# Máquinas de vetores-suporte

## Índice

Nota: Parte deste material conta com a co-autoria de Clodoaldo A. de Moraes Lima e está baseado em resultados de pesquisa de LIMA (2004) e de SEMOLINI (2002).

# 1. Resgate de conceitos fundamentais de aprendizado de máquina

- O objetivo do aprendizado de máquina é construir modelos computacionais que podem se adaptar e aprender a partir da experiência (MITCHELL, 1997).
- Definição de aprendizado indutivo (MITCHELL, 1997): "Um programa de computador aprende a partir de um elenco de experiências *E*, relacionadas a uma classe de tarefas *T* e dispondo de uma medida de desempenho *M*, se seu desempenho medido por *M* junto à tarefa *T* melhora com o elenco de experiências *E*."
- No contexto de redes neurais artificiais, o seu processo de treinamento pode, então, ser caracterizado como aprendizado indutivo, sendo que o uso posterior da rede neural treinada para classificação, regressão ou agrupamento de dados é geralmente denominado de processo de inferência dedutiva. Exemplo: Induz-se um modelo computacional para um classificador e, em seguida, emprega-se este modelo para deduzir a classe de novas amostras ainda não rotuladas.

- Em termos práticos, algoritmos de aprendizado de máquina têm como objetivo descobrir o relacionamento entre as variáveis de um sistema (entrada/saída) a partir de dados amostrados (CHERKASSKY & MULIER, 2007).
- Sendo assim, eles não são necessários quando os relacionamentos entre todas as variáveis do problema (entrada/saída) são completamente compreendidos. Este definitivamente não é o caso de muitos dos problemas reais com os quais nos defrontamos em nosso dia-a-dia.
- Há três frentes principais em aprendizado de máquina:
    - ✓ Aprendizado supervisionado, que volta a ser o centro de atenção deste tópico do curso;
    - ✓ Aprendizado por reforço, que não chegou a ser abordado formalmente ainda;
    - ✓ Aprendizado não-supervisionado, abordado especificamente no Tópico 6 do curso e que voltará a ser tema de estudo no Tópico 8 do curso.

## 1.1 Aprendizado supervisionado

- São elementos básicos do aprendizado supervisionado:

  1. Dados de Treinamento:

     ✓ Exemplos de entrada/saída;

     ✓ Refletem o mapeamento funcional da entrada para a saída.

  2. Espaço de Hipóteses:

     ✓ Um conjunto particular ou uma classe de funções candidatas.

  3. Algoritmo de Aprendizado:

     ✓ Recebe os dados de treinamento como argumento de entrada;

     ✓ Seleciona a hipótese pertencente ao espaço de hipóteses definido.

- Em aprendizado supervisionado, busca-se a síntese de mapeamentos não-lineares com garantia de aproximação universal (funções contínuas definidas em um domínio fechado e limitado). Logo, há flexibilidade suficiente para reagir à demanda de cada aplicação.

- No entanto, permanece o desafio: Como calibrar devidamente a flexibilidade do modelo de aproximação?
- Um equívoco nesta calibragem pode conduzir a resultados de generalização ruins, seja porque o modelo é pouco flexível (conduzindo ao que se convencionou chamar de sub-ajuste do processo de treinamento), seja porque é flexível demais (podendo produzir o que se convencionou chamar de sobre-ajuste do processo de treinamento).

## 1.2 Modelos de aproximação lineares

- Principais propriedades:
  - ✓ Realizam uma soma ponderada dos atributos de entrada;
  - ✓ Supõem que os dados se relacionam linearmente.
- Espaço de Hipóteses:
  - ✓ Restrito (fortemente paramétrico): potencial de modelagem limitado, baixa flexibilidade.
  - ✓ Facilmente explorado na otimalidade.

- Como a flexibilidade é baixa, existem aplicações em que é válido explorar o espaço de hipóteses no limite, empregando técnicas de quadrados mínimos, por exemplo.
- Mesmo assim, há aplicações em que até a baixa flexibilidade dos modelos lineares pode conduzir a sobreajuste, indicando a necessidade de regularização do modelo.

## 1.3 Modelos de aproximação fundamentados em redes neurais

- Principais propriedades:
  - ✓ Muitas unidades de processamento simples, combinadas em camadas e com alta conectividade, produzindo modelos bastante flexíveis;
  - ✓ Exibem capacidade de aproximação universal.
- Espaço de hipóteses:
  - ✓ Pouco restrito (semi-paramétrico): pode modelar um amplo espectro de funções.
  - ✓ Pode não ser explorado na otimalidade, particularmente em função da presença de mínimos locais e/ou do alto custo computacional envolvido.

- Limitações:
  - ✓ Como a flexibilidade é alta, sempre vigora a necessidade de regularização do modelo e a necessidade de lidar com espaços de hipóteses muito amplos;
  - ✓ Requer bem mais dados de treinamento que abordagens lineares;
  - ✓ Geralmente envolve mecanismos de ajuste de pesos sinápticos via otimização não-linear e/ou auto-organização, sujeitos a mínimos locais;
  - ✓ Geralmente envolve um grande número de parâmetros ajustáveis.
- As máquinas de vetores-suporte, fundamentadas na teoria do aprendizado estatístico, dispõem de mecanismos voltados para a superação ou atenuação dessas limitações. Por essa razão, elas são especialmente indicadas em cenários em que uma ou mais das condições a seguir se manifestam:
  - ✓ Baixa amostragem;
  - ✓ Altos níveis de ruídos nos dados;
  - ✓ Dados de elevada dimensão (muitas entradas).

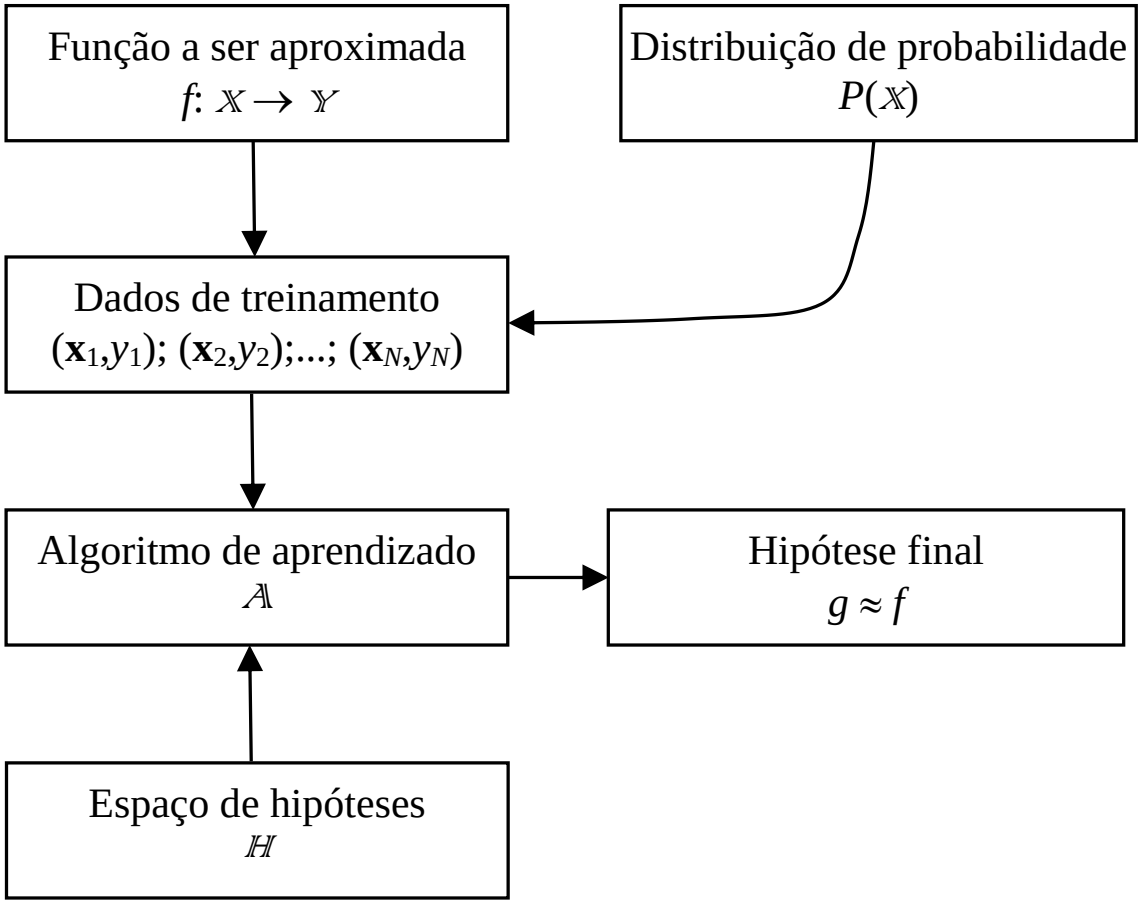



Figura 1 – Conceitos fundamentais em aprendizado supervisionado (baseado em ABU-MOSTAFA et al. (2012))

## 1.4 Regressão de quadrados mínimos

- Considere que você tenha à disposição um conjunto de *N* amostras de treinamento na forma: $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, onde $\mathbf{x}_i \in \Re^n$, $i$=1,...,$N$. Suponha também que $N > n$.
- A regressão de quadrados mínimos busca um vetor $\mathbf{w} \in \Re^n$ que minimiza: $J(\mathbf{w}) = \sum_{i=1}^{N} (\mathbf{x}_i^T \mathbf{w} - y_i)^2$.
- Fazendo $X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \ddots & x_{2n} \\ \vdots & \vdots & & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{Nn} \end{bmatrix}$ e $\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$, constata-se que a regressão de quadrados mínimos requer a solução de um sistema linear sobredeterminado, na forma (geralmente, acrescenta-se uma coluna de 1's à matriz $X$):

$$\min_{\mathbf{w}} \|X\mathbf{w} - \mathbf{y}\|_2^2 \ .$$

- Aplicando as condições necessárias de otimalidade, a solução para este problema de otimização fica:

$$\mathbf{w} = \left( X^T X \right)^{-1} X^T \mathbf{y}.$$

## 1.5 Ridge regression

- Tomando o mesmo cenário da Seção 1.4, é possível adicionar um termo de regularização, que penaliza o crescimento da norma do vetor **w**, produzindo o problema regularizado de regressão de quadrados mínimos, também denominado de *ridge regression*, na forma:

$$\min_{\mathbf{w}} \left\| X\mathbf{w} - \mathbf{y} \right\|_2^2 + \lambda \left\| \mathbf{w} \right\|_2^2, \ \lambda \geq 0.$$

- Aplicando a condição necessária de otimalidade, como feito na Seção 1.4, obtém-se como solução ótima:

$$\mathbf{w} = \left( X^T X + \lambda I \right)^{-1} X^T \mathbf{y}.$$

- A definição de um valor para o parâmetro de regularização $\lambda \geq 0$ pode ser feita por técnicas de validação cruzada (ARLOT & CELISSE, 2010).
- O modelo regularizado de regressão linear assume então a forma:

$$\mathbf{w}^T \mathbf{x} = \mathbf{y}^T X \left( X^T X + \lambda I \right)^{-1} \mathbf{x}.$$

- Podemos agora usar a equivalência $X \left( X^T X + \lambda I \right)^{-1} = \left( X X^T + \lambda I \right)^{-1} X$ de modo a obter o modelo de regressão em sua forma dual:

$$\mathbf{w}^T \mathbf{x} = \mathbf{y}^T \left( X X^T + \lambda I \right)^{-1} X \mathbf{x}$$

- Já sabemos que, para $N > n$, a forma dual não se aplica. Mas esta forma dual pode ser reapresentada de modo a conduzir a um importante resultado que sustenta a regressão por função *kernel*.

## 1.6 Regressão por função kernel

- Tomando o modelo de regressão em sua forma dual da Seção 1.5 e fazendo:

$$K = \begin{bmatrix} \mathbf{x}_1^T\mathbf{x}_1 & \mathbf{x}_1^T\mathbf{x}_2 & \cdots & \mathbf{x}_1^T\mathbf{x}_N \\ \mathbf{x}_2^T\mathbf{x}_1 & \mathbf{x}_2^T\mathbf{x}_2 & \ddots & \mathbf{x}_2^T\mathbf{x}_N \\ \vdots & \vdots & & \vdots \\ \mathbf{x}_N^T\mathbf{x}_1 & \mathbf{x}_N^T\mathbf{x}_2 & \cdots & \mathbf{x}_N^T\mathbf{x}_N \end{bmatrix} \text{ e } \mathbf{k}(\mathbf{x}) = \begin{bmatrix} \mathbf{x}_1^T\mathbf{x} \\ \mathbf{x}_2^T\mathbf{x} \\ \vdots \\ \mathbf{x}_N^T\mathbf{x} \end{bmatrix},$$

obtém-se:

$$\mathbf{w}^T\mathbf{x} = \mathbf{y}^T\left(K + \lambda I\right)^{-1}\mathbf{k}(\mathbf{x}).$$

- Repare que $\mathbf{x}_i \in \Re^n$, $i$=1,...,$N$ e também $\mathbf{x} \in \Re^n$ aparecem sempre aos pares, em produtos internos.
- Suponha agora que exista um mapeamento não-linear $\Phi : \Re^n \to S$, onde $S$ é um espaço vetorial equipado com um produto interno, geralmente denominado de espaço de características, cuja dimensão pode até ser infinita.
- Definindo o produto interno no espaço de características como:

$$\Phi(\mathbf{x}_i)^T\,\Phi(\mathbf{x}_j) = \kappa(\mathbf{x}_i, \mathbf{x}_j),$$

é possível realizar *ridge regression* no espaço de características caso a função $\kappa : \Re^n \times \Re^n \to \Re$, denominada função *kernel*, seja semidefinida positiva, o que, por definição, garante que a matriz simétrica a seguir seja semidefinida positiva:

$$K = \begin{bmatrix} \kappa(\mathbf{x}_1, \mathbf{x}_1) & \kappa(\mathbf{x}_1, \mathbf{x}_2) & \cdots & \kappa(\mathbf{x}_1, \mathbf{x}_N) \\ \kappa(\mathbf{x}_2, \mathbf{x}_1) & \kappa(\mathbf{x}_2, \mathbf{x}_2) & \ddots & \kappa(\mathbf{x}_2, \mathbf{x}_N) \\ \vdots & \vdots & & \vdots \\ \kappa(\mathbf{x}_N, \mathbf{x}_1) & \kappa(\mathbf{x}_N, \mathbf{x}_2) & \cdots & \kappa(\mathbf{x}_N, \mathbf{x}_N) \end{bmatrix}.$$

- Basta fazer também:

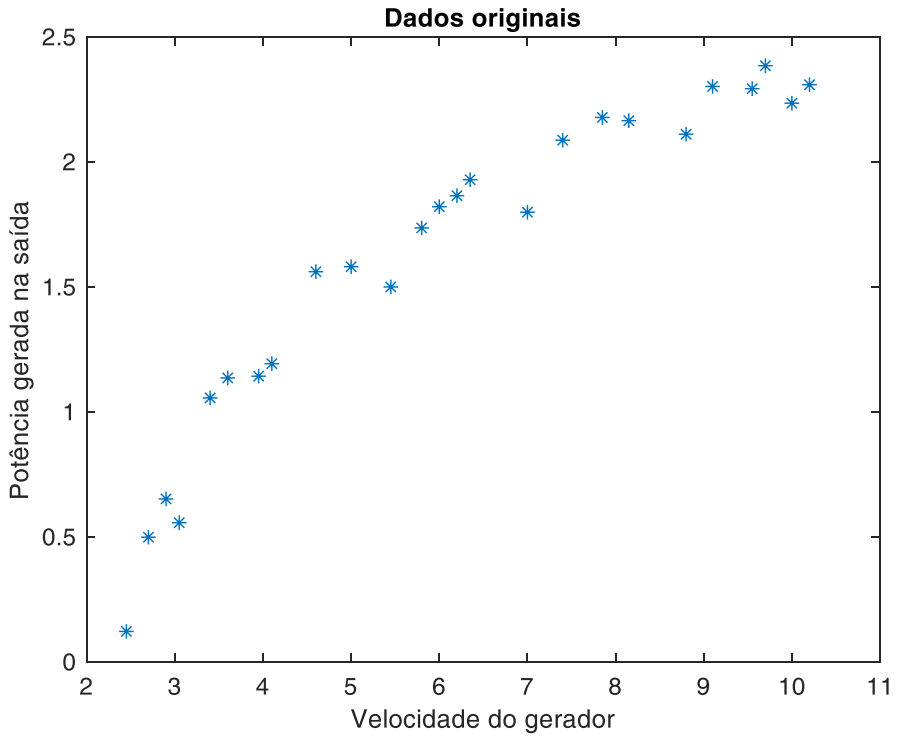
$$\mathbf{k}(\mathbf{x}) = \begin{bmatrix} \kappa(\mathbf{x}_1, \mathbf{x}) \\ \kappa(\mathbf{x}_2, \mathbf{x}) \\ \vdots \\ \kappa(\mathbf{x}_N, \mathbf{x}) \end{bmatrix},$$

e o modelo que faz *ridge regression* no espaço de características toma a forma:

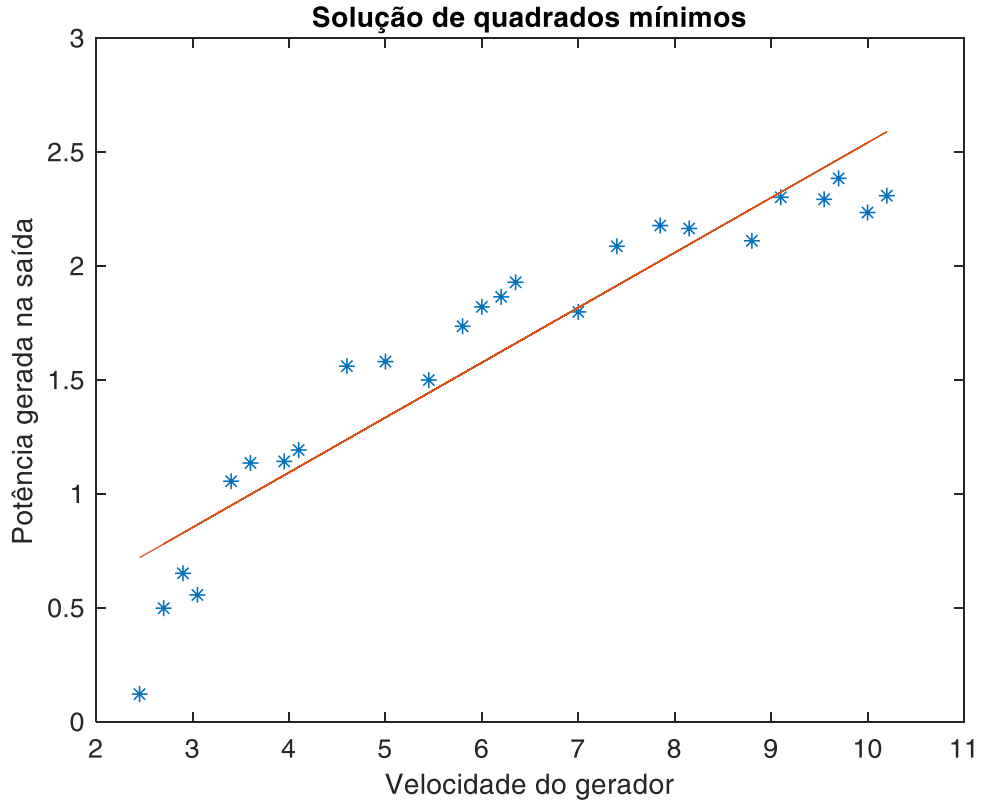
$$RR(\mathbf{x}) = \mathbf{y}^T \left(K + \lambda I\right)^{-1} \mathbf{k}(\mathbf{x}).$$

- Este último resultado representa um salto de qualidade em regressão não-linear, pois ganha-se a flexibilidade do modelo de regressão, mas ainda requerendo a solução de um problema de quadrados mínimos.
- O modelo, no espaço de características, é sempre um hiperplano, mas ao ser trazido de volta ao espaço original dos dados, torna-se uma superfície não-linear.
- Para que seja bem-definida esta operação de ida ao espaço de características e volta ao espaço original dos dados, é necessário que a função *kernel* atenda à condição de Mercer (*http://www.svms.org/mercer/*), que basicamente garante que a matriz $K$ seja semidefinida positiva. Outro ponto a se destacar é o chamado truque do *kernel* (SCHÖLKOPF & SMOLA, 2001), que basicamente torna implícitos os mapeamentos de ida para e volta do espaço de características, pois em lugar de se fazer um produto interno no espaço de características, basta aplicar a função *kernel* no espaço original dos dados. Um desafio importante está na escolha adequada da função *kernel*. Iremos voltar a este tema ainda neste tópico do curso.

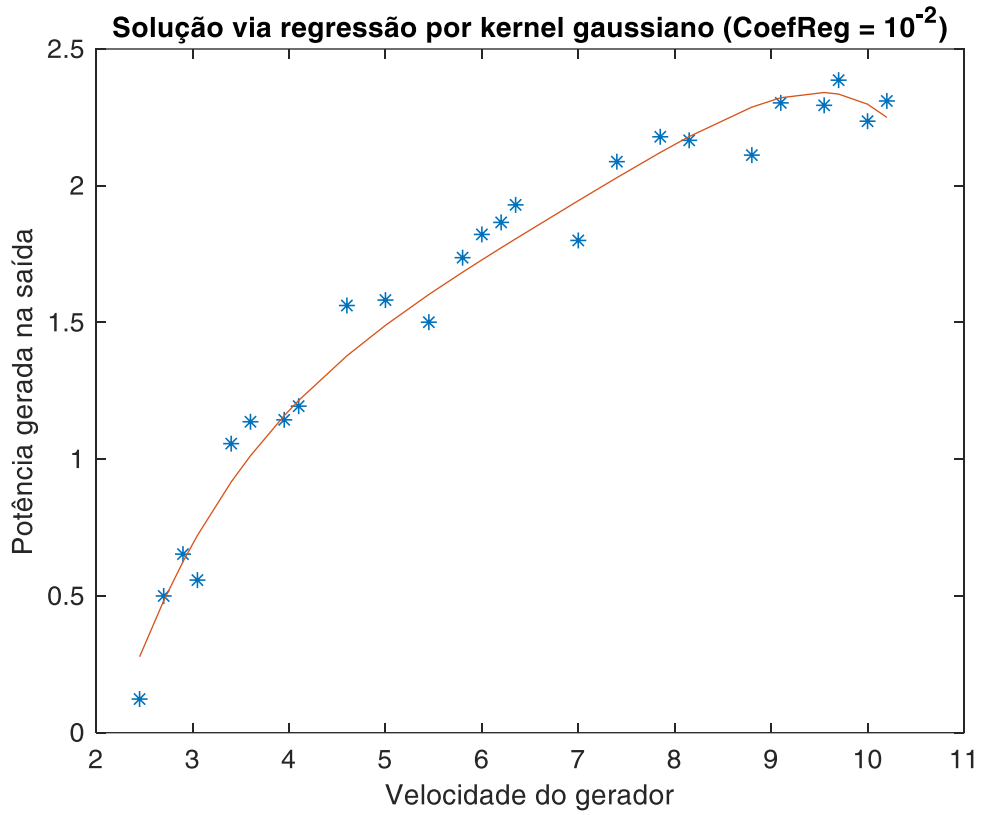
- Caso de estudo (MONTGOMERY et al., 2012):



- A regressão de quadrados mínimos produziu o seguinte modelo: $\hat{y} = 0.1308 + 0.2411x$.
- Lembre-se da necessidade de acrescentar uma coluna de 1's à matriz *X* para fornecer o *offset* do modelo linear.



- Usando um *kernel* gaussiano com desvio padrão = 4 e aplicando um coeficiente de regularização = $10^{-2}$, obtém-se o seguinte resultado:



# 2. Fundamentos da teoria do aprendizado estatístico

- A Teoria do Aprendizado Estatístico (TAE) foi proposta por VAPNIK (1982), VAPNIK (1998) e VAPNIK (1999b). Sugere-se consultar VAPNIK (1999a).
- A TAE investiga formas de estimação de dependências funcionais com base em conjuntos de dados.
- Diferente dos métodos de inferência estatística clássica (CASELLA & BERGER, 2002), como teste de hipóteses, máxima verossimilhança e quadrados mínimos, supõem-se aqui amostras pequenas e nenhuma informação a priori acerca do problema a ser resolvido.
- A TAE permite analisar formalmente o comportamento e as limitações inerentes a algoritmos de aprendizado de máquina.

## 3. Risco esperado × Risco empírico

- Toda a formalização estará vinculada ao problema de classificação binária.
- Considere uma amostra $\Psi$ de tamanho $N$

$$\Psi = \{\mathbf{z}_1, \cdots, \mathbf{z}_N\}$$

em que $\mathbf{z}_i \in Z$, $i$=1,...,$N$, são elementos i.i.d. (independentes e identicamente distribuídos), ou seja, todos eles seguem a mesma distribuição de probabilidade $P(Z)$ e são mutuamente independentes.

- Seja $\mathbf{x}_i \in \Re^m$ o vetor com as primeiras $m$ coordenadas de $\mathbf{z}_i$ e $y_i \in \{-1,+1\}$ a ($m$+1)-ésima coordenada de $\mathbf{z}_i$.
- O problema de classificação binária corresponde a encontrar uma função que aproxime a dependência observada entre os valores de $\mathbf{x}_i$ e $y_i$ com base na amostra $\Psi$.

- Seja $F(\mathbf{x},\alpha)$ um conjunto de funções, em que $\alpha$ especifica uma única função em um universo $\Lambda$. Então o problema de classificação binária requer a minimização do funcional:

$$I(\alpha) = \int Q(\mathbf{z},\alpha)P(\mathbf{z})d\mathbf{z} \qquad (1)$$

em que $Q(\mathbf{z},\alpha)$ é uma função que define a perda da aproximação de *y* por $F(\mathbf{x},\alpha)$.

- Uma possível escolha para $Q(\mathbf{z},\alpha)$ seria:

$$Q(\mathbf{z},\alpha) = [y - F(\mathbf{x},\alpha)]^2 \qquad (2)$$

- A Eq. (1) é conhecida como **risco esperado**. Ela pode ser interpretada como o valor médio da função de perda tomado sobre todo o universo de instâncias de entrada **z**.
- Duas observações são fundamentais aqui:
  - ✓ A distribuição de probabilidade *P*(*Z*) é desconhecida, impossibilitando a minimização direta do risco esperado. A única informação disponível é a amostra $\Psi$.

- ✓ Por ser um problema de classificação binária, $y$ e $F(\mathbf{x},\alpha)$, para qualquer $\alpha$, só podem assumir os valores 0 e 1. Logo, $Q(\mathbf{z},\alpha)$ será uma função indicadora, também assumindo apenas os valores 0 e 1.

- Como viabilizar a minimização da Eq. (1)?
- Uma possibilidade seria realizar uma estimativa da distribuição de probabilidade $P(Z)$, produzindo $\hat{P}(Z)$. Mas, para tanto, seria necessário partir de suposições fortes acerca da forma paramétrica desta distribuição.
- Uma estratégia que não requer conhecimento a priori é adotar uma aproximação do risco esperado com base no **risco empírico**, dado na forma:

$$I_{emp}(\alpha) = \frac{1}{N}\sum_{i=1}^{N} Q(\mathbf{z}_i,\alpha) \qquad (3)$$

em que $\mathbf{z}_i \in \Psi$, $i$=1,...,$N$, é a amostra dada.

- Sob que condições a aproximação $I_{emp}(\alpha)$ de $I(\alpha)$ é válida?

- Na verdade, não há a necessidade desses dois funcionais serem próximos para todo $\alpha$, mas deve-se garantir que o mínimo do risco empírico $I_{emp}(\alpha)$ se aproxime do mínimo do risco esperado $I(\alpha)$. Ou seja, o $\alpha$ que minimiza $I_{emp}(\alpha)$ deve estar próximo do $\alpha$ que minimiza $I(\alpha)$. No entanto, não se sabe a priori onde este $\alpha$ se encontra, de modo que se sugere buscar a aproximação desses dois funcionais para todo $\alpha$.
- Uma candidata seria a métrica $L_2$, dada na forma:

$$\rho_{L_2}(f_1(\mathbf{x}), f_2(\mathbf{x})) = \left\{ \int [f_1(\mathbf{x}) - f_2(\mathbf{x})]^2 d\mathbf{x} \right\}^{1/2} \quad (4)$$

- Outra candidata seria a métrica $L_\infty$, dada na forma:

$$\rho_{L_\infty}(f_1(\mathbf{x}), f_2(\mathbf{x})) = \sup_{\mathbf{x}} |f_1(\mathbf{x}) - f_2(\mathbf{x})| \quad (5)$$

- Note que a métrica $\rho_{L_\infty}(f_1(\mathbf{x}), f_2(\mathbf{x}))$ é mais restrita que a métrica $\rho_{L_2}(f_1(\mathbf{x}), f_2(\mathbf{x}))$, pois duas funções próximas segundo a Eq. (5) são sempre próximas segundo a Eq. (4), mas a recíproca pode não valer. E mais ainda, quando $\rho_{L_\infty}(f_1(\mathbf{x}), f_2(\mathbf{x}))$ tende a zero, é garantido que o mínimo de $f_1(\mathbf{x})$ coincide com o mínimo de $f_2(\mathbf{x})$.

- Já quando $\rho_{L_2}\left(f_1(\mathbf{x}), f_2(\mathbf{x})\right)$ tende a zero, não há convergência uniforme, ou seja, não é possível garantir que o mínimo de $f_1(\mathbf{x})$ coincide com o mínimo de $f_2(\mathbf{x})$, conforme ilustrado na Figura 2 a seguir, numa situação arbitrária, mas possível.

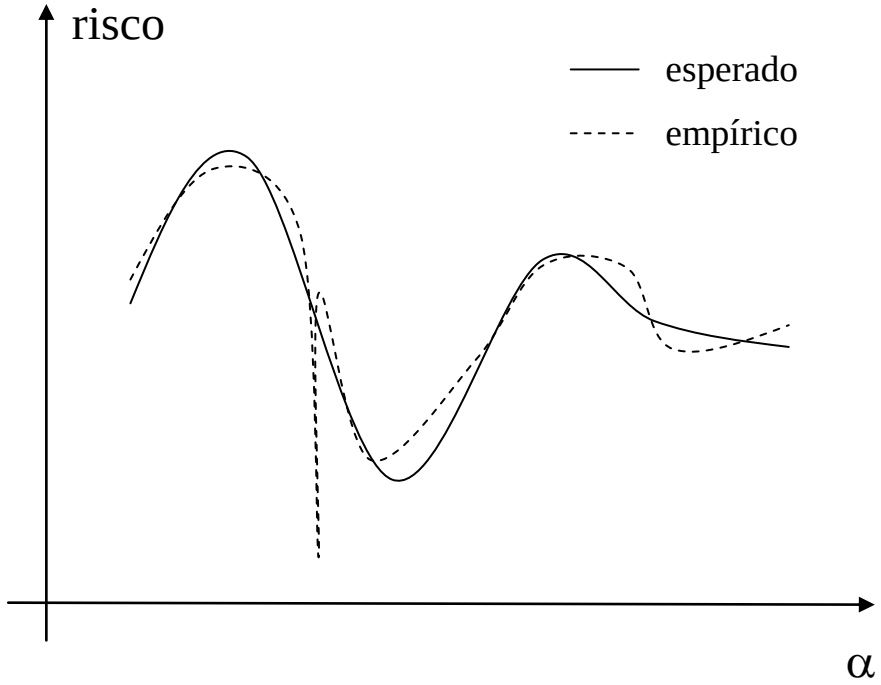



Figura 2 – Situação possível em que o mínimo do risco empírico difere significativamente do mínimo do risco esperado.

- Na Figura 2, $\rho_{L_2}\left(f_1(\mathbf{x}), f_2(\mathbf{x})\right)$ assume um valor reduzido quando comparado ao valor assumido por $\rho_{L_\infty}\left(f_1(\mathbf{x}), f_2(\mathbf{x})\right)$.
- Para evitar situações como a ilustrada na Figura 2, deve-se exigir que $I_{emp}(\alpha)$ não se distancie de $I(\alpha)$ mais que um valor dado $\delta$, para todo $\alpha$, produzindo:

$$\rho_{L_\infty}\left(I_{emp}(\alpha), I(\alpha)\right) < \delta, \quad \forall \alpha \in \Lambda \tag{6}$$

- Sendo assim, valendo a relação:

$$P\left[\sup_{\alpha} \left| I_{emp}(\alpha) - I(\alpha) \right| > \delta\right] \xrightarrow[N\to\infty]{} 0 \tag{7}$$

  Diz-se que o risco empírico converge uniformemente para o risco esperado.
- Em TAE, esta convergência uniforme é tida como sinônimo de aprendizado, de modo que a análise das condições sob as quais essa convergência ocorre permite obter garantias acerca do desempenho de algoritmos de aprendizado de máquina.

# 4. Aprendizagem em um conjunto finito de funções

- Considere o caso em que o conjunto de funções (modelos / hipóteses) é finito:

$$\Im = \{F(\mathbf{x}, \alpha_1), F(\mathbf{x}, \alpha_2), \cdots, F(\mathbf{x}, \alpha_l)\} \tag{8}$$

  onde $l$ é a quantidade de funções candidatas. Trata-se, portanto, de um espaço de hipóteses com um número $l$ de elementos.

- Sendo assim, o conjunto $\Im$ contém algumas hipóteses possíveis, que podem resultar do processo de aprendizado a partir dos dados disponíveis.
- Cada hipótese terá associada a si um valor de erro empírico e deseja-se chegar à hipótese, ou equivalentemente ao valor de $\alpha$, que apresente o menor erro esperado.
- É possível provar que, com grau de confiança $1 - \eta$ e para uma amostra de tamanho $N$, vale a desigualdade:

$$I(\alpha_i) \leq I_{emp}(\alpha_i) + \sqrt{\frac{\ln l - \ln(\eta/2)}{2N}} \tag{9}$$

- Como $l$ e $\eta$ são constantes, então:

$$\lim_{N\to\infty}\sqrt{\frac{\ln l-\ln(\eta/2)}{2N}}=0 \qquad (10)$$

- Conclusão: Sempre há convergência uniforme quando se considera um conjunto finito de funções.

## 5. Aprendizagem em um conjunto infinito de funções

- Quando $l\to\infty$, cria-se uma dificuldade. E este é o caso real em aprendizado de máquina.
- Como obter resultados válidos para $l\to\infty$?
- A solução está ao observar que, na amostra $\Psi$, $N$ é finito, ou seja, há um número finito de classificações binárias possíveis para os elementos de $\Psi$.

## 5.1 Dimensão Vapnik-Chervonenkis (Dimensão VC)

- Em linhas gerais e em termos práticos, a dimensão Vapnik-Chervonenkis (VC) de um espaço de hipóteses $\mathbb{H}$ pode ser interpretada como o número **efetivo** de parâmetros de modelos pertencentes a este espaço de hipóteses $\mathbb{H}$, sendo denotada por $h = d_{VC}(\mathbb{H})$.
- Outro resultado prático muito relevante, o qual será apresentado mais adiante, é que quando $h = d_{VC}(\mathbb{H})$ é finita, então modelos pertencentes a $\mathbb{H}$ podem generalizar, no sentido de que, tomando-se um conjunto de dados $N$, com $N$ suficientemente elevado, o risco empírico tende ao risco esperado tanto quanto se queira.
- Formalmente, $h = d_{VC}(\mathbb{H})$ está associada ao maior número de pontos que um modelo pertencente a $\mathbb{H}$ pode classificar corretamente, para toda proposta de rotulação binária possível desses pontos. Diz-se então que $\mathbb{H}$ **particiona** esses pontos.
- Sendo assim, quando se tem $d_{VC}(\mathbb{H}) \geq Q$, então existe algum conjunto de $Q$ pontos que é particionado por $\mathbb{H}$. Por outro lado, quando $d_{VC}(\mathbb{H}) < Q$, então é possível afirmar que não existe nenhum conjunto de $Q$ pontos que pode ser particionado por $\mathbb{H}$.

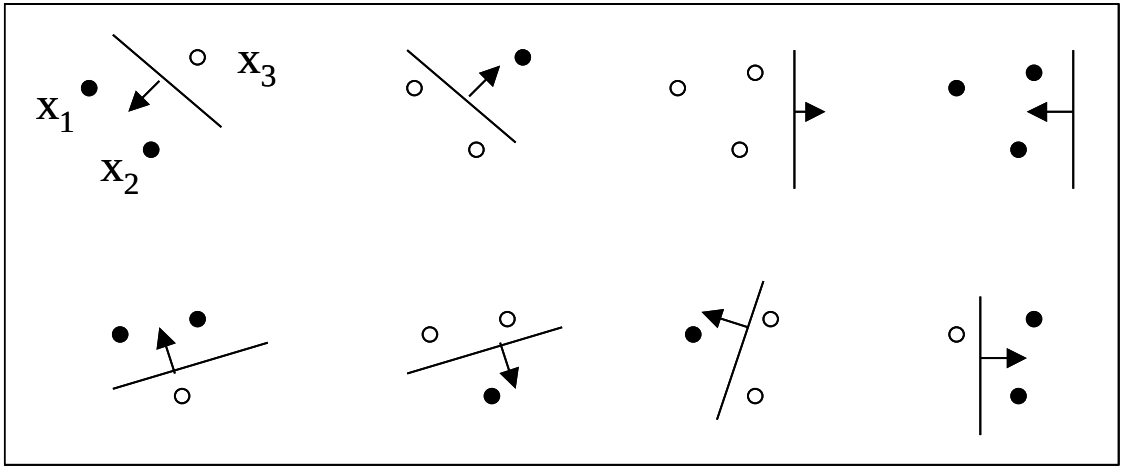



Figura 3 – Possibilidades de rotulação de três amostras no $\Re^2$ e a classificação realizada por uma função linear (LIMA, 2004).

- Considerando um neurônio do tipo McCulloch & Pitts, com duas entradas, então a Figura 3 mostra que a dimensão VC deste modelo de classificação binária é $h = 3$, pois é possível particionar os 3 pontos fornecidos, para qualquer proposta de rotulação binária desses pontos.
- Generalizando, é possível provar que a dimensão VC de um neurônio do tipo McCulloch & Pitts com $m$ entradas vai ser $h = m + 1$.

- De fato, o problema do OU-exclusivo já indicava a impossibilidade de um neurônio do tipo McCulloch & Pitts resolver um problema de classificação binária com 4 pontos, dada a rotulação específica desses pontos no problema do OU-exclusivo.

## 5.2 Dimensão Vapnik-Chervonenkis e aprendizado

- Supondo que dois modelos de classificação são equivalentes quando eles produzem a mesma classificação para a amostra $\Psi$, há então um número finito de classes de equivalência, mesmo que o número de funções seja infinito.
- Repare que este número finito de classes de equivalência depende tanto do conjunto de funções (hipóteses / modelos) quanto da amostra $\Psi$.
- Definição: A dimensão VC de um conjunto de funções indicadoras $Q(\mathbf{z},\alpha)$, $\alpha\in\Lambda$, é igual a um número *h* de elementos da amostra $\Psi$ que podem ser separados em duas classes diferentes, considerando todas as $2^h$ maneiras possíveis, usando este conjunto de funções.

- É possível provar que, com grau de confiança $1 - \eta$ e para uma amostra de tamanho $N$, vale a desigualdade:

$$I(\alpha_i) \leq I_{emp}(\alpha_i) + \frac{\varepsilon}{2}\left(1 + \sqrt{1 + \frac{4 I_{emp}(\alpha_i)}{\varepsilon}}\right) \tag{11}$$

onde $\varepsilon = b_1 \dfrac{\left[\ln\left(b_2 N / h\right) + 1\right] - \ln\left(\eta / 4\right)}{N / h}$. No pior caso, toma-se $b_1 = 4$ e $b_2 = 2$.

- Assim, o limitante para o risco esperado é composto por dois fatores: o risco empírico e um termo que depende da razão entre $N$ e $h$.
- Repare que $\varepsilon$ cresce rápido quando se aumenta o grau de confiança. De fato, quando $\eta \to 0$ tem-se que $\varepsilon \to \infty$, tornando o limitante sem efeito.
- Há uma interpretação intuitiva para isso: Qualquer função (hipótese / modelo) obtida a partir de uma amostra de tamanho finito não pode atender a um grau de confiança arbitrariamente alto.

- Logo, há sempre um *trade-off* entre a acurácia fornecida pelo limitante e o grau de confiança que se atribui a esse limitante.
- Uma consequência imediata do resultado da Eq. (11) é que, para *h* (dimensão VC) finito, um aumento no tamanho da amostra faz com que o risco empírico se aproxime do risco esperado.
- VAPNIK (1999b) percebeu então que é possível estabelecer uma relação entre o grau de confiança no limitante e o número de amostras. A seguinte regra empírica foi proposta:

$$\eta = \min\left(\frac{4}{\sqrt{N}}, 1\right).$$

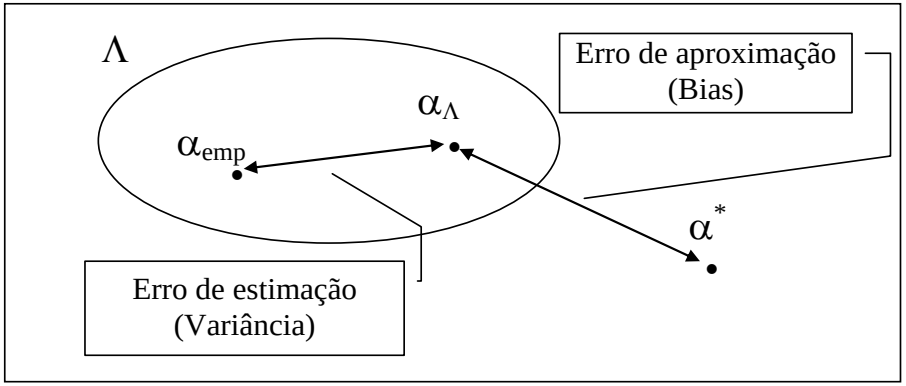
## 5.3 Minimização do risco estrutural

- Com base no resultado apresentado na Eq. (11), constata-se que a minimização do risco empírico funciona bem quando a razão $N/h$ assume valores elevados, pois isso reduz bastante o valor de ε.

- Quando esse não é o caso (por exemplo $N/h < 20$), então os dois termos na Eq. (11) precisam ser minimizados. Para tanto, VAPNIK (1982) propôs o princípio da minimização do risco estrutural.
- O primeiro termo da Eq. (11) depende de uma hipótese particular, dado o espaço de hipóteses, enquanto que o segundo termo da Eq. (11) depende da dimensão VC do espaço de hipóteses.
- Ou seja, minimizar o risco estrutural equivale a transformar a dimensão VC numa variável controlável, buscando **uma complexidade (flexibilidade) ótima do modelo para o tamanho da amostra**.
- Ao minimizar o risco estrutural, obtém-se um resultado mais próximo do risco esperado, o que implica na maximização da capacidade de generalização.
- Note que possuir dimensão VC finita é condição necessária para a convergência uniforme de $I_{emp}(\alpha_i)$ para $I(\alpha_i)$ quando $N$ cresce.
- Repare que não há uma dependência explícita da dimensão $m$ dos dados de entrada.

## 5.4 Comentários finais

- Pode-se afirmar que um algoritmo de aprendizado generaliza se este garante que o modelo estimado, a partir da amostra fornecida, se aproxima do modelo esperado, que seria o modelo obtido caso houvesse informação completa sobre a distribuição de probabilidade da amostra.




- Generalizar bem, então, equivale a minimizar o erro de estimação.
- Manter a dimensão VC finita, mas permitindo que ela cresça o suficiente, equivale a minimizar o erro de aproximação.

- Isso é o que se busca obter com máquinas de vetores-suporte (BURGES, 1998; CORTES & VAPNIK, 1995; HAYKIN, 2008; SCHÖLKOPF & SMOLA, 2001).

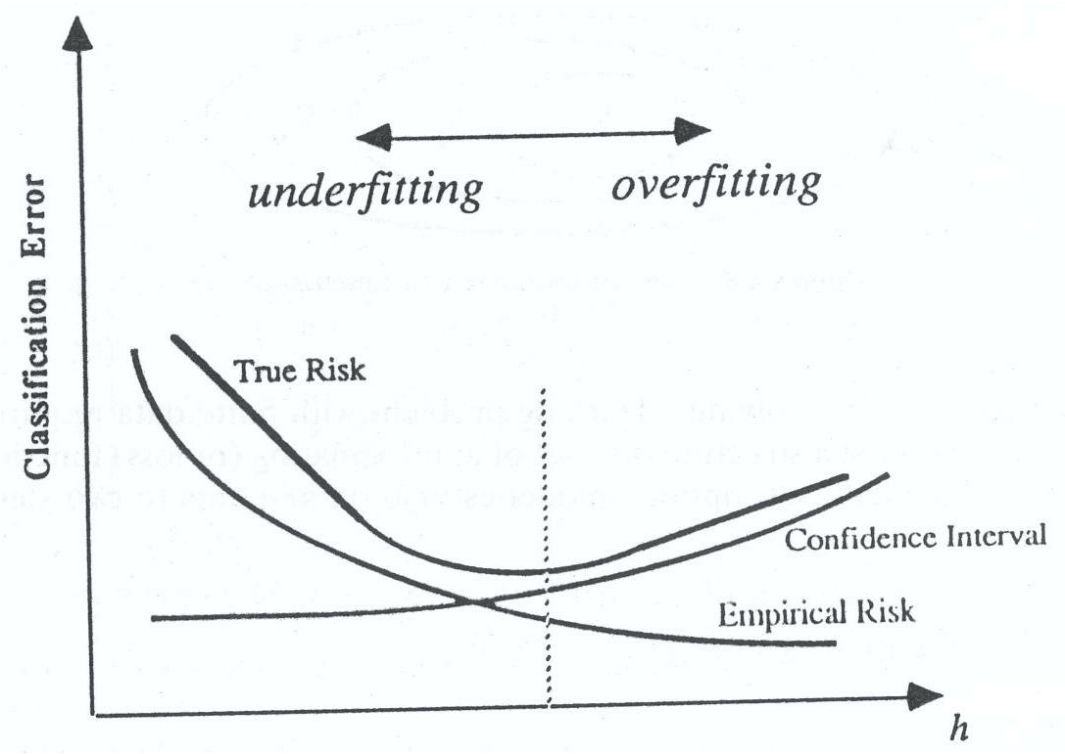



Figura 4 – Limitante superior para o risco esperado e risco empírico em função de *h*, onde *h* é a dimensão VC.

# 6. Máquinas de vetores-suporte

- Máquinas de vetores-suporte (SVM, do inglês *Support Vector Machines*) derivam de uma técnica de aprendizado de máquina fundamentada nos princípios indutivos da Minimização do Risco Estrutural. Estes princípios são provenientes da Teoria do Aprendizado Estatístico, a qual está baseada no fato de que o erro da técnica de aprendizagem junto aos dados de validação (erro de generalização) é limitado pelo erro de treinamento mais um termo que depende da dimensão VC.
- SVM implementa um mapeamento não-linear (executado por um produto interno kernel escolhido a priori) dos dados de entrada para um espaço de características (*feature space*) de alta dimensão, em que um hiperplano ótimo é construído para separar os dados linearmente em duas classes.
- Quando os dados de treinamento são separáveis, o hiperplano ótimo no espaço de características é aquele que apresenta a máxima margem de separação $\rho$ (ver Figura 5).

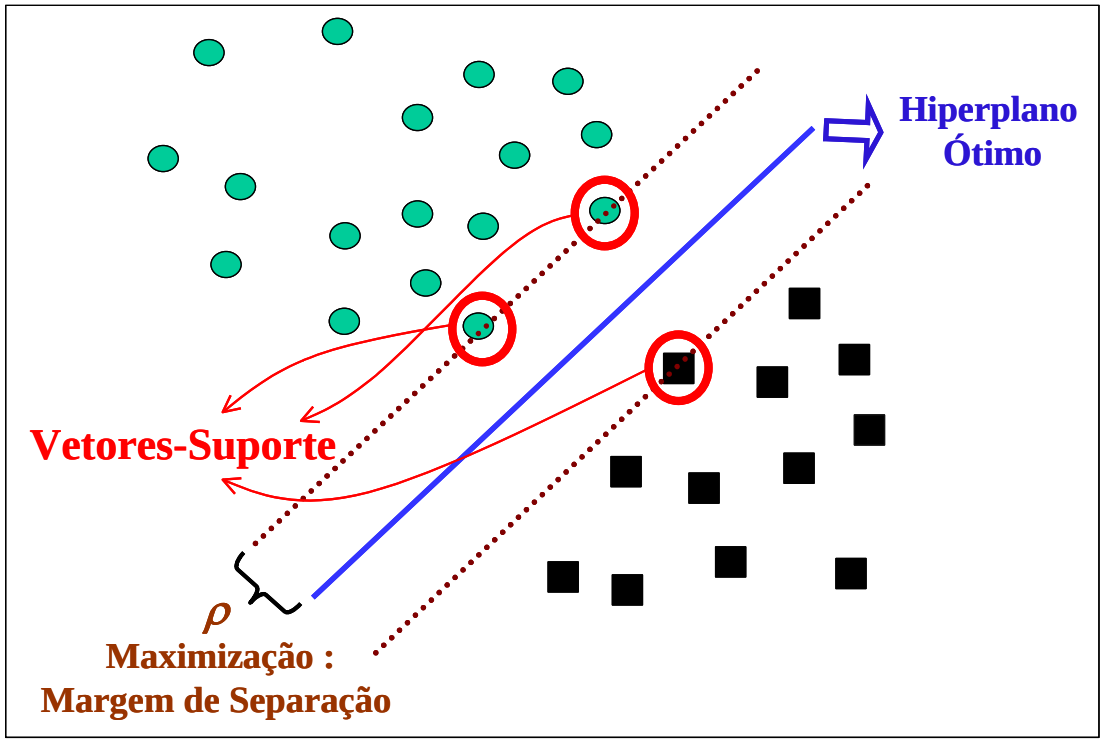



Figura 5 – Hiperplano de máxima margem

- Para dados de treinamento em que as amostras das diversas classes apresentam superposição (dados não separáveis), uma generalização deste conceito é utilizada.

## 6.1 Hiperplano ótimo para classes linearmente separáveis

- Quando o aprendizado supervisionado é aplicado ao problema de classificação, as amostras de treinamento são formadas pelo conjunto de dados de entrada associados às suas correspondentes respostas pré-classificadas (rótulos ou dados de saída). Após o treinamento, o objetivo é classificar novas amostras, ainda não rotuladas.
- Considere o seguinte conjunto de dados de treinamento:

$$( \mathbf{x}_i , y_i )_{\,1 \le i \le N} \ , \quad \mathbf{x} \in R^m, \quad y \in \{+1, -1\}$$

  onde $\mathbf{x}_i$ é o dado de entrada para a amostra $i$ e $y_i$ é a correspondente resposta desejada.
- Classificações binárias são frequentemente realizadas pelo uso de funções $g : X \subseteq R^m \rightarrow R$ com a seguinte estratégia: as amostras são designadas para a classe positiva, se $g(x) \geq 0$, e, caso contrário, para a classe negativa.
- Será considerado nesta seção que as classes representadas pelos rótulos $y_i = +1$ e $-1$ são linearmente separáveis. A superfície de decisão será representada por um hiperplano na forma:

$$g(\mathbf{x}) = ( \mathbf{w}^T\mathbf{x} ) + b = 0 \qquad (12)$$

onde $\mathbf{w} \in R^m$ é o vetor de pesos, e $b \in R$ é o intercepto.

- Assim, pode-se aplicar a seguinte **regra de decisão**:

$$( \mathbf{w}^T\mathbf{x}_i ) + b \geq 0 \qquad \text{para } y_i = +1$$

$$( \mathbf{w}^T\mathbf{x}_i ) + b < 0 \qquad \text{para } y_i = -1$$

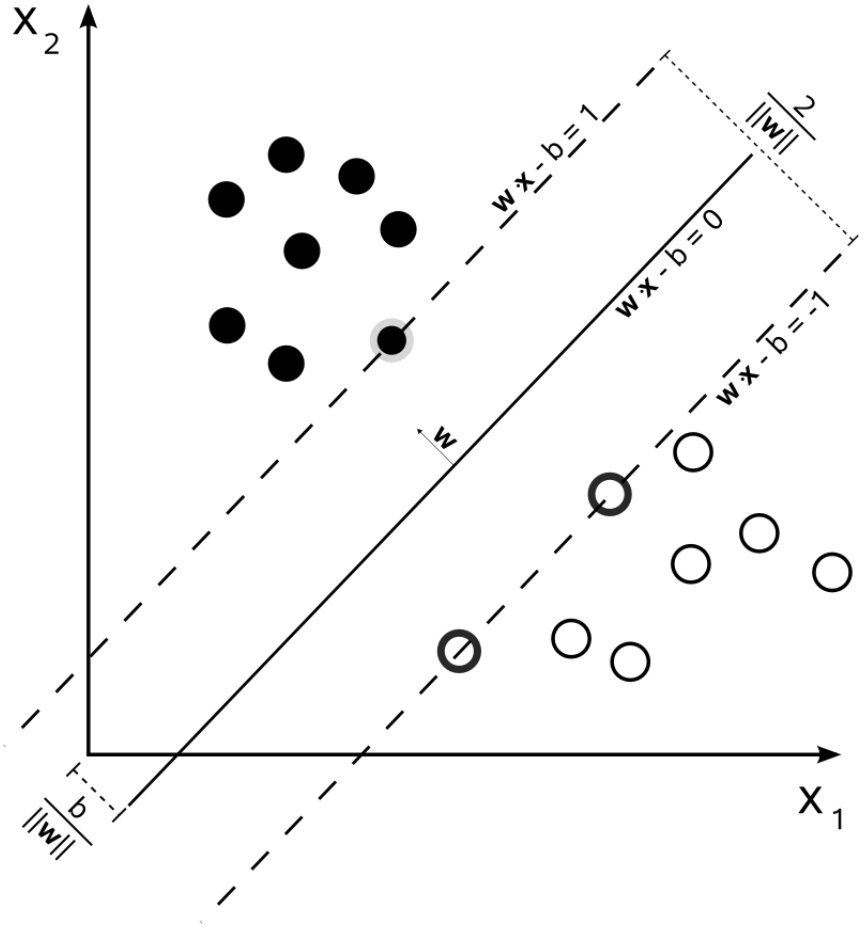
- Para descrever o lugar geométrico dos hiperplanos separadores, será utilizada a seguinte forma canônica (onde o vetor **w** e o escalar *b* são re-escalados de tal maneira a atender as desigualdades):

$$( \mathbf{w}^T\mathbf{x}_i ) + b \geq +1 \qquad \text{para } y_i = +1 \qquad (13)$$

$$( \mathbf{w}^T\mathbf{x}_i ) + b \leq -1 \qquad \text{para } y_i = -1$$

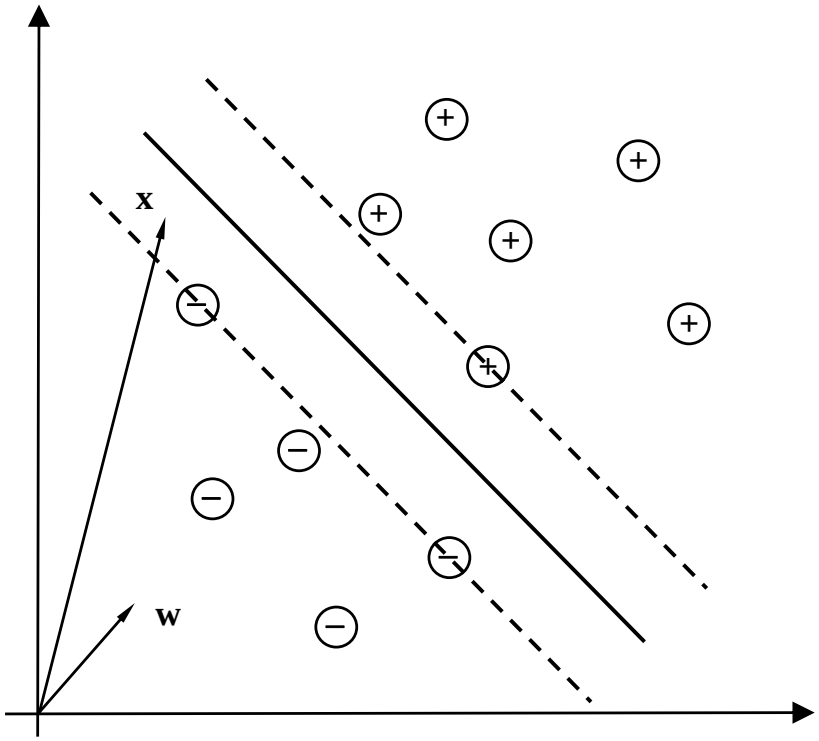
- A seguir, é apresentada a notação compacta para as desigualdades em (13):

$$y_i [( \mathbf{w}^T\mathbf{x}_i) + b] \geq 1 \qquad (14)$$




**Nota**: O *b* desta figura corresponde a –*b* da formulação que está sendo adotada.

Figura 6 – A margem é função do módulo do vetor normal ao hiperplano




$$( \mathbf{w}^T \mathbf{x}_i ) + b \geq 0 \qquad \text{para } y_i = +1$$

$$( \mathbf{w}^T \mathbf{x}_i ) + b < 0 \qquad \text{para } y_i = -1$$

Figura 7 – Interpretação geométrica da regra de decisão.

- Para um dado vetor de pesos **w** e intercepto *b*, a separação entre o hiperplano $g(\mathbf{x}) = (\mathbf{w}^T\mathbf{x}) + b = 0$ e o dado de entrada mais próximo é chamada de *margem de separação* e é denotada por $\rho$.
- Sempre que for possível obter um $\rho > 0$, existirão infinitos hiperplanos, dentre os quais se busca um hiperplano particular em que a *margem de separação* $\rho$ *é maximizada*. De acordo com esta condição, a superfície de decisão é dita ser *o hiperplano ótimo* e o método de aprendizado de máquina é denominada máquina de vetores-suporte (SVM, do inglês *support vector machines*), sendo que os dados de treinamento que se encontram à distância $\rho$ do hiperplano são chamados vetores-suporte (*support vectors*)
- O conceito de hiperplano ótimo foi desenvolvido por Vapnik e Chervonenkis em 1965 (VAPNIK & CHERVONENKIS, 1974). As Figuras 5, 6 e 7 ilustram a geometria da construção do hiperplano ótimo para um espaço bidimensional, além da interpretação geométrica dos vetores-suporte.

- Os dados para os quais o resultado da equação (14) é igual a 1 são os vetores-suporte, pois são aqueles que se encontram à distância $\rho$ do hiperplano ótimo, produzindo:

$$y_i\,[(\,\mathbf{w}^{\mathrm{T}}\mathbf{x}_i^{sv}) + b] - 1 = 0 \qquad (15)$$

- Os vetores-suporte exercem um papel importante nas operações deste tipo de aprendizado de máquina. Em termos conceituais, eles são os pontos que se encontram mais perto da superfície de decisão e, portanto, são os de classificação mais difícil. Como tal, eles têm uma relação direta com a localização da superfície de decisão.
- Usando a Eq. (15), é possível calcular quanto vale a margem de separação, cujo valor já foi apresentado na Figura 6.
- Para tanto, serão tomados dois vetores-suporte, sendo um da classe positiva e o outro da classe negativa. A margem de separação é dada pela metade da projeção na direção de **w** da diferença entre esses vetores-suporte, conforme ilustrado na Figura 8.

$$\rho = \frac{1}{2}\left[\left(\frac{\mathbf{w}}{\|\mathbf{w}\|}\right)^T\left(\mathbf{x}^{sv+} - \mathbf{x}^{sv-}\right)\right] = \frac{1}{2\|\mathbf{w}\|}\left(\mathbf{w}^T\mathbf{x}^{sv+} - \mathbf{w}^T\mathbf{x}^{sv-}\right) = \frac{1}{2\|\mathbf{w}\|}\left[(1-b)-(-1-b)\right] = \frac{1}{\|\mathbf{w}\|} \qquad (16)$$

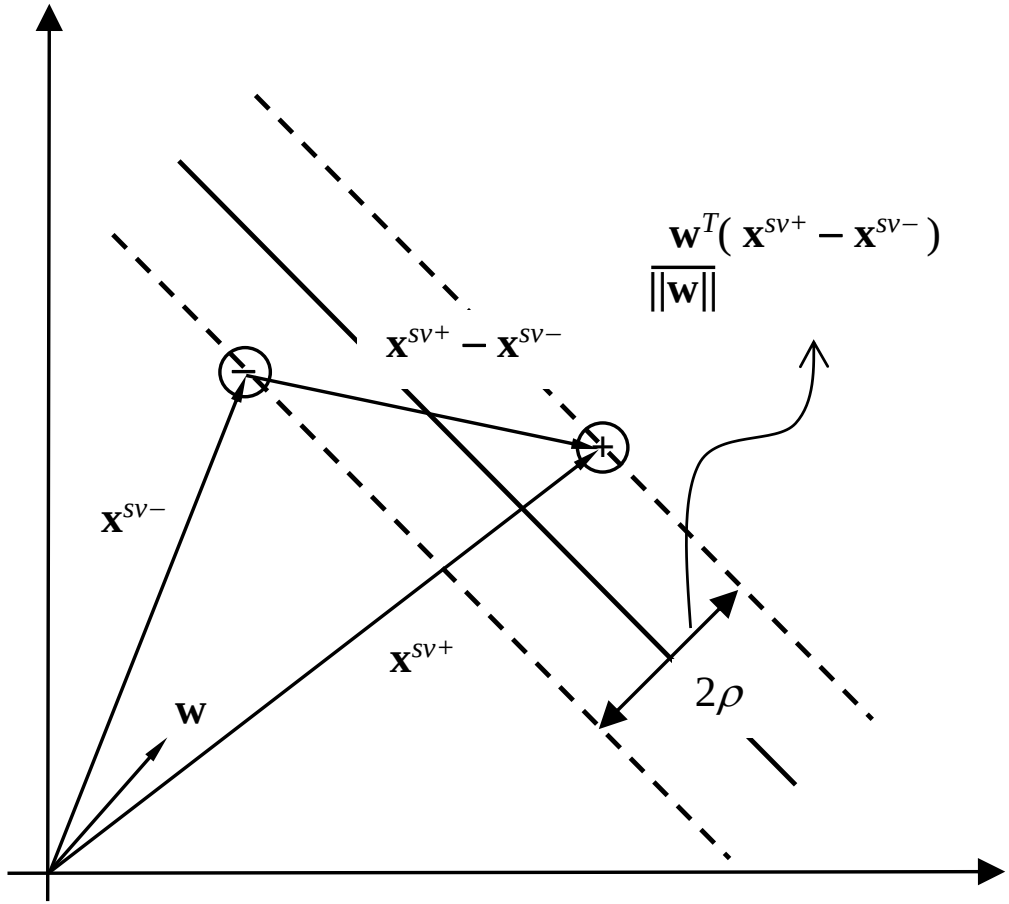



Figura 8 – Estratégia para o cálculo da margem de separação.

- A equação (16) mostra que maximizar a margem de separação entre as classes é equivalente a minimizar a norma euclidiana do vetor de pesos **w**:

$$\max \rho \equiv \max \frac{1}{\|\mathbf{w}\|} \equiv \min \|\mathbf{w}\| \equiv \min \frac{1}{2}\|\mathbf{w}\|^2$$

- Em resumo, o hiperplano ótimo, definido por **w** e *b*, proporciona a máxima separação possível entre as amostras positivas e negativas. Esta condição ótima é alcançada minimizando a norma euclidiana do vetor de pesos **w**.
- É possível, então, formular o problema de otimização, em sua representação primal, para encontrar o hiperplano ótimo para classes linearmente separáveis. A partir dos dados de treinamento $(\mathbf{x}_i, y_i)_{1 \le i \le N}$, $\mathbf{x} \in \mathrm{R}^m$, $y \in \{+1, -1\}$, encontre o valor do vetor de pesos **w** e intercepto *b* que resolvem o seguinte problema:

$$\begin{aligned} &\text{Minimizar} \quad V(\mathbf{w}, b) = \frac{1}{2}\|\mathbf{w}\|^2 \\ &\text{Sujeito a} \quad \forall_{i=1}^{N}: \; y_i\left(\mathbf{w}^T\mathbf{x}_i + b\right) \ge 1 \end{aligned} \tag{17}$$

- O problema (17) pode ser resolvido utilizando o método dos multiplicadores de Lagrange. Considere a função lagrangeana referente ao problema (17):

$$L(\mathbf{w},b,\alpha)=\frac{1}{2}\left(\mathbf{w}^T\mathbf{w}\right)+\sum_{i=1}^{N}\alpha_i\left[1-y_i\left(\mathbf{w}^T\mathbf{x}_i+b\right)\right] \tag{18}$$

onde os multiplicadores de Lagrange $\alpha_i$ são todos não-negativos. Sendo a função-objetivo do problema (17) convexa e todas as suas restrições funções afins, a solução é encontrada aplicando as condições de Karush-Kuhn-Tucker junto à função lagrangeana (18), produzindo:

$$\begin{aligned}\frac{\partial L(\mathbf{w},b,\alpha)}{\partial \mathbf{w}}&=\mathbf{w}-\sum_{i=1}^{N}\alpha_i y_i\mathbf{x}_i=0\Rightarrow \mathbf{w}=\sum_{i=1}^{N}\alpha_i y_i\mathbf{x}_i\\ \frac{\partial L(\mathbf{w},b,\alpha)}{\partial b}&=-\sum_{i=1}^{N}\alpha_i y_i=0\end{aligned} \tag{19}$$

- Substituindo as relações obtidas na função lagrangeana (18), obtém-se:

$$L(\mathbf{w},b,\alpha)=\frac{1}{2}\left(\sum_{i=1}^{N}\alpha_i y_i \mathbf{x}_i\right)^T\left(\sum_{j=1}^{N}\alpha_j y_j \mathbf{x}_j\right)+\sum_{i=1}^{N}\alpha_i\left\{1-y_i\left[\left(\sum_{j=1}^{N}\alpha_j y_j \mathbf{x}_j\right)^T \mathbf{x}_i+b\right]\right\}$$

$$=\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j\left(\mathbf{x}_i{}^T\mathbf{x}_j\right)+\sum_{i=1}^{N}\alpha_i-\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j\left(\mathbf{x}_i{}^T\mathbf{x}_j\right)-b\sum_{i=1}^{N}\alpha_i y_i \qquad (20)$$

$$=\sum_{i=1}^{N}\alpha_i-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j\left(\mathbf{x}_i{}^T\mathbf{x}_j\right)$$

- Problema de otimização dual: A partir dos dados de treinamento linearmente separáveis $(\mathbf{x}_i, y_i)_{1\le i\le N}$, $\mathbf{x}\in\Re^m$, $y\in\{+1,-1\}$, encontre os multiplicadores de Lagrange $(\alpha^*_i)_{1\le i\le N}$ que resolvem o problema de otimização quadrático:

$$\text{Maximizar}\quad W(\alpha)=\sum_{i=1}^{N}\alpha_i-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j\left(\mathbf{x}_i{}^T\mathbf{x}_j\right)$$

$$\text{Sujeito a}\quad \sum_{i=1}^{N}\alpha_i y_i=0 \qquad (21)$$

$$\forall_{i=1}^{N}:\alpha_i\ge 0$$

- O problema de otimização dual (21) é totalmente formulado em termos dos dados de treinamento. Além disso, a função $W(\alpha)$ a ser maximizada depende somente dos dados de entrada na forma de produto interno $\left(\mathbf{x}_i^T \mathbf{x}_j\right)_{1 \le i \le N \; ; 1 \le j \le N}$.
- Outro ponto importante é que este problema de otimização tem uma única solução, a qual pode ser eficientemente encontrada. Portanto não há a presença de mínimos locais, como em outras técnicas de classificação.
- Determinando os multiplicadores de Lagrange ótimos $\alpha$*, pode-se calcular o vetor de pesos ótimo **w*** utilizando a primeira equação em (19), produzindo:

$$\mathbf{w}^* = \sum_{i=1}^{N} \alpha_i^* y_i \mathbf{x}_i \text{ onde } y \in \{+1, -1\}, \alpha_i^* \in \Re \text{ e } \mathbf{x} \in \Re^m \qquad (22)$$

- Assim, **w*** é o vetor associado ao hiperplano ótimo com a máxima margem de separação $\rho$ (veja equação (16)).
- O valor do intercepto ótimo *b** é encontrado utilizando a equação (22), com o auxílio das restrições primais em (17), levando a:

$$b^* = -\frac{1}{2}\left[\max_{\{i|y_i=-1\}}\left(\sum_{j=1}^{N_{sv}}\alpha_j^{sv} y_j^{sv}\left(\mathbf{x}_i{}^T \mathbf{x}_j^{sv}\right)\right) + \min_{\{i|y_i=+1\}}\left(\sum_{j=1}^{N_{sv}}\alpha_j^{sv} y_j^{sv}\left(\mathbf{x}_i{}^T \mathbf{x}_j^{sv}\right)\right)\right] \quad (23)$$

onde $N_{sv}$ é o número de vetores-suporte.

- Utilizando a condição de complementariedade de Karush-Kuhn-Tucker, obtém-se a seguinte relação:

$$\forall_{i=1}^{N}: \ \alpha_i^*\left[y_i\left(\mathbf{w}^{*T}\ \mathbf{x}_i + b^*\right) - 1\right] = 0 \quad (24)$$

que proporciona uma importante informação sobre a estrutura da solução. Isto implica que somente para os dados de entrada $\mathbf{x}_i$ para os quais a margem é igual a 1 (e, portanto, localizados à distância $\rho$ do hiperplano) tem-se seu correspondente $\alpha^*$ diferente de zero. Todos os outros dados de entrada têm o parâmetro $\alpha^*$ igual a zero.

- Através das condições de complementariedade de Karush-Kuhn-Tucker, pode-se demonstrar que:

$$\left(\mathbf{w}^{*T}\ \mathbf{w}^*\right) = \sum_{i=1}^{N_{sv}} \alpha_i^{sv^*} \quad (25)$$

- Portanto, a norma do vetor de pesos *w**, que está associado ao hiperplano de máxima margem, é também dada por:

$$\rho = \frac{1}{\|\mathbf{w}\|} = \left( \sum_{i=1}^{N_{sv}} \alpha_i^{sv^*} \right)^{-\frac{1}{2}} \tag{26}$$

- Como já mencionado, os dados de entrada com a margem igual a 1 são chamados de vetores-suporte, sendo justamente aqueles com os multiplicadores de Lagrange $\alpha^*$ diferentes de zero. Logo, são os únicos pontos que exercem influência na construção do hiperplano de máxima margem.
- Além disso, o hiperplano ótimo é expresso somente em termos deste conjunto de vetores-suporte, como descrito a seguir:

$$f(\mathbf{x}) = \text{sgn}\left( \sum_{i=1}^{N_{sv}} \alpha_i^{sv^*} y_i^{sv} \left( \mathbf{x}_i^{sv^T} \mathbf{x} \right) + b^* \right) \tag{27}$$

- Os dados de entrada que não são vetores-suporte também não têm nenhuma influência na função de decisão produzida pela SVM.
- A função de decisão (27) é utilizada da seguinte maneira: Se o resultado de $f(\mathbf{x})$ for negativo, o ponto $\mathbf{x}$ pertence à classe negativa; se o resultado de $f(\mathbf{x})$ for positivo, o ponto $\mathbf{x}$ pertence à classe positiva.

## 6.2 Teorema de Mercer e truque do kernel (*kernel trick*)

- O teorema de Mercer foi proposto em: Mercer, J. "Functions of positive and negative type and their connection with the theory of integral equations", Philosophical Transactions of the Royal Society A 209 (441–458): 415–446, 1909.
- Ele afirma que uma função simétrica $K(\mathbf{x}_i, \mathbf{x}_j)$, ou seja, $K(\mathbf{x}_i, \mathbf{x}_j) = K(\mathbf{x}_j, \mathbf{x}_i)$, com $\mathbf{x}_i, \mathbf{x}_j \in \Re^m$, pode ser expressa como um produto interno

$$K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$$

para alguma função $\phi(\cdot): \Re^m \rightarrow \Re^q$, se e somente se $K(\mathbf{x}_i, \mathbf{x}_j)$ é semidefinida positiva, ou seja:

$$\int K(\mathbf{x}_i, \mathbf{x}_j) g(\mathbf{x}_i) g(\mathbf{x}_j) d\mathbf{x}_i d\mathbf{x}_j \geq 0 \quad \forall g(\cdot)$$

ou, de forma equivalente, se e somente se a matriz

$$\begin{bmatrix} K(\mathbf{x}_1, \mathbf{x}_1) & K(\mathbf{x}_1, \mathbf{x}_2) & \cdots & K(\mathbf{x}_1, \mathbf{x}_N) \\ K(\mathbf{x}_2, \mathbf{x}_1) & \ddots & & K(\mathbf{x}_2, \mathbf{x}_N) \\ \vdots & & \ddots & \vdots \\ K(\mathbf{x}_N, \mathbf{x}_1) & K(\mathbf{x}_N, \mathbf{x}_2) & & K(\mathbf{x}_N, \mathbf{x}_N) \end{bmatrix}$$

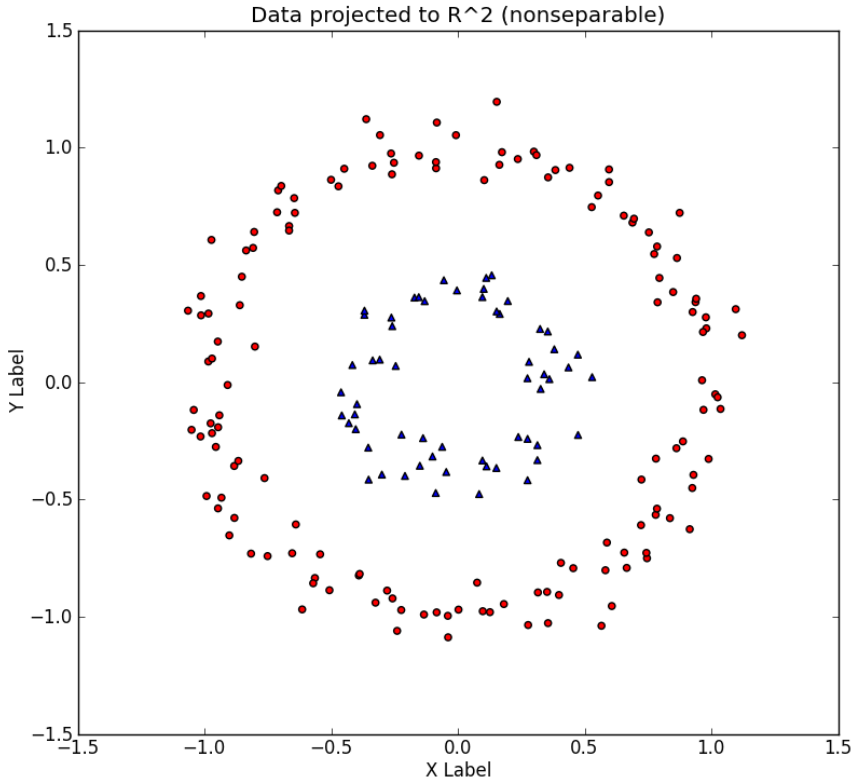
é semidefinida positiva para qualquer coleção de pontos $\{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$ no $\Re^m$.

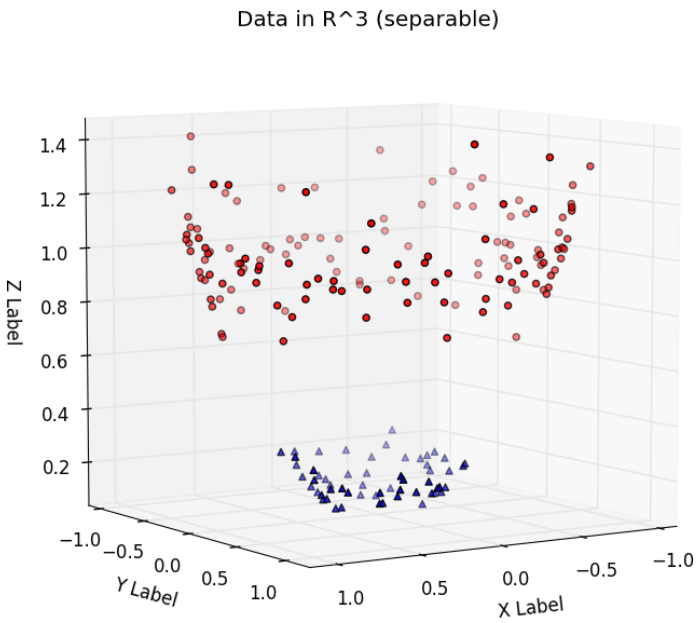
- Nos casos em que se tem *q* bem maior que *m*, em lugar de realizar o produto interno $\phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$, é possível aplicar diretamente $K(\mathbf{x}_i, \mathbf{x}_j)$, que passa então a ser chamada de produto interno kernel.

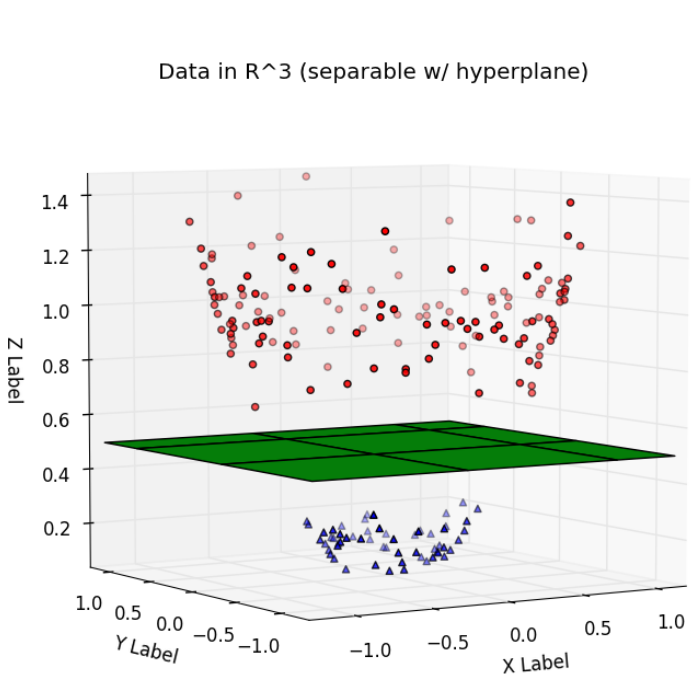
- Tem-se aqui o chamado truque do kernel (*kernel trick*), pois substitui-se um produto interno num espaço de elevada dimensão $q$ (que será chamado de espaço de características, do inglês *feature space*) pela aplicação da função kernel no espaço original.
- Na verdade, não é sequer necessário conhecer a função $\phi(\cdot): \Re^m \rightarrow \Re^q$.

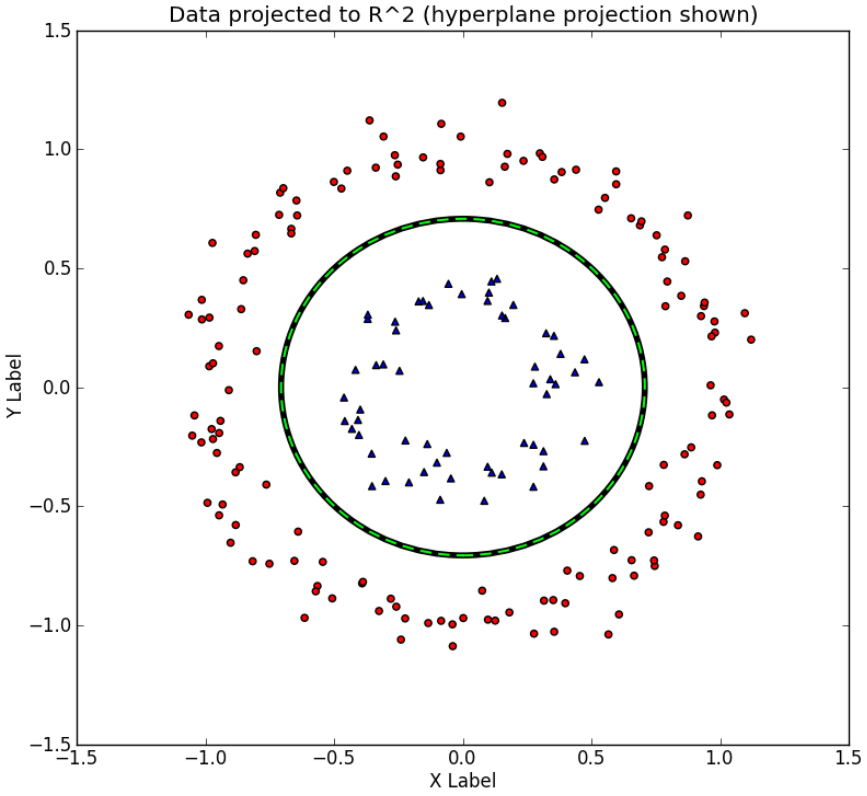
## 6.3 Separabilidade linear com o aumento da dimensão do espaço

- Extraído de [http://www.eric-kim.net/eric-kim-net/posts/1/kernel_trick.html], o exemplo abaixo ilustra como obter um problema de classificação linear ao expandir o espaço original de um problema de classificação não-linear.


Data projected to R^2 (nonseparable)
Y Label
X Label
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5


Data in R^3 (separable)
Z Label
Y Label
X Label
1.4
1.2
1.0
0.8
0.6
0.4
0.2




- O exemplo indica que é possível interpretar um problema de classificação não-linear como a projeção para um espaço de menor dimensão de um problema de classificação linear em um espaço de maior dimensão. A fronteira discriminante, que era um hiperplano no espaço expandido, torna-se não-linear no espaço original.

## 6.4 Extensão para o espaço de características (*feature space*)

- A partir dos dados de treinamento $(\mathbf{x}_i, y_i)_{1 \le i \le N}$, $\mathbf{x} \in \Re^m$ e $y \in \{+1, -1\}$, linearmente separáveis no espaço de características definido pelo produto interno kernel $K(\mathbf{x}_i, \mathbf{x}_j)$, encontre os multiplicadores de Lagrange $(\alpha^*_i)_{1 \le i \le N}$ que resolvem o problema de otimização quadrático:

$$\begin{aligned} &\text{Maximizar} \quad W(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \\ &\text{Sujeito a} \quad \sum_{i=1}^{N} \alpha_i y_i = 0 \quad ; \quad \forall_{i=1}^{N} : \alpha_i \ge 0 \end{aligned} \tag{28}$$

- Assim, a função de decisão dada pela SVM assume a forma:

$$f(\mathbf{x}) = \text{sgn}\left( \sum_{i=1}^{N_{sv}} \alpha_i^{sv^*} y_i^{sv} K(\mathbf{x}_i^{sv}, \mathbf{x}) + b^* \right) \tag{29}$$

que é equivalente ao hiperplano de máxima margem no espaço de características, definido implicitamente pelo produto interno kernel $K(\cdot, \mathbf{x})$.

- Formalmente, se $\phi(\cdot): \Re^m \to \Re^q$ for a função que mapeia do espaço original (de dimensão *m*) para o espaço de características (de dimensão *q*), então vale a seguinte relação: $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$.
- Basicamente, $K(\mathbf{x}_i, \mathbf{x}_j): \Re^m \times \Re^m \to \Re$ calcula a similaridade entre os vetores $\mathbf{x}_i, \mathbf{x}_j \in \Re^m$ quando esses são mapeados por $\phi(\cdot): \Re^m \to \Re^q$ do espaço original para um espaço expandido, denominado de espaço de características.
- Aqui, é necessário satisfazer o teorema de Mercer para que o problema (28) seja convexo e com uma única solução.
- O único grau de liberdade deste hiperplano ótimo é a escolha de qual produto interno kernel utilizar. Algum conhecimento prévio do problema pode ajudar na escolha do tipo de produto interno kernel mais adequado e, com isso, restará apenas ajustar seus parâmetros.

## 6.5 Hiperplano ótimo para classes não-linearmente separáveis

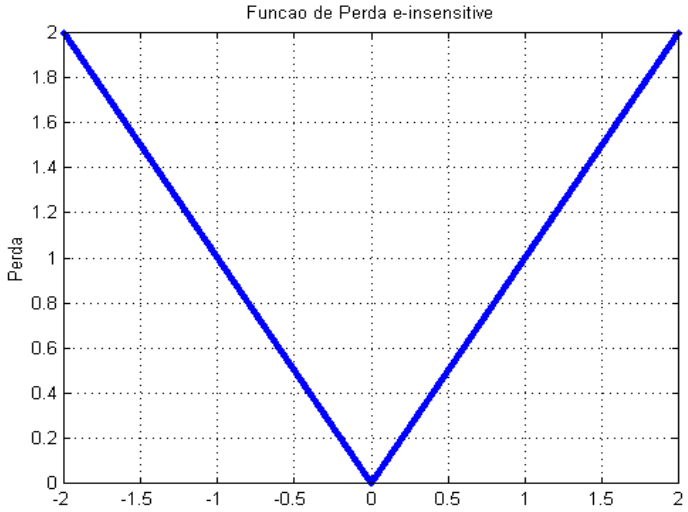
- Não será apresentada aqui a formulação completa para o caso em que os dados não são linearmente separáveis no espaço de características.
- Será dada ênfase à motivação geométrica e antecipa-se que é necessário trabalhar com variáveis de folga ($\xi_i$), de modo a admitir erros no processo de classificação.
- Cria-se, assim, um compromisso entre a maximização da margem e a minimização do somatório desses erros admissíveis, conforme indicado na formulação a seguir, referente ao problema primal:

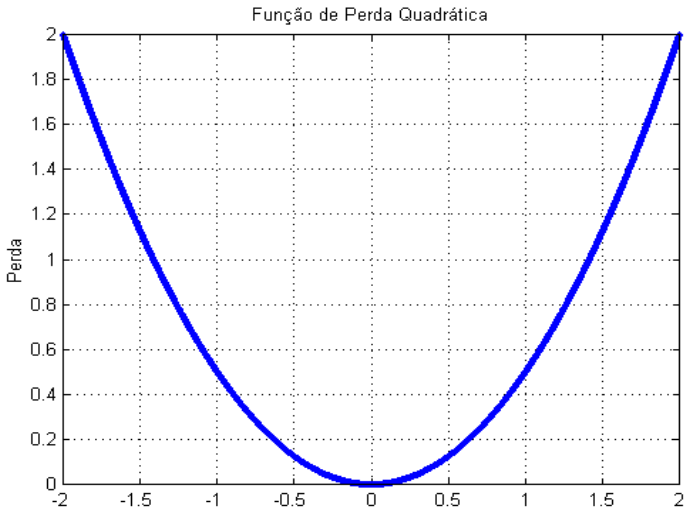
$$\text{Minimizar} \quad V(\mathbf{w},b,\xi)=\frac{1}{2}\|\mathbf{w}\|^2+C_+\sum_{i:y_i=+1}\xi_i+C_-\sum_{j:y_j=-1}\xi_j$$

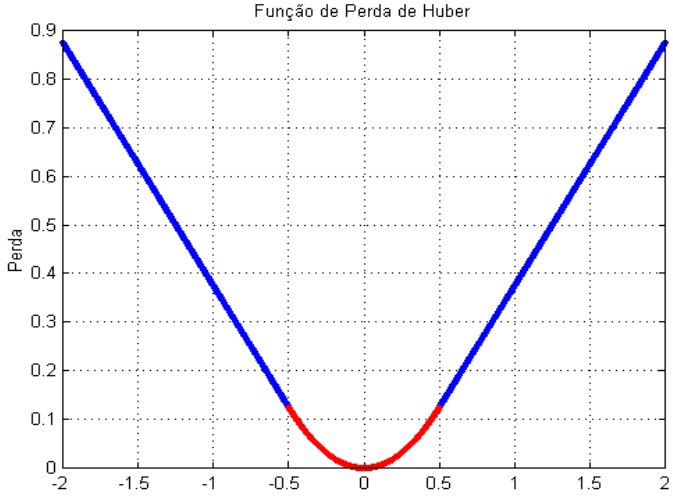
$$\text{Sujeito a} \quad \forall_{i=1}^{N}: \quad y_i\left[\mathbf{w}^T\mathbf{x}_i+b\right]\geq 1-\xi_i$$

$$\forall_{i=1}^{N}: \quad \xi_i\geq 0$$

- Outras funções de perda podem ser consideradas (a atual é a da esquerda):



- A formulação dual é dada como segue:

$$\text{Maximizar} \quad W(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,\, j=1}^{N} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$$

$$\text{Sujeito a} \quad \sum_{i=1}^{N} \alpha_i y_i = 0$$

$$\forall_{i=1}^{N} : 0 \leq \alpha_i \leq C_+ \text{, caso } y_i = +1$$

$$\forall_{i=1}^{N} : 0 \leq \alpha_i \leq C_- \text{, caso } y_i = -1$$

- É preciso definir os coeficientes de penalização $C_+$ e $C_-$, que influenciam no desempenho de generalização do classificador. Para tanto, pode-se empregar validação cruzada, por exemplo.
- A Figura 9 ilustra o emprego das variáveis de folga.

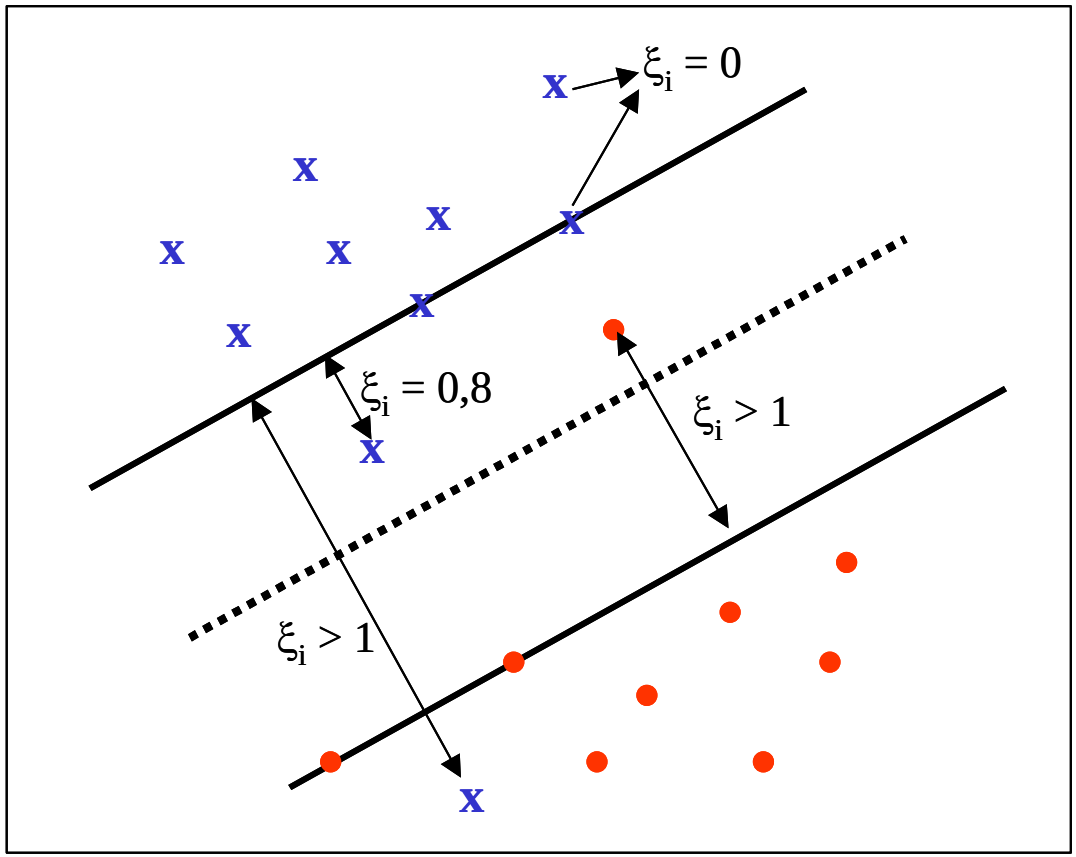



Figura 9 – Papel exercido pelas variáveis de folga

# 7. Alguns tipos de funções kernel

- Linear:

$$K(\mathbf{x},\mathbf{y}) = \mathbf{x}^T\mathbf{y} + c$$

- Polinomial:

$$K(\mathbf{x},\mathbf{y}) = \left(a\left(\mathbf{x}^T\mathbf{y}\right) + c\right)^d$$

- Função de base radial gaussiana:

$$K(\mathbf{x},\mathbf{y}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|_2^2}{2\sigma^2}\right)$$

- Função de base radial laplaciana:

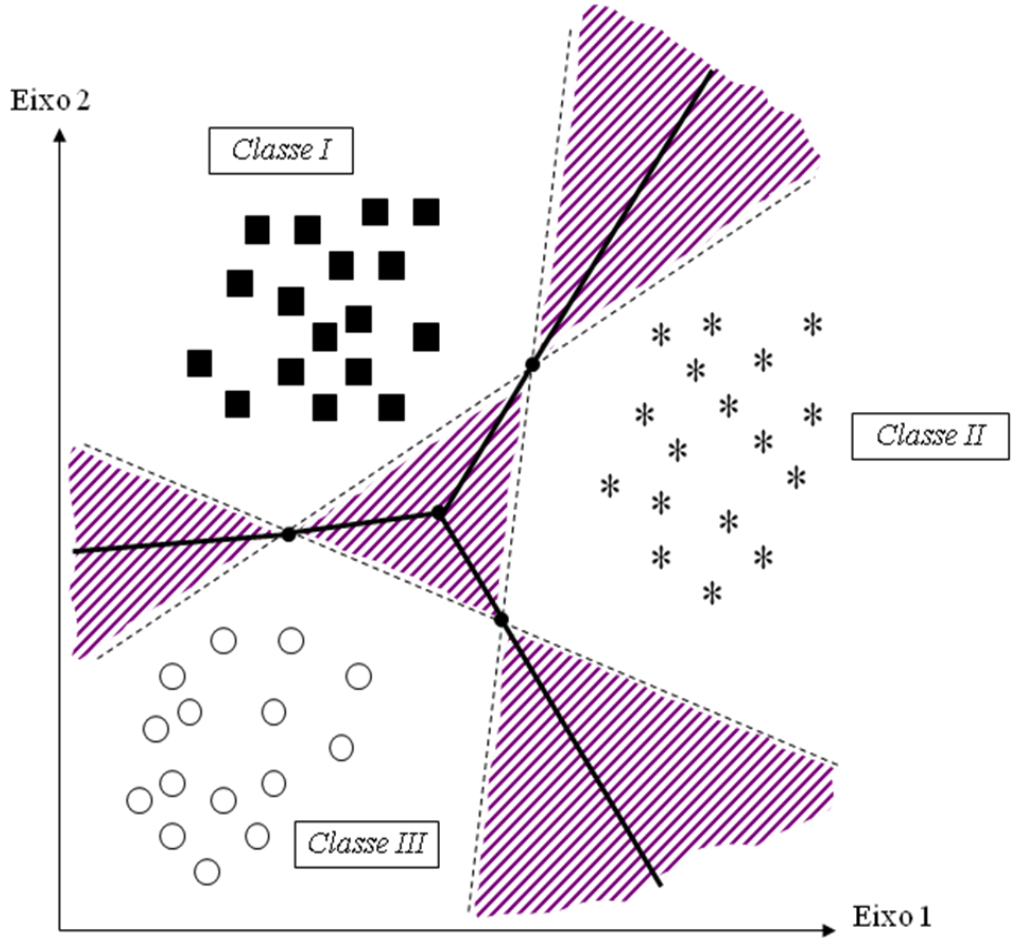
$$K(\mathbf{x},\mathbf{y}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|_1}{2\sigma^2}\right)$$

- Tangente hiperbólica:

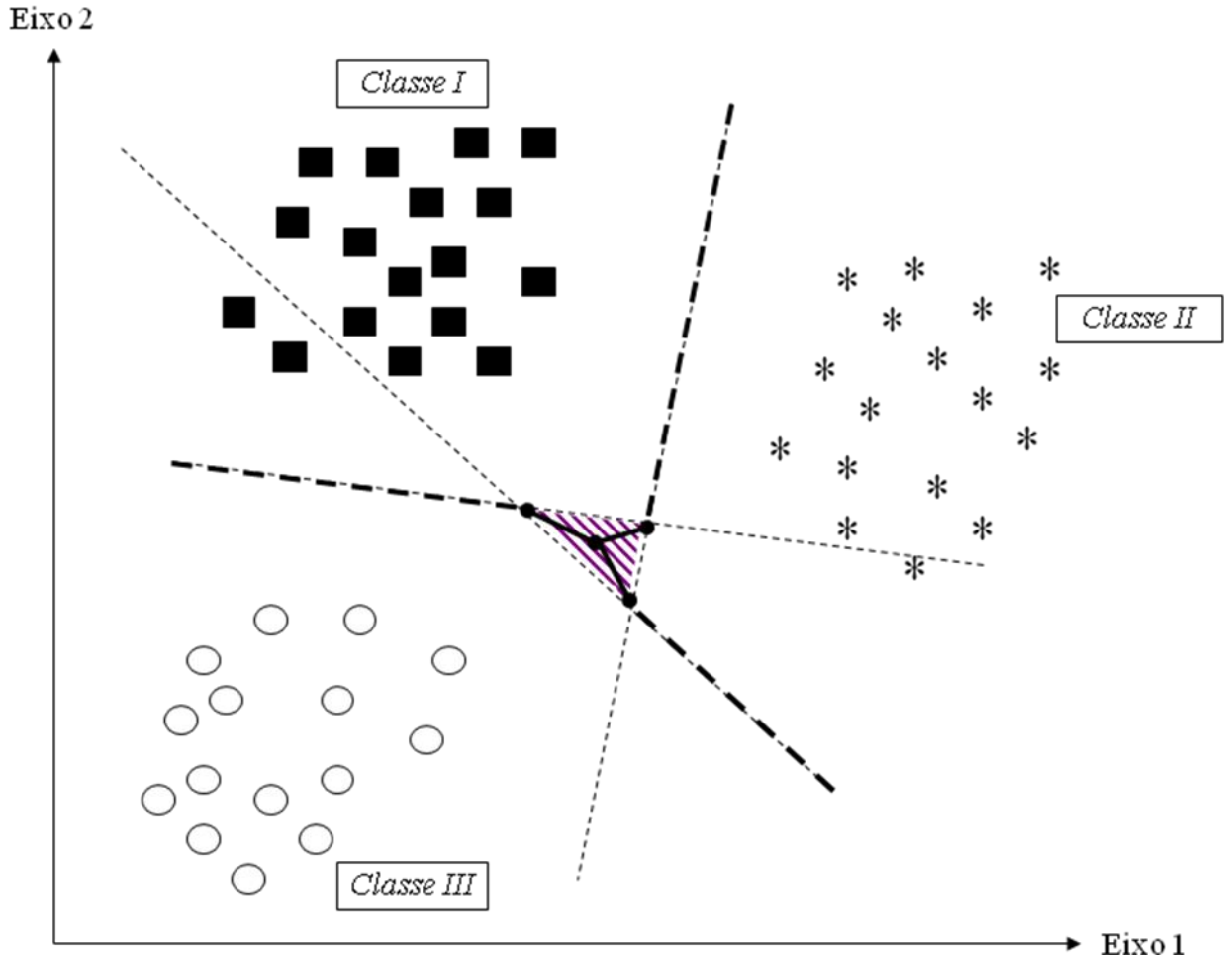
$$K(\mathbf{x},\mathbf{y}) = \tanh\left(a\left(\mathbf{x}^T\mathbf{y}\right) + c\right)$$

# 8. Extensões para o tratamento de múltiplas classes

## 8.1 Um contra todos



## 8.2 Um contra um



## 8.3 Grafo direcionado acíclico (DAGSVM)

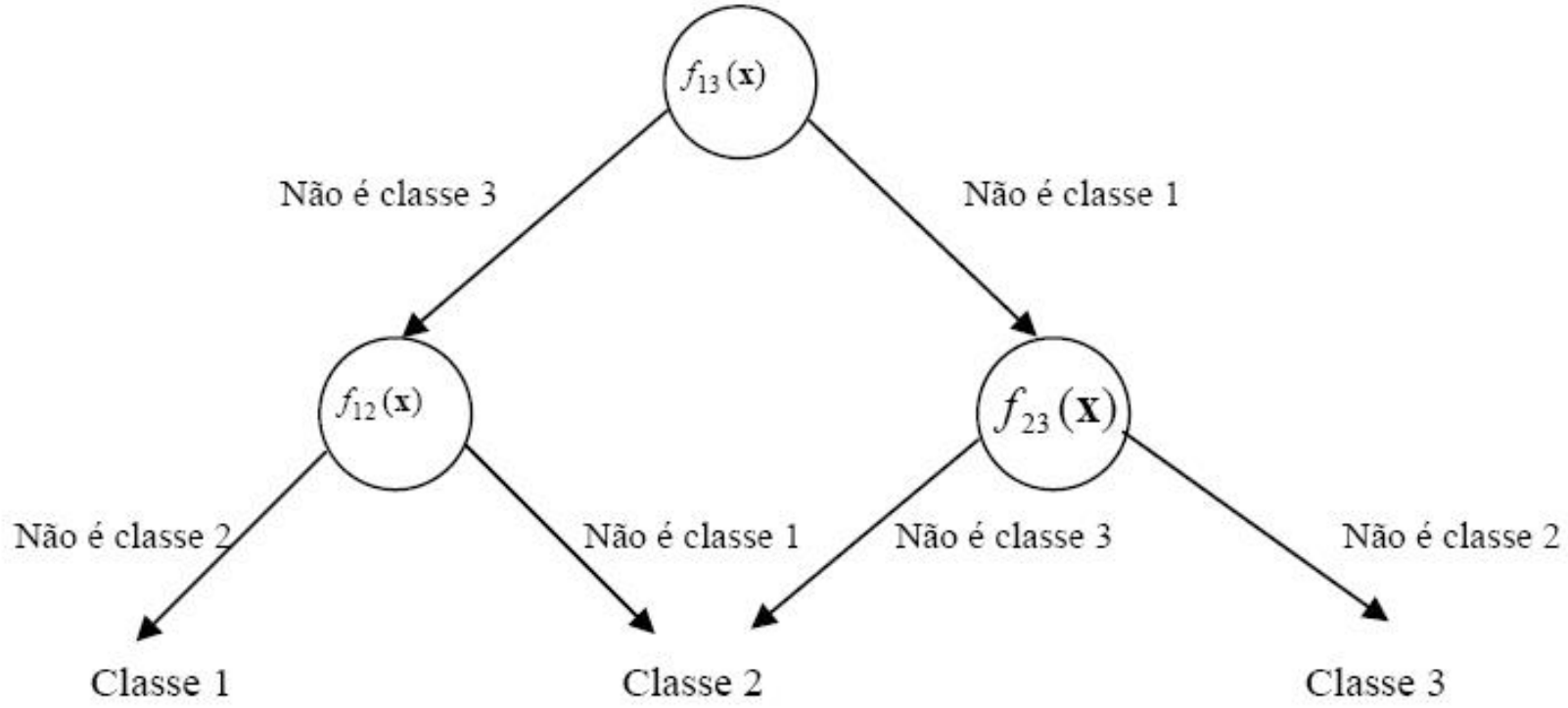
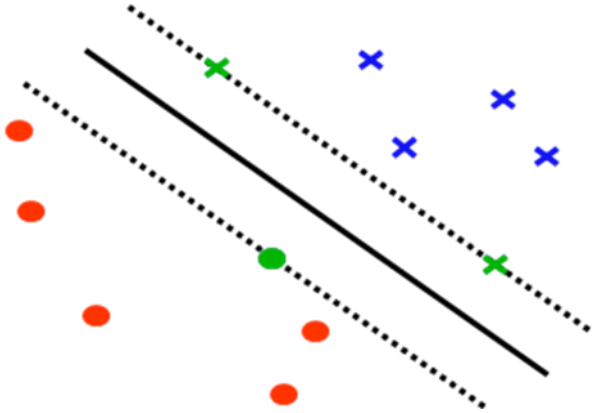
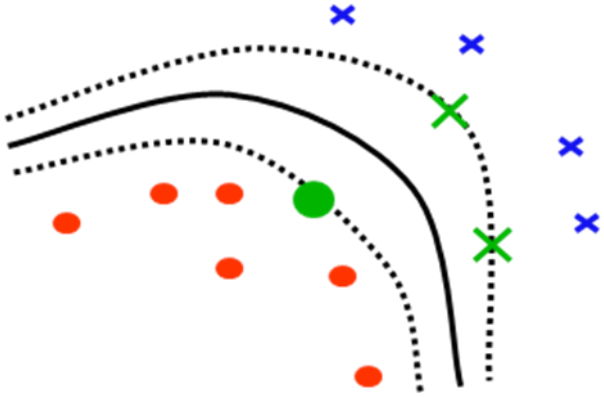



Figura 10 – Ilustração de um grafo direcionado acíclico para um problema com 3 classes no método DAGSVM.

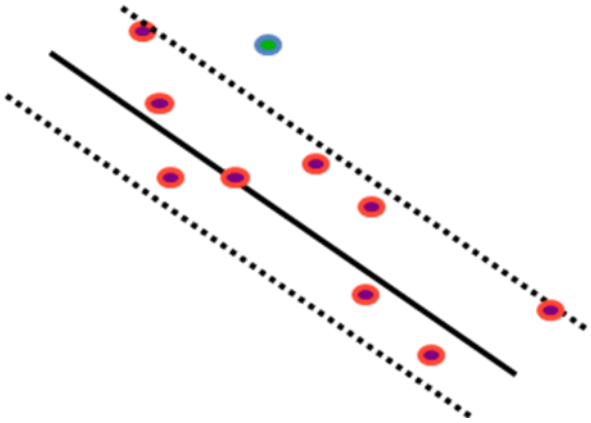
# 9. Extensão para o tratamento de problemas de regressão

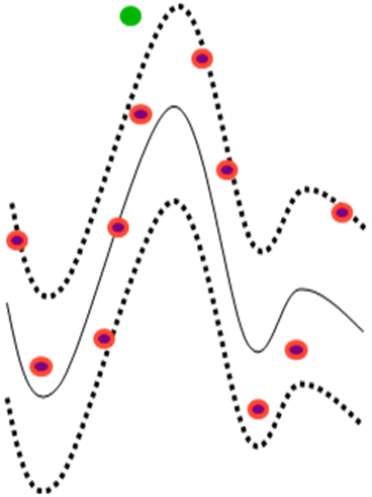
- Enquanto para classificação de dados tem-se:





- Para regressão de dados, são necessárias adaptações, produzindo:





- Cria-se, assim, um compromisso entre a minimização da margem e a minimização do somatório dos erros admissíveis, conforme indicado na formulação a seguir, referente ao problema primal:

$$\text{Minimizar} \quad V\left(\mathbf{w}, b, \xi^{+}, \xi^{-}\right) = \frac{1}{2}\|\mathbf{w}\|^2 + C\left(\sum_{i: y_i - \left[\mathbf{w}^T \mathbf{x}_i + b\right] \geq 0} \xi_i^{+} + \sum_{i: y_i - \left[\mathbf{w}^T \mathbf{x}_i + b\right] \leq 0} \xi_i^{-}\right)$$

$$\text{Sujeito a} \quad \forall_{i=1}^{N}: \quad y_i - \left[\mathbf{w}^T \mathbf{x}_i + b\right] \leq \varepsilon + \xi_i^{+}$$

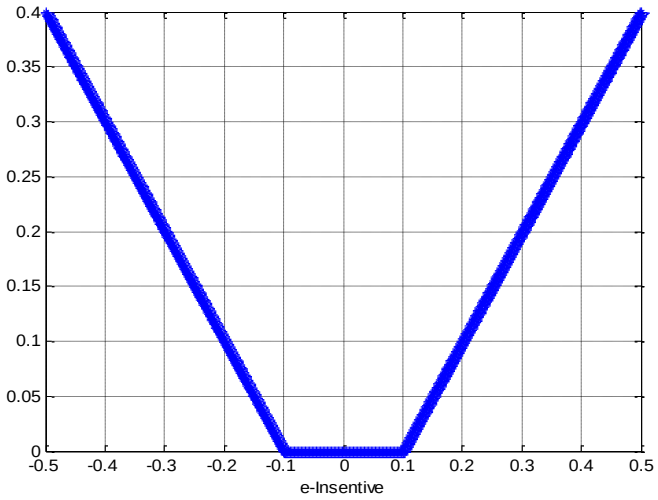
$$\forall_{i=1}^{N}: \quad \left[\mathbf{w}^T \mathbf{x}_i + b\right] - y_i \leq \varepsilon + \xi_i^{-}$$

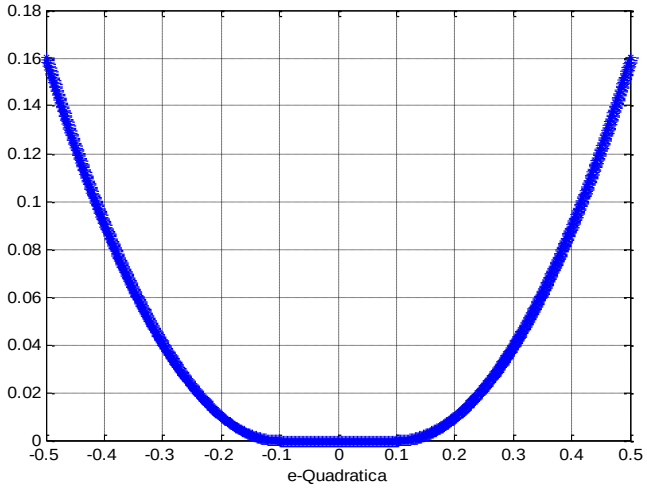
$$\forall_{i=1}^{N}: \quad \xi_i^{+} \geq 0$$

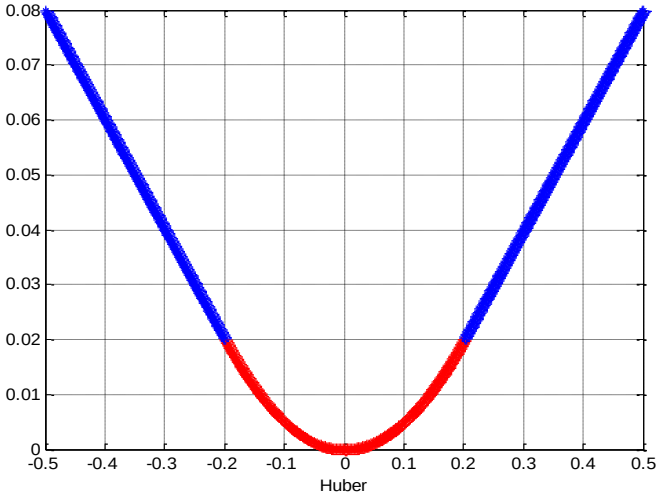
$$\forall_{i=1}^{N}: \quad \xi_i^{-} \geq 0$$

- O hiperplano ótimo é aquele de mínima margem e que mais se aproxima da distribuição dos dados, ou seja, o hiperplano deve estar o mais próximo possível dos dados (deve "passar" pelos dados).
- Quando aplicada no contexto de regressão, SVM é denominada regressão de vetores-suporte (SVR, do inglês *Support Vector Regression*).

- Outras funções de perda podem ser consideradas (a atual é a da esquerda):



## 9.1 Exemplos de comportamento para diferentes ε's

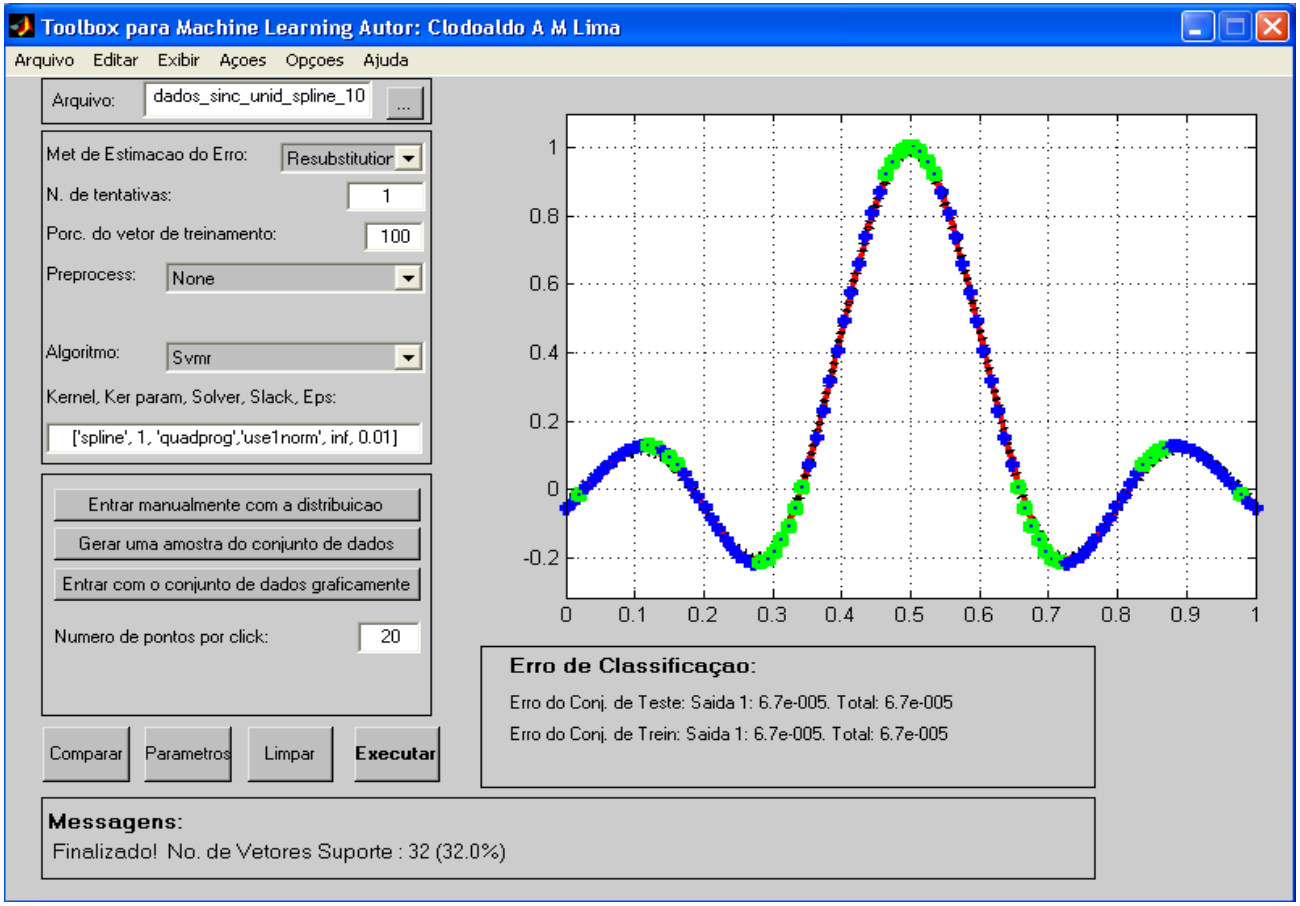



Figura 11 – ε = 0,01; No. de amostras = 100; No. de vetores-suporte = 32

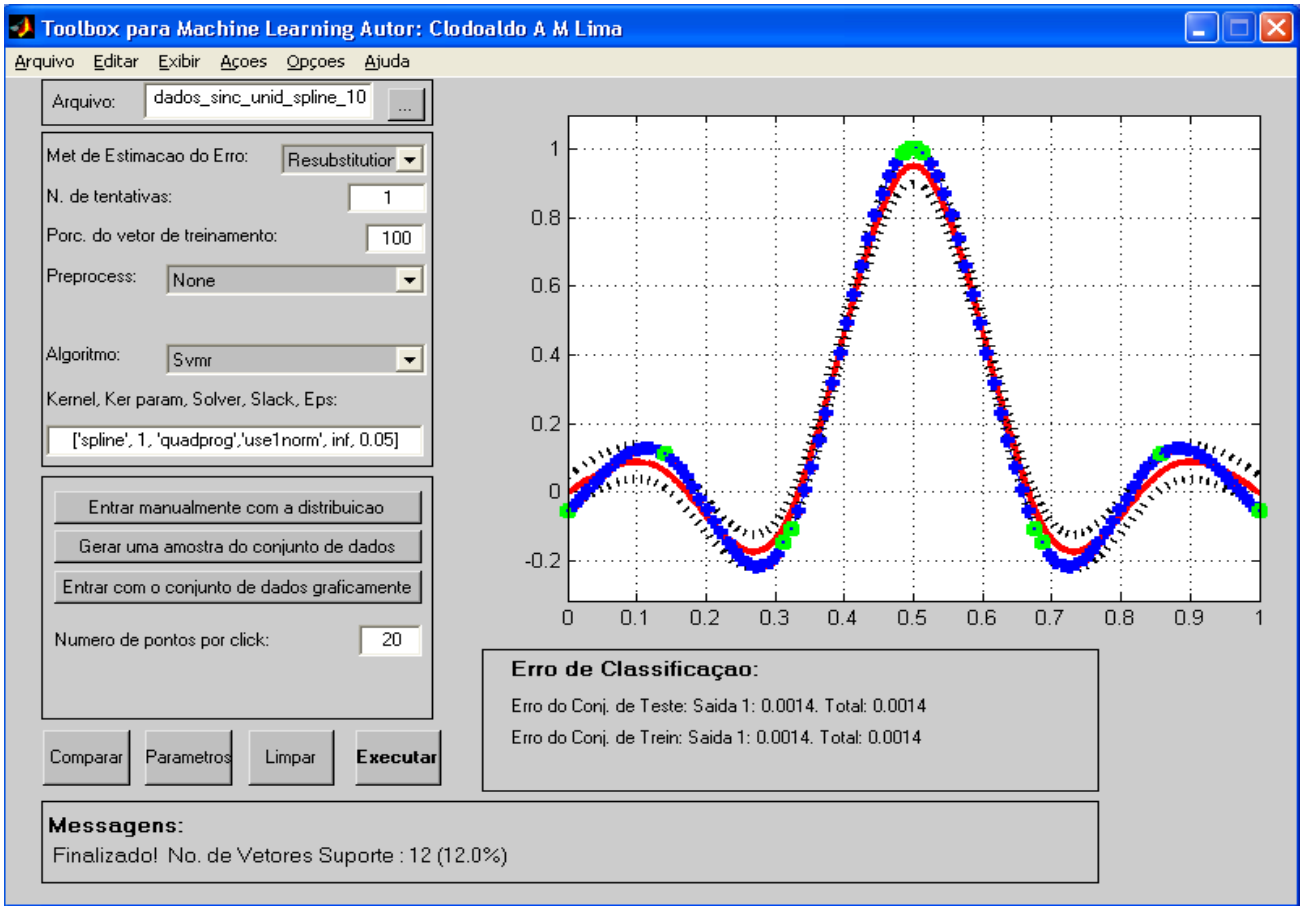



Figura 12 – ε = 0,05; No. de amostras = 100; No. de vetores-suporte = 12

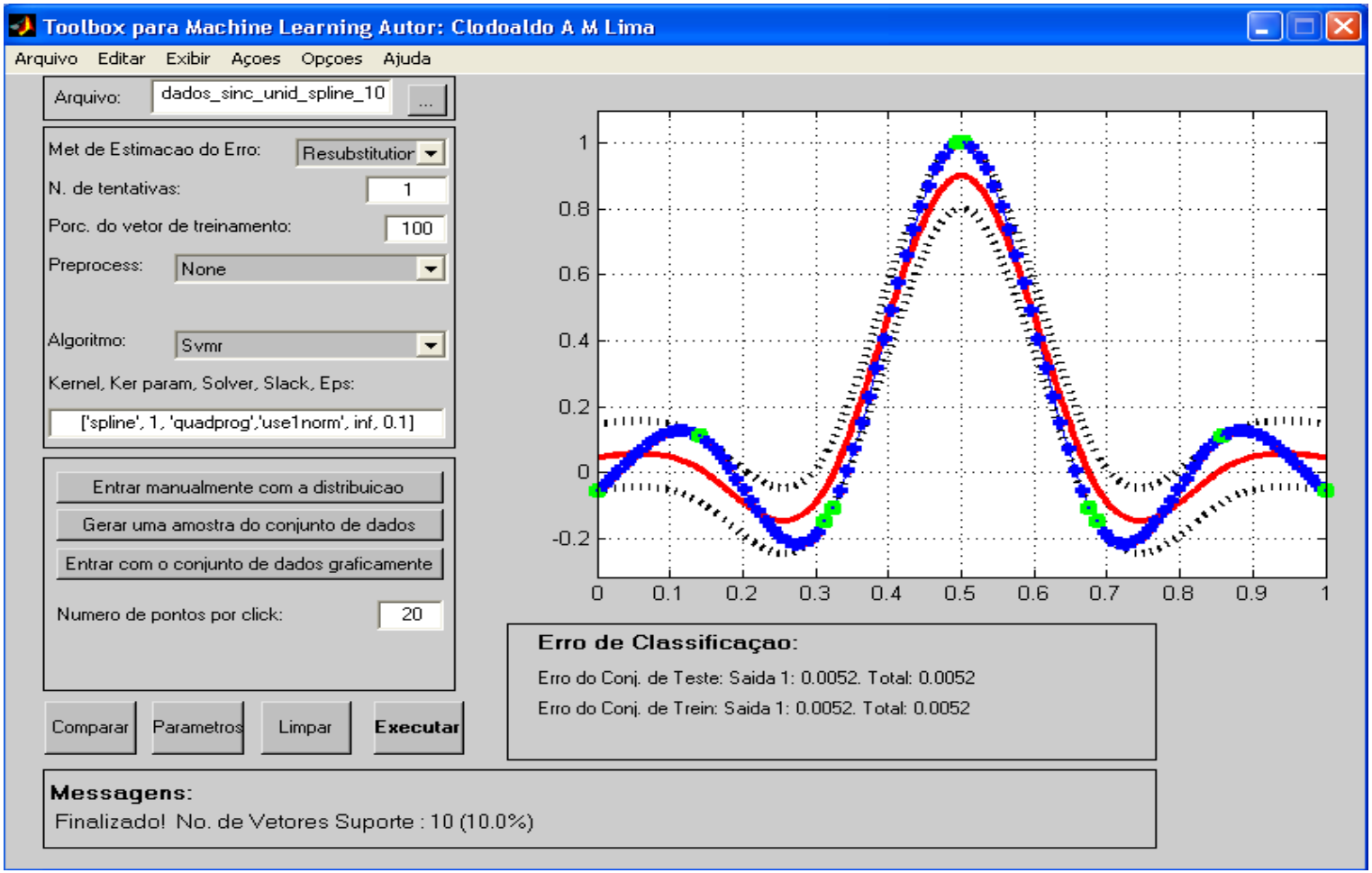



Figura 13 – ε = 0,1; No. de amostras = 100; No. de vetores-suporte = 10

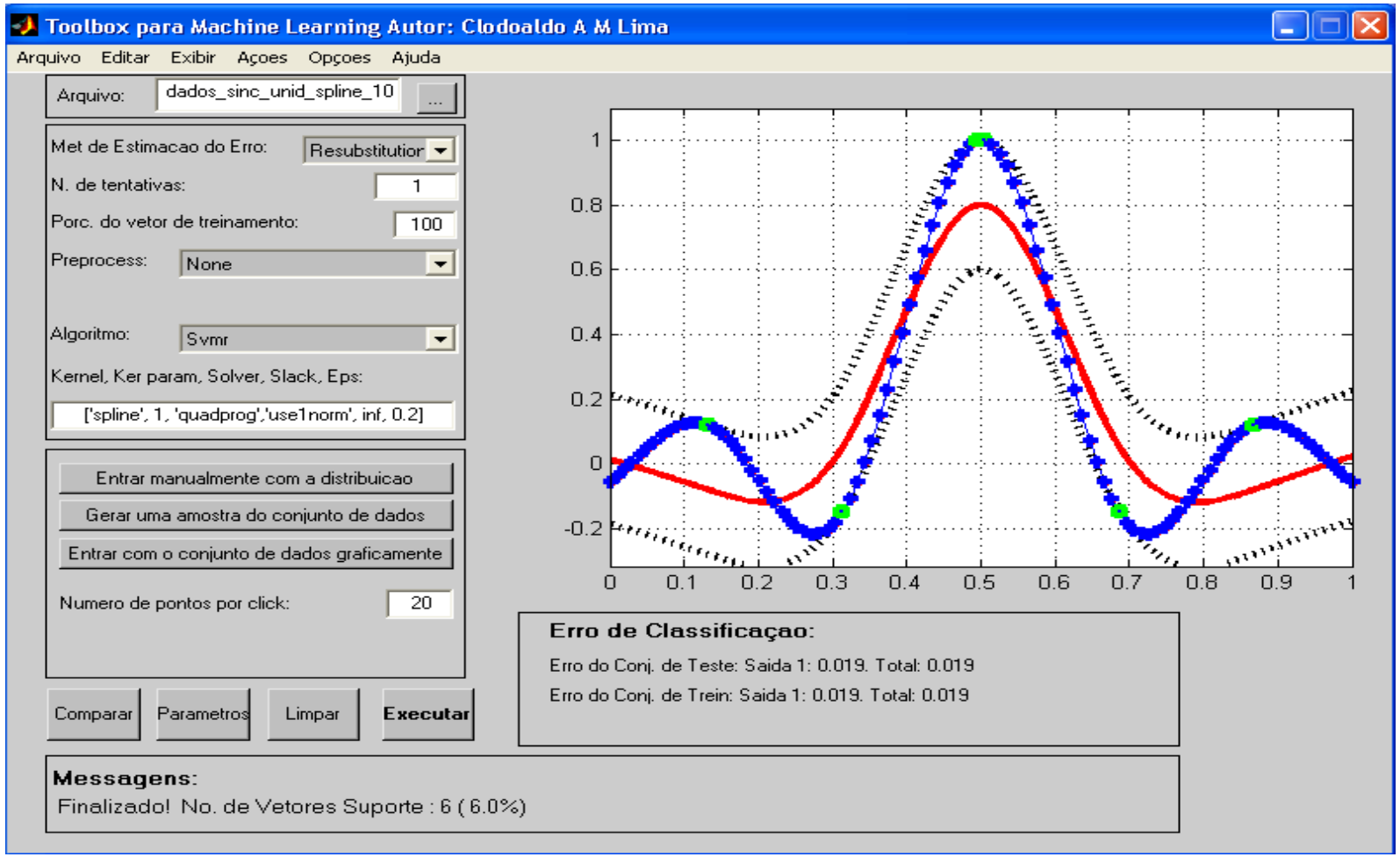



Figura 14 – ε = 0,2; No. de amostras = 100; No. de vetores-suporte = 6

# 10. Interpretação dos principais resultados

- As SVMs mapeiam os dados originais em espaços de maior dimensão e, neste espaço expandido, denominado de espaços de características, resolvem um problema linear.

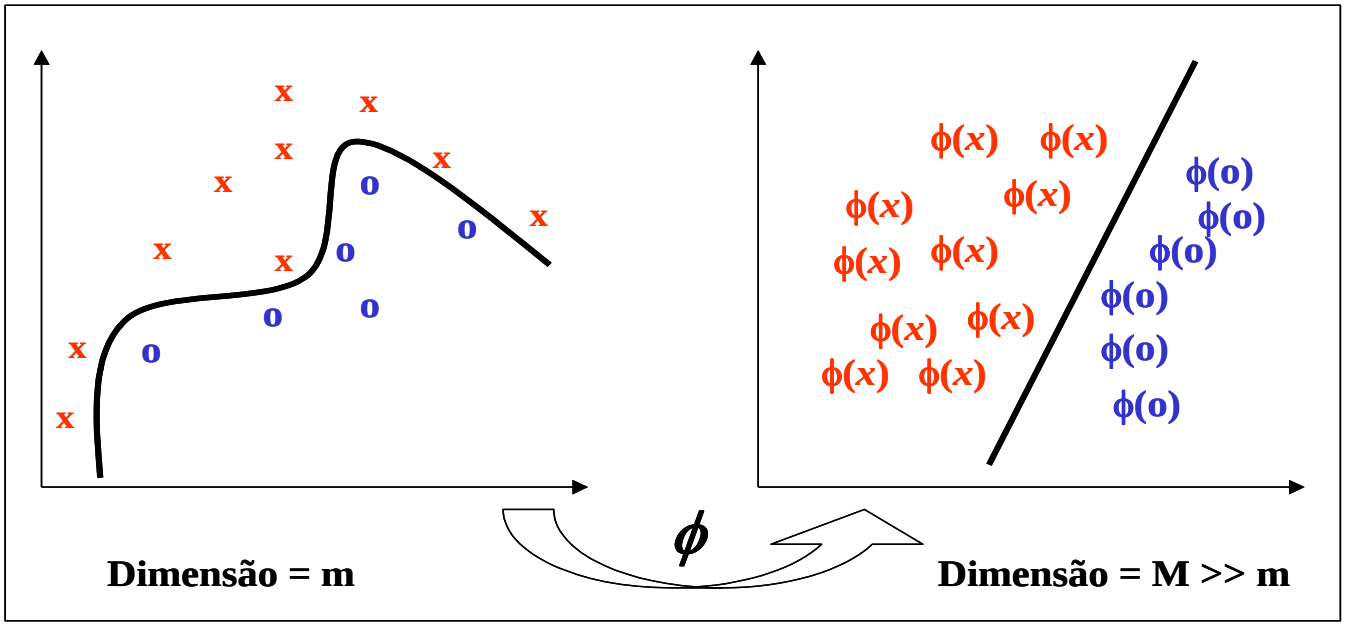



Figura 15 – Mapeamento para o espaço de características

- O resultado final do processo de otimização vai ser uma rede neural com uma camada intermediária.

- O número de nós da camada de entrada é igual à dimensão do vetor de entrada. A quantidade de vetores-suporte determina a quantidade de nós na camada intermediária. O nó de saída constrói uma função linear no espaço de características, o qual é determinado por uma transformação não-linear escolhida a priori, por meio de produtos internos kernel.

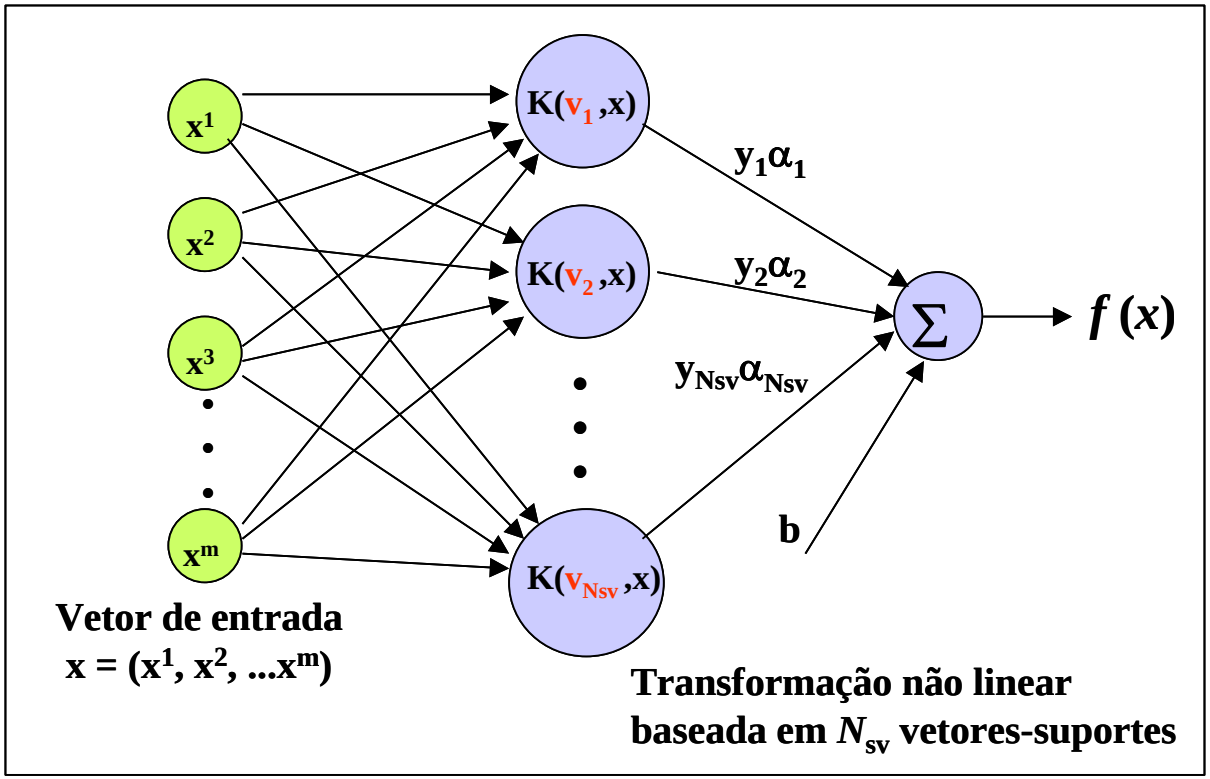



Figura 16 – Resultado final de uma SVM


Figura 17 – Princípio de minimização do risco estrutural

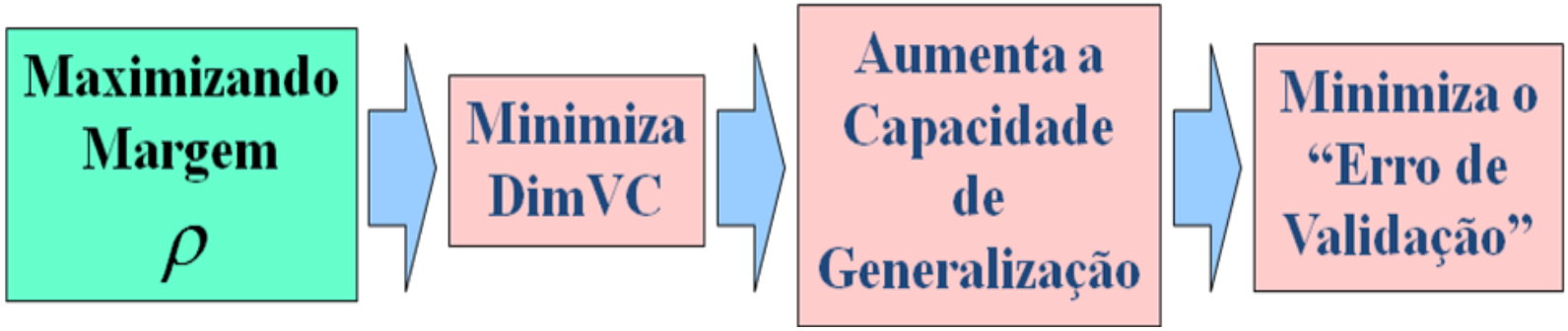



Figura 18 – Princípio dos classificadores de máxima margem

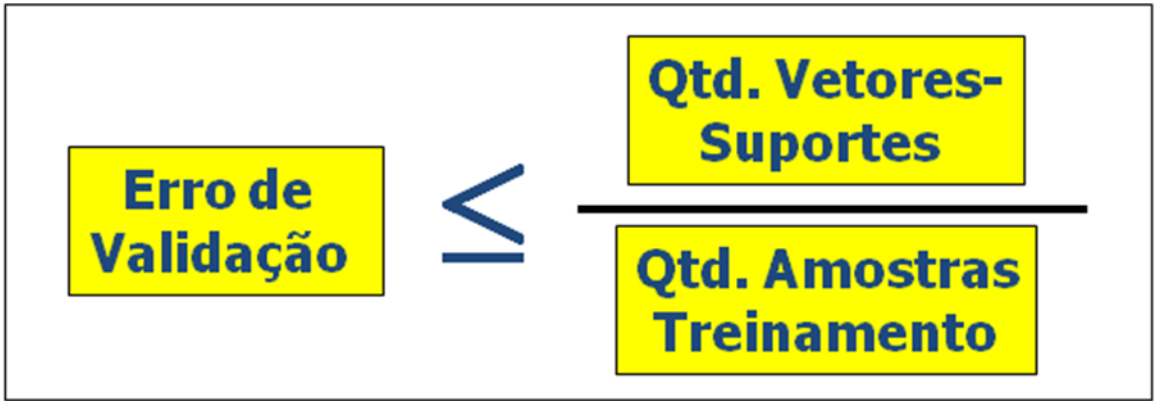
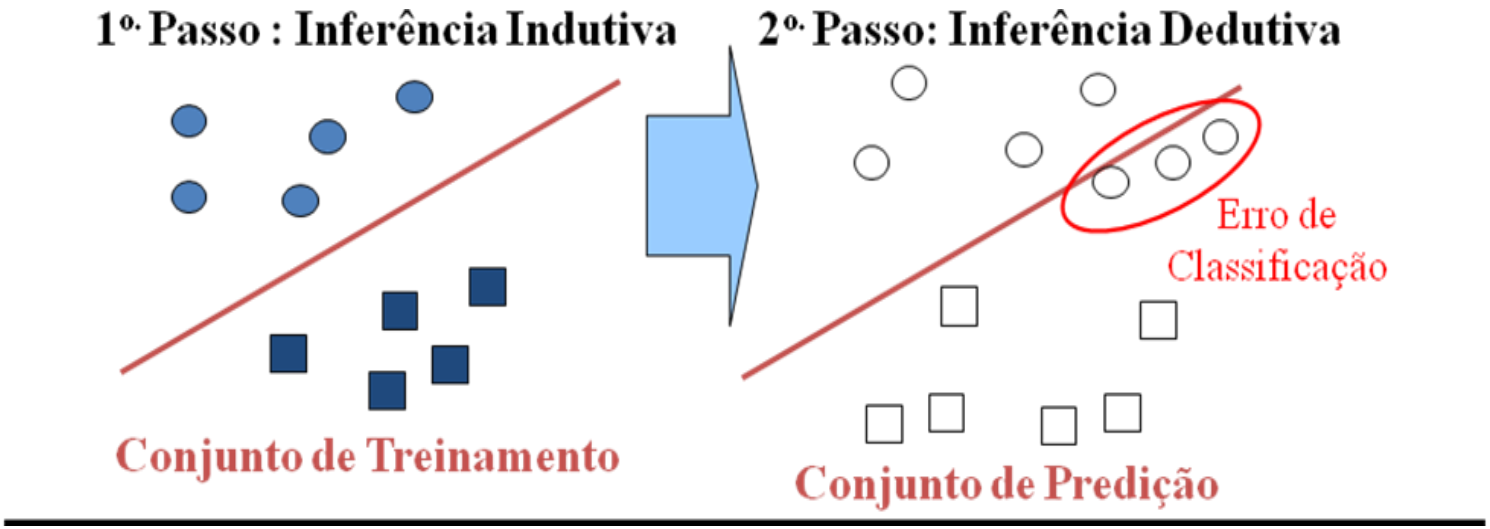



Figura 19 – Aspectos fundamentais da teoria do aprendizado estatístico

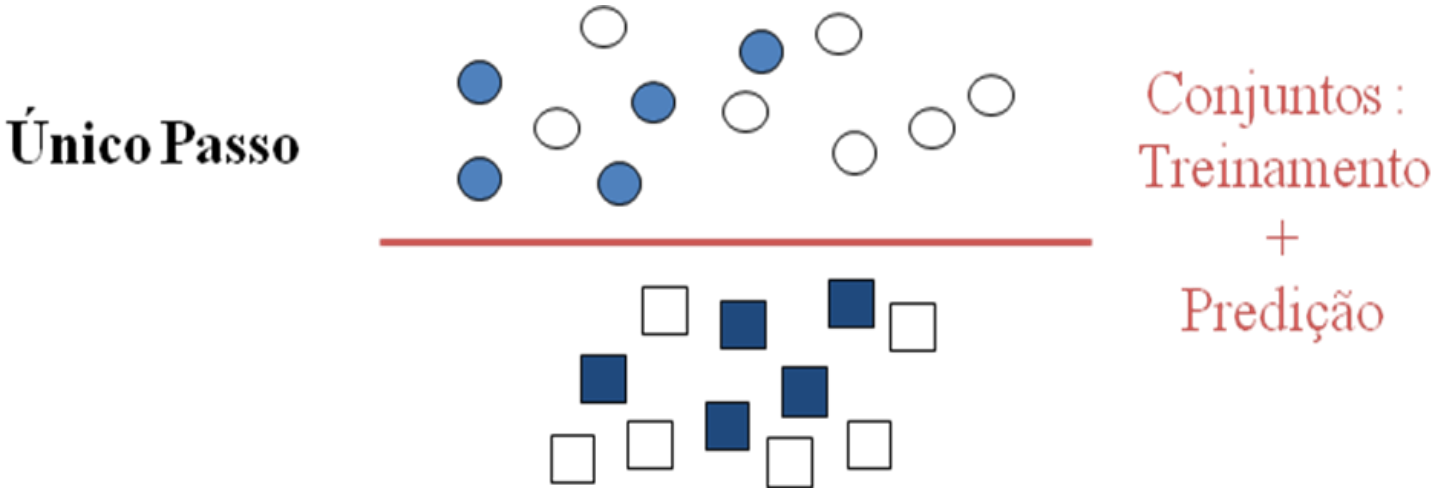

Figura 20 – Aspectos fundamentais da teoria do aprendizado estatístico quando aplicados a máquinas de vetores-suporte

# 11. Inferência transdutiva

- A inferência transdutiva é um método de treinamento semi-supervisionado que representa uma alternativa ao conhecido processo de inferência indutiva-dedutiva.
- No processo de inferência indutiva-dedutiva, primeiro induz-se um modelo de classificação ou regressão (a partir do conjunto de treinamento) e, em seguida, deduz-se a saída para novas amostras de entrada (para o conjunto de predição).



- Já no processo de inferência transdutiva, a inferência do modelo e das saídas para as amostras não-rotuladas se dá simultaneamente.




- Problema primal de otimização associado ao SVM transdutivo: A partir dos dados de treinamento $( \mathbf{x}_i , y_i )_{1 \le i \le N}$ , $\mathbf{x} \in \Re^m$, $y \in \{+1, -1\}$, e dos dados de predição $(\mathbf{x}_j^*)_{1 \le j \le K}$ , encontre o valor do vetor de pesos **w**, intercepto *b*, variáveis de folga $(\xi_i)_{1 \le i \le N}$, $(\xi_j)_{1 \le j \le K}$ e os valores da classificação do conjunto de predição $(y_j^*)_{1 \le j \le K}$, $y^* \in \{+1, -1\}$ que resolvem o seguinte problema:

$$\text{Minimizar} \quad V\left(\mathbf{w}, b, \xi, \xi^*, y^*\right) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i + C^*\sum_{j=1}^{K}\xi_j^*$$

$$\text{Sujeito a} \quad \forall_{i=1}^{N}: \quad y_i\left[\mathbf{w}^T\mathbf{x}_i + b\right] \geq 1 - \xi_i$$

$$\forall_{j=1}^{K}: \quad y_j^*\left[\mathbf{w}^T\mathbf{x}_j^* + b\right] \geq 1 - \xi_j^*$$

$$\forall_{i=1}^{N}: \quad \xi_i \geq 0$$

$$\forall_{j=1}^{K}: \quad \xi_j^* \geq 0$$

- A solução exata do problema primal, ou de seu dual equivalente (não apresentado aqui), requer uma busca sobre todas as $2^K$ possibilidades de classificação do conjunto de predição, visando produzir a SVM com a máxima margem de separação baseada em todo o conjunto de dados $N$+$K$.
- Para um número grande de amostras de predição, deve-se utilizar algum procedimento de busca heurística para encontrar uma boa solução, que até pode ser o ótimo global do problema.

1) Treinar SVM indutiva com os dados de treinamento

2) Classificar dados de predição de acordo com função de decisão (1)

3) Estipular uma baixa influência para os dados de predição

4) Treinar SVM indutiva :

dados de treinamento + predição classificados em (2)

5) Verificar a existência de 2 amostras de predição em que a

troca dos labels diminua o valor da função objetivo

6) Treinar SVM indutiva trocando labels das 2 amostras de pred.

7) Voltar ao passo (5)

8) Aumentar a influência dos dados de predição e voltar passo (4)

Figura 21 – Algoritmo proposto por JOACHIMS (1999)

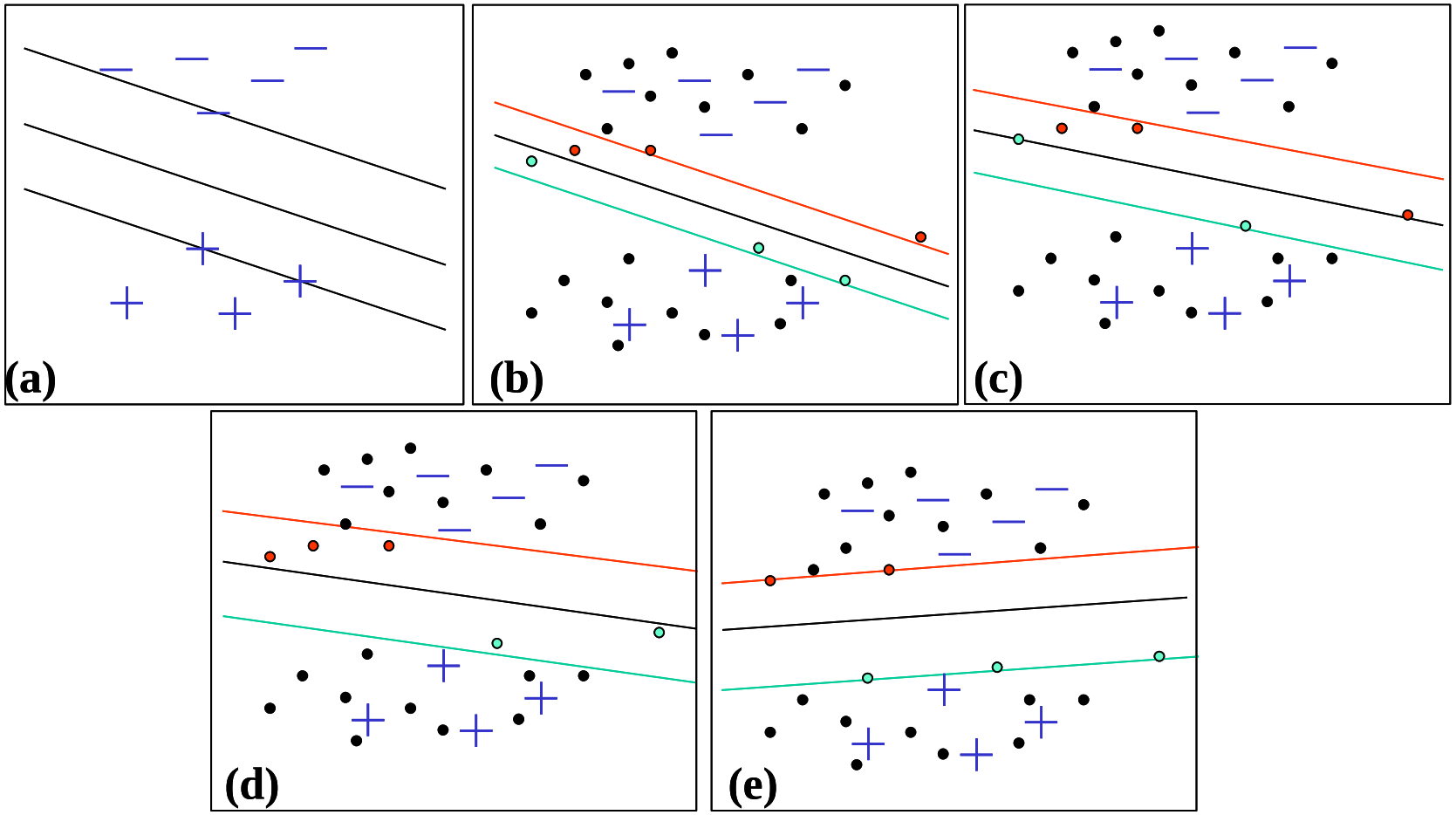



Figura 22 – Análise gráfica do funcionamento do algoritmo de JOACHIMS (1999)

# 12. Referências bibliográficas

ABU-MOSTAFA, Y.S.; MAGDON-ISMAIL, M. & LIN, H.-T. "Learning from data: A short course", AMLBook, 2012.

ARLOT, S.; CELISSE, A. "A survey of cross-validation procedures for model selection", Statistics Surveys, vol. 4, pp. 40–79, 2010.

BURGES, C.J.C. "A tutorial on support vector machines for pattern recognition", Data Mining and Knowledge Discovery, vol. 2, no 2, pp. 121-167, 1998.

CASELLA, G. & BERGER, R.L. "Statistical Inference", 2nd. edition, Duxbury Press, 2001.

CHERKASSKY, V. & MULIER, F. "Learning from Data: Concepts, Theory, and Methods", 2nd edition, Wiley-IEEE Press, 2007.

CORTES, C. & VAPNIK, V.N. "Support vector networks", Machine Learning, vol. 20, pp. 273-297, 1995.

HAYKIN, S. "Neural Networks and Learning Machines", 3rd edition, Prentice Hall, 2008.

JOACHIMS, T. "Transductive Inference for Text Classification using Support Vector Machines", Proceedings of the International Conference on Machine Learning, pp. 200-209, 1999.

LIMA, C.A.M. "Comitê de Máquinas: Uma Abordagem Unificada Empregando Máquinas de Vetores-Suporte", Tese de Doutorado, Faculdade de Engenharia Elétrica e de Computação, Unicamp, 2004.

MITCHELL, T.M. “Machine Learning”, McGraw-Hill, 1997.

MONTGOMERY, D.C.; PECK, E.A.; VINING, G.G. “Introduction to Linear Regression Analysis”, Wiley, 5th Edition, 2012.

SCHÖLKOPF, B.; PLATT, J.C.; SHAWE-TAYLOR, J.; SMOLA, A.J. & WILIAMSON, R.C. “Estimating the support of a high-dimensional distribution”, Microsoft Research Corporation Report MSR-TR-99-87, 1999.

SCHÖLKOPF, B. & SMOLA, A.J. “Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond”, The MIT Press, 2001.

SEMOLINI, R. “Support Vector Machines, Inferência Transdutiva e o Problema de Classificação”, Tese de Mestrado, Faculdade de Engenharia Elétrica e de Computação, Unicamp, 2002.

VAPNIK, V.N. “An overview of statistical learning theory”, IEEE Transactions Neural Networks, vol. 10, pp. 988-999, 1999a.

VAPNIK, V.N. “Estimation of Dependences Based on Empirical Data”, Springer Series in Statistics, 1982.

VAPNIK, V.N. “Statistical Learning Theory”, Wiley-Interscience, 1998.

VAPNIK, V.N. “The Nature of Statistical Learning Theory”, 2nd edition, Springer, 1999b.

VAPNIK, V.N. & CHERVONENKIS, A. Theory of Pattern Recognition (in Russian). Nauka, Moscow, 1974.