROSI, 1997).

## 3 Aprendizado estatístico

Sendo  $f$  um classificador e  $F$  o conjunto dos classificadores possíveis gerados pelo algoritmo de aprendizado, pode-se descrever o aprendizado como a busca por um classificador  $\hat{f}$  que melhor aproxime os resultados aos dos exemplos  $T$ . Para descobrir qual é o melhor classificador do conjunto  $F$ , utiliza-se uma *medida de discrepância* ou perda, que indica o quanto de erro houve com um classificador  $f$  em relação aos exemplos conhecidos (VAPNIK, 1998). Para problemas de classificação binária, emprega-se a função de custo descrita na equação 1, que retorna 1 quando classificado corretamente e 0 quando erra.

$$c(f(\mathbf{x}), y) = \frac{1}{2}|y - f(\mathbf{x})| \quad (1)$$

Quando a distribuição  $P(\mathbf{x}, y)$  é conhecida, pode-se calcular o risco esperado de um classificador com a equação 2. Porém, como a distribuição dos exemplos não é conhecida, é necessário induzir um classificador  $\hat{f}$  a partir dos exemplos disponíveis de maneira que seja possível diminuir o erro sobre os dados (SCHÖLKOPF; SMOLA, 2002).

$$R(f) = \int c(f(\mathbf{x}))dP(\mathbf{x}, y) \quad (2)$$

É possível medir o risco do classificador  $\hat{f}$  avaliando-se o *risco empírico*, que é descrito na equação 3.

$$R_{emp}(f) = \frac{1}{n} \sum_{i=1}^n c(f(\mathbf{x}_i), y_i) \quad (3)$$

A partir da lei dos grandes números, pode-se observar que o risco empírico aproxima-se do risco esperado quando o número de amostras tende para infinito (VAPNIK, 1998 apud OSUNA; FREUND; GIROSI, 1997). Porém, para exemplos pequenos não é possível garantir que o risco empírico aproxime-se do risco esperado. Lorena e Carvalho (2003) exemplifica esta situação com o caso de um classificador que aprende um conjunto de exemplos mas classifica casos novos aleatoriamente, no qual o risco empírico seria 0, mas o risco esperado seria 0,5. Além disso, a simples minimização do risco empírico não é ideal para um algoritmo de aprendizado porque o classificador resultante poderia estar sobreajustado ao conjunto de exemplos.

Com o objetivo de dificultar a possibilidade de que haja sobreajuste do classificador, Vapnik (2000) define um limite no risco empírico baseado na complexidade do conjunto de funções de classificadores  $F$ . A inequação 4 apresenta este limite, aonde o risco empírico está associado também há um *termo de capacidade* e, aonde  $h$  representa a complexidade do conjunto de classificadores que está sendo avaliado.

$$R(f) \leq R_{emp}(f) + \sqrt{\frac{h(\ln(\frac{2n}{h}) + 1) - \ln(\frac{\phi}{4})}{n}} \quad (4)$$

Essa medida  $h$ , é conhecida também como *dimensão VC* (de Vapnik-Chervonenkis) e representa a quantidade de exemplos do treinamento que podem ser separados com um determinado conjunto de funções classificadoras de  $F$ . Estes conjuntos  $F_k$  são subconjuntos de  $F$  e são ordenados de acordo com seus valores de  $k$ , de maneira que podem ser representados na forma  $F_0 \subset F_1 \subset \dots \subset F_q \subset F$ .

Como o risco empírico diminui à medida que o valor de  $h$  aumenta, pode-se com isso encontrar um valor ótimo, ou em outras palavras um conjunto  $F_k$ , que represente um compromisso aceitável entre o valor do risco empírico (sendo este

sempre o menor dentre todas as funções de  $F_k$ ) e a complexidade da função que representa o classificador. Porém, o cálculo da dimensão VC nem sempre é definido ou seu valor é infinito. A escolha da melhor função de  $F_k$  representa o principal conceito da *minimização do risco empírico* (MULLER et al., 2001 apud LORENA; CARVALHO, 2003). A figura 1 apresenta este conceito.

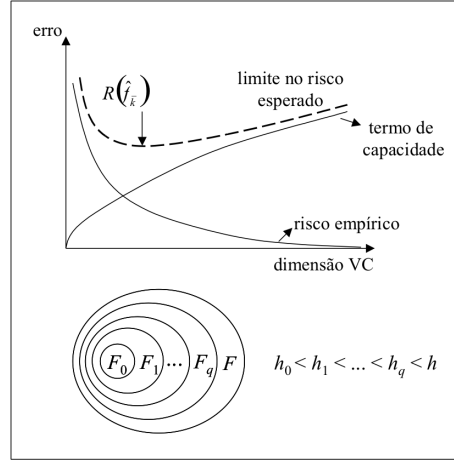


Figura 1: Princípio da minimização do risco estrutural e dimensões VC (LORENA, 2003)

Para o caso de classificadores lineares  $f(\mathbf{x}) = \mathbf{x} \cdot \mathbf{w} + b$ , é possível melhorar a qualidade da classificação introduzindo o conceito de *margem* entre os exemplos e o hiperplano separador definido por  $f(x)$  (SMOLA; BARTLETT, 2000). Para o caso de problemas binários com classificação  $y_i$  sendo  $y_i \in -1, +1$ , é possível definir esta margem (que é negativa em caso de classificação incorreta, por meio da definição de  $\varrho(f(\mathbf{x}_i, y_i))$  na equação 5.

$$\varrho(f(\mathbf{x}_i), y_i) = y_i f(\mathbf{x}_i) \quad (5)$$

Sendo que  $I(q) = 1$  se  $q$  é verdadeiro e  $I(q) = 0$  se  $q$  é falso.

Tendo a definição de erro de classificação baseado em uma margem até o hiperplano separado, é possível então estabelecer uma medida para a qualidade da separação de um dado classificador  $f(\mathbf{x})$  por meio da equação 6.

$$R_\rho(f) = \frac{1}{n} \sum_{i=1}^n I(y_i f(\mathbf{x}_i) < \rho) \quad (6)$$

Com isso, é possível estabelecer o limite descrito pela equação 7. Nela, tem-se  $c$  com probabilidade  $1 - \theta \in [0, 1]$ , para  $\rho > 0$  e  $F$  sendo uma função linear, com  $\|\mathbf{x}\| \leq R$  e  $\|\mathbf{w}\| \leq 1$ . Esse limite constitui a *minimização do risco*

estrutural, introduzido por (VAPNIK, 1998).

$$R(f) \leq R_\rho(f) + \sqrt{\frac{c}{n} \left( \frac{R^2}{\rho^2} \log^2 \left( \frac{n}{\rho} \right) + \log \left( \frac{1}{\theta} \right) \right)} \quad (7)$$

## 4 Support Vector Machines — SVMs

A técnica de aprendizado de máquina supervisionado conhecida como *Support Vector Machine* (SVM), ou *Máquinas de Vetores de Suporte* foi introduzida por Boser, Guyon e Vapnik em 1992 com a publicação de *A training algorithm for optimal margin classifiers*. Ela basei-se no trabalho da teoria do aprendizado estatístico desenvolvida por Vapnik et al. desde a década de 1960.

SVMs usam o princípio de que quanto mais largas as margens de um hiperplano separador de uma função, que foi explicado na seção 3 por meio do conceito de minimização do risco estrutural, maiores são as chances de que ele possa prover uma boa generalização dos dados que estão sendo usados como exemplo (CHAPELLE et al., 2002). Então, SVMs buscam encontrar um hiperplano descrito por  $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b = 0$  que tenha maior margem entre as classes e menor complexidade estrutural por meio da resolução de um problema de otimização.

Para encontrar o hiperplano separador ideal, é necessário utilizar (pelo menos) dois pontos do conjunto de exemplos. Sejam  $\mathbf{x}_1$  e  $\mathbf{x}_2$  dois exemplos do conjunto de treinamento  $T$  que possui um conjunto de exemplos  $X$  com rótulos do conjunto de exemplos  $Y = \{-1, +1\}$ , e cada um deles fica em um lado diferente do hiperplano separador. Para encontrar o hiperplano separador  $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b = 0$  entre  $\mathbf{x}_1$  e  $\mathbf{x}_2$ , é necessário conhecer o vetor  $\mathbf{w}$ , que deve ser normal a este hiperplano e que é usado para calcular o tamanho da margem.

Como um dos objetivos do problema de otimização é obter uma margem larga entre os exemplos, pode-se restringir a busca por  $\mathbf{w}$  apenas em termos do cálculo do tamanho da margem. Para isso, é necessário calcular a distância entre dois hiperplanos que são formados pelos exemplos  $\mathbf{x}_1$  e  $\mathbf{x}_2$ . Sejam  $H_1 : \mathbf{w} \cdot \mathbf{x} + b = +1$  e  $H_2 : \mathbf{w} \cdot \mathbf{x} + b = -1$  dois hiperplanos que ficam paralelamente acima e abaixo, respectivamente, do hiperplano separador. Além disso, assume-se que  $H_1$  passa por  $\mathbf{x}_1$  e  $H_2$  passa por  $\mathbf{x}_2$ . A figura 2 mostra a relação entre os hiperplanos e os pontos  $\mathbf{x}_1$  e  $\mathbf{x}_2$ .

Conhecendo os hiperplanos, agora é possível calcular a distância entre os hiperplanos que servem de fronteira entre cada ponta da margem do hiperplano separador. A equação 8 apresenta o cálculo necessário para projetar  $\mathbf{x}_1 - \mathbf{x}_2$  na direção de  $\mathbf{w}$ , que é perpendicular ao hiperplano separador  $\mathbf{w} \cdot \mathbf{x} + b = 0$ .

$$(x_1 - x_2) \left( \frac{\mathbf{w}}{\|\mathbf{w}\|} \cdot \frac{(\mathbf{x}_1 - \mathbf{x}_2)}{\|\mathbf{x}_1 - \mathbf{x}_2\|} \right) \quad (8)$$

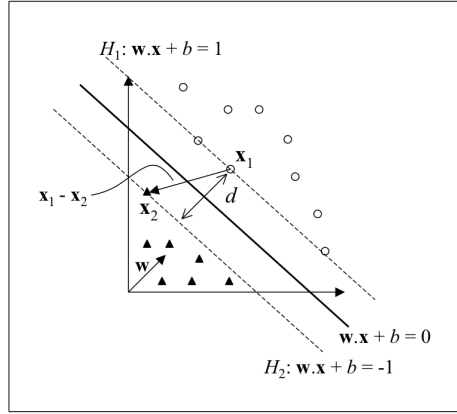


Figura 2: Hiperplano separador e margens (LORENA, 2003)

Sabendo que  $\mathbf{w} \cdot \mathbf{x}_1 + b = +1$  e  $\mathbf{w} \cdot \mathbf{x}_2 + b = -1$ , e levando em conta que deseja-se saber o comprimento do vetor resultante, é usada a norma da equação 8 para chegar à equação 9, que indica a distância  $d$  utilizada na figura 2:

$$\frac{2}{\|\mathbf{w}\|} \quad (9)$$

Portanto, as distâncias entre  $\mathbf{x}_1$ ,  $\mathbf{x}_2$  e o hiperplano separador  $\mathbf{w} \cdot \mathbf{x} + b = 0$  são  $\frac{1}{\|\mathbf{w}\|}$ , e com isso, é possível definir o problema de otimização como definido pelo problema de otimização 10. A restrição 11 indica que  $H_1$  e  $H_2$  devem passar, respectivamente, pelos vetores  $\mathbf{x}_1$  e  $\mathbf{x}_2$ .

$$\text{Maximizar } \frac{2}{\|\mathbf{w}\|} \quad (10)$$

$$\text{sujeito a } y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \geq 0 \quad i = 1, \dots, n. \quad (11)$$

No problema de maximização 10,  $\frac{2}{\|\mathbf{w}\|}$  pode também ser descrito como um problema de minimização de  $\|\mathbf{w}\|^2/2$ :

$$\text{Minimizar } \frac{\|\mathbf{w}\|^2}{2} \quad (12)$$

$$\text{sujeito a } y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \geq 0 \quad i = 1, \dots, n.$$

A partir deste ponto, o problema 12 pode ser resolvido com técnicas de programação quadrática (PQ) (OSUNA; FREUND; GIROSI, 1997). Este tipo de problema pode ser solucionado utilizando uma função Lagrangiana e adicionando as restrições à função objetivo junto com os multiplicadores de Lagrange  $\alpha_i$  (SMOLA; BARTLETT, 2000). A equação 13 deve ser minimizada, o que significa maximizar  $\alpha_i$  e minimizar  $\mathbf{w}$  e  $b$ . O problema é representado desta

forma para que a restrição 11 possa ser representada na forma dos multiplicadores  $\alpha_i$ , o que facilita os cálculos mais adiante, e também porque os dados de treinamento apenas aparecem na forma de produtos entre vetores (BURGES, 1998), o que permite o uso de *kernels*, que são apresentados na seção 4.2.

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^n \alpha_i (y_i (\mathbf{w} \cdot \mathbf{x}_i + b) - 1) \quad (13)$$

Tem-se ponto de sela, então:

$$\frac{\partial L}{\partial b} = 0 \quad \text{e} \quad \frac{\partial L}{\partial \mathbf{w}} = 0 \quad (14)$$

E com isso:

$$\sum_{i=1}^n \alpha_i y_i = 0 \quad (15)$$

$$\mathbf{w} = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i \quad (16)$$

Assim, substituindo equações 15 e 16, é possível formular o problema de otimização:

$$\begin{aligned} \text{Maximizar}_{\alpha} \quad & \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j) \\ \text{sujeito a} \quad & \begin{cases} \alpha_i \geq 0, \forall i = 1, \dots, n \\ \sum_{i=1}^n \alpha_i y_i = 0 \end{cases} \end{aligned} \quad (17)$$

A equação 17 é denominada a *forma dual* do problema, ao passo que a formulação original na equação 12 é denominada a *forma primal*, baseada no trabalho de Fletcher (apud BURGESS, 1998).

É possível utilizar as condições KKT (de Karush-Kuhn-Tucker), descritas em Bertsekas (1999, proposição 3.3.1), visto que o problema de otimização 17 possui restrições lineares e a função objetivo é convexa (BURGES, 1998). Assim, segundo essas condições, é possível encontrar  $\mathbf{w}^*$  e  $b^*$  que podem ser considerados solução ótima para o problema a partir da solução do problema dual ao encontrar  $\alpha_i^*$ :

$$\alpha_i^* (y_i (\mathbf{w}^* \cdot \mathbf{x}_i + b^*) - 1) = 0, \forall i = 1, \dots, n \quad (18)$$

Nesta equação  $\alpha_i^*$  é diferente de zero apenas para os valores que tocam a borda das margens do hiperplano de decisão ( $H_1$  e  $H_2$ ). Assim, esses dados são chamados de *vetores de suporte*, pois são os dados mais significativos para a localização de hiperplano  $\mathbf{w} \cdot \mathbf{x} + b = 0$ .

E para calcular  $b^*$ , de acordo com 18:

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

Aonde  $n_{SV}$  é o número de vetores de suporte e  $SV$  o conjunto dos mesmos.

Finalmente, obtém-se a seguinte função classificadora:

$$g(\mathbf{x}) = \text{sinal}(f(\mathbf{x})) = \text{sinal}\left(\sum_{\mathbf{x}_i \in SV} y_i \alpha_i^* \mathbf{x}_i \cdot \mathbf{x} + b^*\right) \quad (19)$$

## 4.1 Margens suaves

Para tratar os casos em que há *outliers* nos exemplos, ou seja, dados que estão rotulados incorretamente ou com algum ruído, utiliza-se a técnica de margens suaves, que é uma saída mais simples que SVMs não-lineares (BURGES, 1998).

Utiliza-se variáveis de folga  $\xi_i$  para cada exemplo  $\mathbf{x}_i$  do conjunto de treinamento. Essas variáveis são adicionadas à restrição do problema primal:

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i, \quad \forall i = 1, \dots, n \quad (20)$$

Ao passo que a função objetivo é reformulada como:

$$\underset{\mathbf{w}, b, \xi}{\text{Minimizar}} \quad \frac{1}{2} \|\mathbf{w}\|^2 + C \left( \sum_{i=1}^n \xi_i \right) \quad (21)$$

A constante  $C$  impõe uma penalização à violação das restrições 20 do problema de otimização. O valor desta constante é definida pelo usuário e sua definição depende de testes baseados no conjunto de treinamento. Algumas abordagens para a escolha deste parâmetro foram apresentadas por Chapelle et al. (2002), Cherkassky e Ma (2004), Quang, Zhang e Li (2002) e (BEN-HUR; WESTON, 2010).

## 4.2 Classificação não-linear

Segundo o teorema de Cover (1965), as chances de que um conjunto de exemplos não linearmente separável possa ser separado por um hiperplano é grande quando este é disposto em um espaço de maior dimensionalidade. Assim, a implementação das máquinas de vetores de suporte utiliza esta técnica para conseguir separar dados ainda de maneira *linear* (BURGES, 1998).

Com isso, os dois vetores  $\mathbf{x}$  e  $\mathbf{x}_i$ , que são utilizados na função de decisão 19, são convertidos para o espaço de maior dimensão por um mapeamento  $\Phi : X \rightarrow \mathfrak{S}$ , aonde  $X$  é o espaço de entrada e  $\mathfrak{S}$  o *espaço de características*.

Adicionalmente, o produto entre os vetores  $\mathbf{x}$  e  $\mathbf{x}_i$  é representado como uma função  $K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$ . Esta função é chamada de *função kernel* e tem

Tipo de kernel	Função	Parâmetros
Polinomial	$(\delta(\mathbf{x}_i \cdot \mathbf{x}_j) + \kappa)^d$	$\delta, \kappa, e d$
Gaussiano	$\exp(-\sigma\ \mathbf{x}_i - \mathbf{x}_j\ ^2)$	$\sigma$
Sigmoidal	$\tanh(\delta(\mathbf{x}_i \cdot \mathbf{x}_j) + \kappa)$	$\delta e \kappa$

Tabela 1: Funções de kernel comumente usadas

por finalidade permitir o uso de funções que atendem às condições definidas pelo teorema de Mercer (BURGES, 1998, p. 141). Esta forma de representação permite que use-se estas funções de kernel dentro da implementação de uma SVM sem a necessidade de conhecimento dos detalhes internos das mesmas. O quadro 1 apresenta alguns dos kernels mais populares utilizados com SVMs (LORENA; CARVALHO, 2003).

### 4.3 Classificação multiclass

Como SVMs fazem classificação binária, é necessário o uso de alguma técnica para adaptar problemas de classificação multiclass. As mais populares são *um contra todos* (*one-against-all*) e *um contra um* (*one-against-one*).

Na técnica *um contra todos*, é treinada uma SVM para cada classe contra todas as outras classes ao mesmo tempo. Vapnik (1998) propôs uma extensão a esta técnica para utilizar os valores contínuos de cada SVM (ao invés do retornado por sinal) e ordenar as classes decendentemente de acordo com o módulo da classificação de cada uma (ABE, 2003).

Na técnica de classificação *um contra um*, ou também chamado de *pairwise*, cada classe é treinada contra outra classe do problema; a classe selecionada é a que foi selecionada mais vezes nas classificações contra todas as outras classes. Isso resulta, em um problema de classificação de  $n$  classes, em  $n(n-1)/2$  SVMs (KRESSEL, 1999).

## Referências

- ABE, S. Analysis of multiclass support vector machines. In: *Proceedings of International Conference on Computational Intelligence for Modelling Control and Automation (CIMCA '2003)*. [S.l.: s.n.], 2003. p. 385–396.
- BEN-HUR, A.; WESTON, J. A User's Guide to Support Vector Machines. *Methods in molecular biology (Clifton, NJ)*, Springer, v. 609, p. 223, 2010.
- BERTSEKAS, D. Nonlinear programming. Athena Scientific, Belmont, MA, USA, 1999.

- BOSER, B.; GUYON, I.; VAPNIK, V. A training algorithm for optimal margin classifiers. In: ACM. *Proceedings of the fifth annual workshop on Computational learning theory*. [S.l.], 1992. p. 144–152.
- BURGES, C. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, Springer, v. 2, n. 2, p. 121–167, 1998.
- CHAPELLE, O. et al. Choosing multiple parameters for support vector machines. *Machine Learning*, Springer, v. 46, n. 1, p. 131–159, 2002.
- CHERKASSKY, V.; MA, Y. Practical selection of SVM parameters and noise estimation for SVM regression. *Neural Networks*, Elsevier, v. 17, n. 1, p. 113–126, 2004.
- COVER, T. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE transactions on electronic computers*, p. 326–334, 1965.
- FLETCHER, R. *Practical Methods of Optimization, Vol. 1 and 2*. [S.l.]: John Wiley and Sons, 1987.
- HAYKIN, S. *Neural networks: a comprehensive foundation*. [S.l.]: Prentice Hall PTR Upper Saddle River, NJ, USA, 1994.
- JOACHIMS, T.; NEDELLEC, C.; ROUVEIROL, C. Text categorization with support vector machines: learning with many relevant features. In: SPRINGER. *Machine Learning: ECML-98 10th European Conference on Machine Learning*. Chemnitz, Alemanha, 1998. p. 137–142.
- KRESSEL, U. Pairwise classification and support vector machines. In: MIT PRESS. *Advances in kernel methods*. [S.l.], 1999. p. 268.
- LORENA, A.; CARVALHO, A. Introdução as Máquinas de Vetores Suporte. *Relatório Técnico do Instituto de Ciências Matemáticas e de Computação (USP/Sao Carlos)*, v. 192, 2003.
- MULLER, K. et al. An introduction to kernel-based learning algorithms. *IEEE transactions on neural networks*, Citeseer, v. 12, n. 2, p. 181–201, 2001.
- OSUNA, E.; FREUND, R.; GIROSI, F. Support vector machines: Training and applications. *CBCL-144*, 1997.
- QUANG, A.; ZHANG, Q.; LI, X. Evolving support vector machine parameters. In: *Proceedings of the First International Conference on Machine Learning and Cybernetics*. [S.l.: s.n.], 2002. v. 4, p. 5.
- SCHÖLKOPF, B.; SMOLA, A. *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. [S.l.]: the MIT Press, 2002.
- SMOLA, A.; BARTLETT, P. *Advances in Large Margin Classifiers*. MIT Press Cambridge, MA, USA, 2000.

VAPNIK, V. *Statistical learning theory*. New York: John Wiley and Sons, 1998.

VAPNIK, V. *The nature of statistical learning theory*. Nova Iorque, Estados Unidos: Springer Verlag, 2000.