

Regularização e Outras Máquinas de Aprendizado

Índice Geral

1	Resgate de conceitos de aprendizado de máquina	3
1.1	Aprendizado supervisionado.....	5
1.2	Modelos de aproximação lineares	7
1.3	Modelos de aproximação fundamentados em redes neurais	8
1.4	Aprendizado baseado em instâncias	11
2	Síntese de mapeamentos multidimensionais.....	12
2.1	As tarefas mais comuns em aprendizado de máquina	12
2.2	Alguns exemplos de mapeamentos multidimensionais	16
3	O dilema bias-variância	22
4	A efetividade não-intuitiva dos dados	33
5	Regularização de funções que aproximam mapeamentos	35
5.1	Regressão de quadrados mínimos.....	36
5.2	Ridge regression	37
5.3	Regressão por função kernel.....	38
6	LASSO	45
7	Elastic Net.....	49
8	Fundamentos da teoria do aprendizado estatístico.....	50

8.1	Risco esperado $\times$ Risco empírico	51
8.2	Aprendizagem em um conjunto finito de funções	57
8.3	Aprendizagem em um conjunto infinito de funções.....	58
8.4	Dimensão Vapnik-Chervonenkis (Dimensão VC)	59
8.5	Dimensão Vapnik-Chervonenkis e aprendizado.....	61
8.6	Minimização do risco estrutural.....	64
8.7	Comentários finais.....	65
9	Máquinas de vetores-suporte.....	68
9.1	Hiperplano ótimo para classes linearmente separáveis	70
9.2	Teorema de Mercer e truque do kernel (<i>kernel trick</i>)	84
9.3	Separabilidade linear com o aumento da dimensão do espaço	86
9.4	Extensão para o espaço de características (<i>feature space</i>).....	88
9.5	Hiperplano ótimo para classes não-linearmente separáveis	90
9.6	Alguns tipos de funções kernel	94
9.7	Extensões para o tratamento de múltiplas classes.....	95
9.7.1	Um contra todos.....	95
9.7.2	Um contra um	96
9.7.3	Grafo direcionado acíclico (DAGSVM)	97
9.8	Extensão para o tratamento de problemas de regressão	98
9.9	Exemplos de comportamento para diferentes ϵ 's.....	101
9.10	Interpretação dos principais resultados	105
9.11	Inferência transdutiva	109
9.12	Support Vector Clustering (SVC).....	113
10	Referências bibliográficas.....	114

1 Resgate de conceitos 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 a uma tarefa em T melhora com uma experiência em E .”
- No contexto de redes neurais artificiais, o 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).
- Fazendo uma mescla com as definições de (MITCHELL, 1997), o aprendizado de máquina faz com que as máquinas se tornem melhores em alguma tarefa ao aprender diretamente dos dados, em lugar de codificar o conhecimento explicitamente em regras do tipo condição-ação.
- Sendo assim, aprendizado de máquina não é necessário 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:
 1. Aprendizado supervisionado, que volta a ser o centro de atenção deste tópico do curso;

2. Aprendizado por reforço, que vai ser melhor formalizado no Tópico 8;
3. Aprendizado não-supervisionado, abordado especificamente no Tópico 6 do curso e que voltará a ser tema de estudo no Tópico 8.

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 melhor solução no 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, *underfitting*), seja porque é flexível demais (podendo produzir o que se convencionou chamar de sobre-ajuste do processo de treinamento, *overfitting*).
- O aprendizado pode ser *offline* em batelada ou padrão-a-padrão, e também existe o aprendizado *online* (*evolving systems*, *concept-drift learning*, *never-ending learning*, *lifelong learning*) para fluxo de dados (*data streams*).

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 a 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).

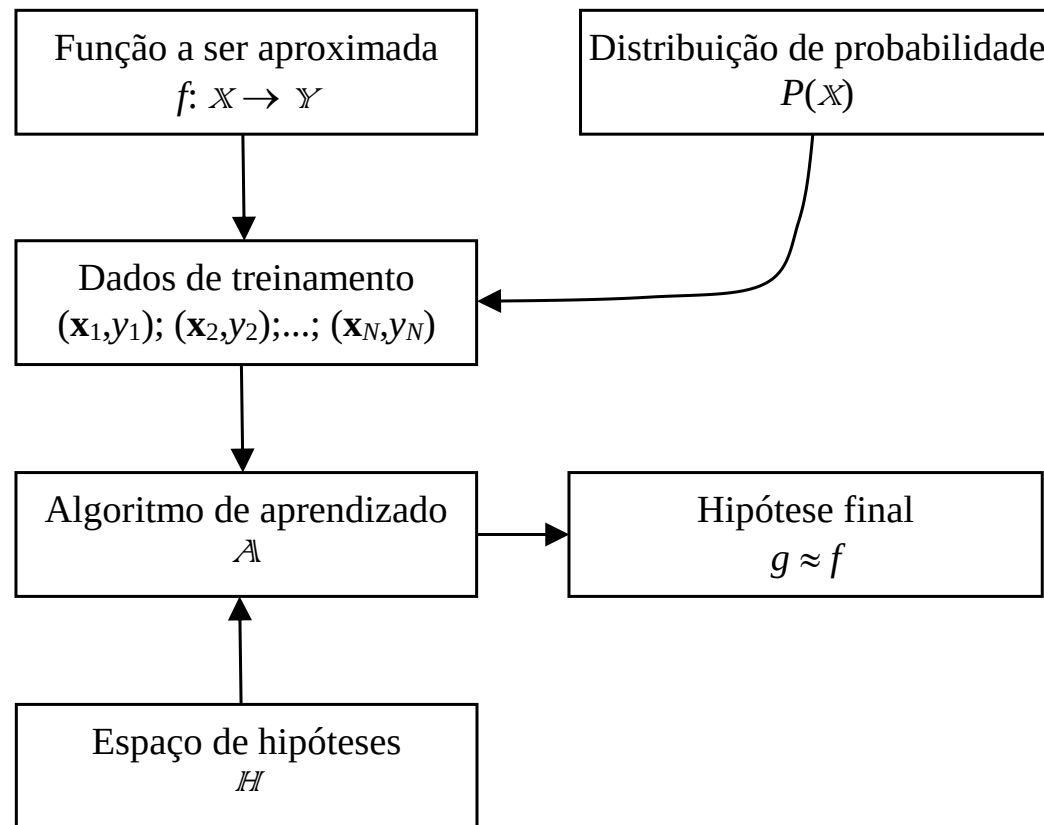


Figura 1 – Conceitos fundamentais em aprendizado supervisionado (baseado em ABU-MOSTAFA et al. (2012))

1.4 Aprendizado baseado em instâncias

- Estamos focados neste curso em aprendizado baseado em modelo (*model-based learning*), mas cabe mencionar que existem as técnicas de aprendizado baseado em instâncias (*instance-based learning*).
- O aprendizado baseado em instâncias (também chamado de aprendizado baseado em memória) é uma família de algoritmos de aprendizado que, em vez de executar uma generalização explícita, faz a inferência por comparação, entre uma nova instância do problema e instâncias armazenadas na memória, ou seja, constrói a hipótese diretamente das instâncias disponíveis, sem passar por um modelo (função de aproximação).
- Exemplos de técnicas: k vizinhos mais próximos e máquinas baseadas em *kernel*, que serão vistas na seção 5.3 (Regressão por função kernel).

2 Síntese de mapeamentos multidimensionais

2.1 As tarefas mais comuns em aprendizado de máquina

- O conteúdo desta seção não deve ser visto como uma taxonomia fechada, mas apenas como uma amostragem de tarefas que vêm sendo resolvidas com sucesso empregando técnicas de aprendizado de máquina.
- Segundo GOODFELLOW et al. (2016), as tarefas mais comuns em aprendizado de máquina são:
 - ✓ **Classificação:** a qual dentre k classes a entrada pertence? Variantes envolvem fornecer a distribuição de probabilidade ao longo das classes.
 - ✓ **Classificação com entradas faltantes:** não havendo a garantia de que todas as variáveis de entradas sejam fornecidas a todo momento, pode-se modelar a distribuição de probabilidade envolvendo todas as variáveis relevantes, de modo a definir a distribuição de probabilidade ao longo das classes. Com isso, é possível empregar a lei da

probabilidade total e marginalizar as variáveis ausentes. De fato, considerando n variáveis de entrada, esta técnica permite obter 2^n diferentes funções de classificação, todas deriváveis de uma única distribuição de probabilidade conjunta. Muitas das demais tarefas de aprendizado de máquina a serem apresentadas a seguir admitem este mesmo tratamento.

Considere duas variáveis aleatórias A e B . Caso se conheça a probabilidade conjunta $P(A=a, B=b)$ para todos os valores possíveis de a e b , pode-se marginalizar a variável A como segue:

$$P(B=b) = \sum_a P(A=a, B=b)$$

- ✓ **Regressão:** a máquina de aprendizado deve predizer um ou mais valores numéricos na saída, dada uma entrada multidimensional. É similar ao problema de classificação, mas difere no tipo de saída.

- ✓ **Transcrição:** a máquina de aprendizado deve observar uma informação não-estruturada e a descrever numa forma estruturada. Por exemplo, a partir da imagem de um texto, deve-se retornar o texto na forma de uma sequência de caracteres. Como um outro exemplo, tem-se o reconhecimento de fala, em que uma máquina de aprendizado recebe na entrada uma onda sonora, a qual deve ser transcrita na saída em formato de texto.
- ✓ **Tradução de máquina:** a entrada já aparece na forma de uma sequência de símbolos em alguma linguagem, sendo que a máquina deve converter esta entrada numa sequência de símbolos de uma outra linguagem, capaz de expressar um conteúdo semântico equivalente.
- ✓ **Saída estruturada:** trata-se de uma ampla classe de aplicações em potencial, onde a saída expressa relações importantes entre seus múltiplos elementos. Transcrição e tradução, descritas acima, se

encaixam aqui também. O exemplo mais impressionante é o recebimento de uma imagem na entrada e sua rotulação na saída pela geração automática de um texto em linguagem natural que descreve o seu conteúdo. Outro exemplo não menos impressionante é a segmentação pixel a pixel de imagens, com cada pixel sendo classificado numa categoria específica.

- ✓ **Detecção de anomalia:** varrer um amplo conjunto de eventos ou objetos e sinalizar quando alguns deles são atípicos. Como um exemplo, tem-se detecção de fraude em operações de cartão de crédito, que envolve a modelagem dos hábitos de compra do cliente.
- ✓ **Síntese e amostragem:** a máquina de aprendizado deve gerar novos exemplos similares àqueles do conjunto de treinamento. Uma possibilidade de aplicação está em videogames, em que se deve gerar uma grande quantidade de conteúdo, por exemplo, em fundo de tela.

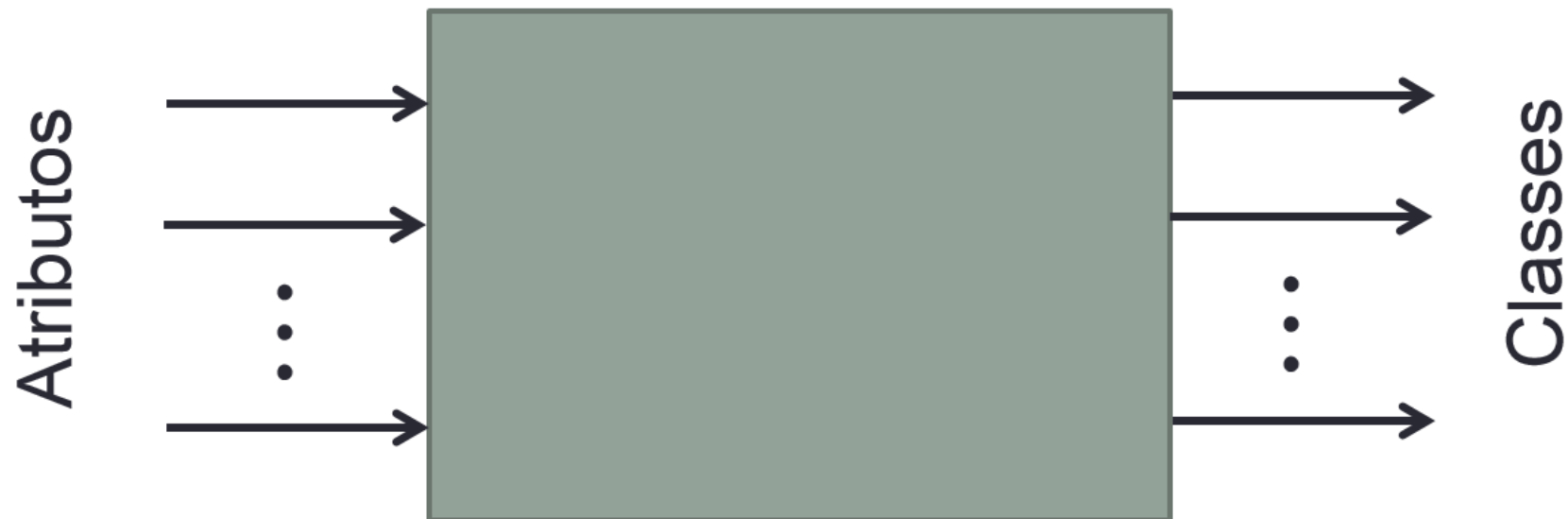
- ✓ **Imputação de dados faltantes:** a máquina de aprendizado deve ser capaz de preencher adequadamente os elementos faltantes da entrada.
- ✓ **Eliminação / atenuação de ruído:** a entrada fornecida é uma versão corrompida de um padrão puro, sendo que não se conhece o processo de corrupção do conteúdo original, e o objetivo é reproduzir este padrão puro na saída.
- ✓ **Estimação de densidade ou estimação da função massa de probabilidade:** a máquina de aprendizado deve aprender a distribuição estatística dos dados vistos na fase de treinamento, tomando as variáveis como sendo contínuas ou discretas. Um exemplo de aplicação está na imputação de dados faltantes.

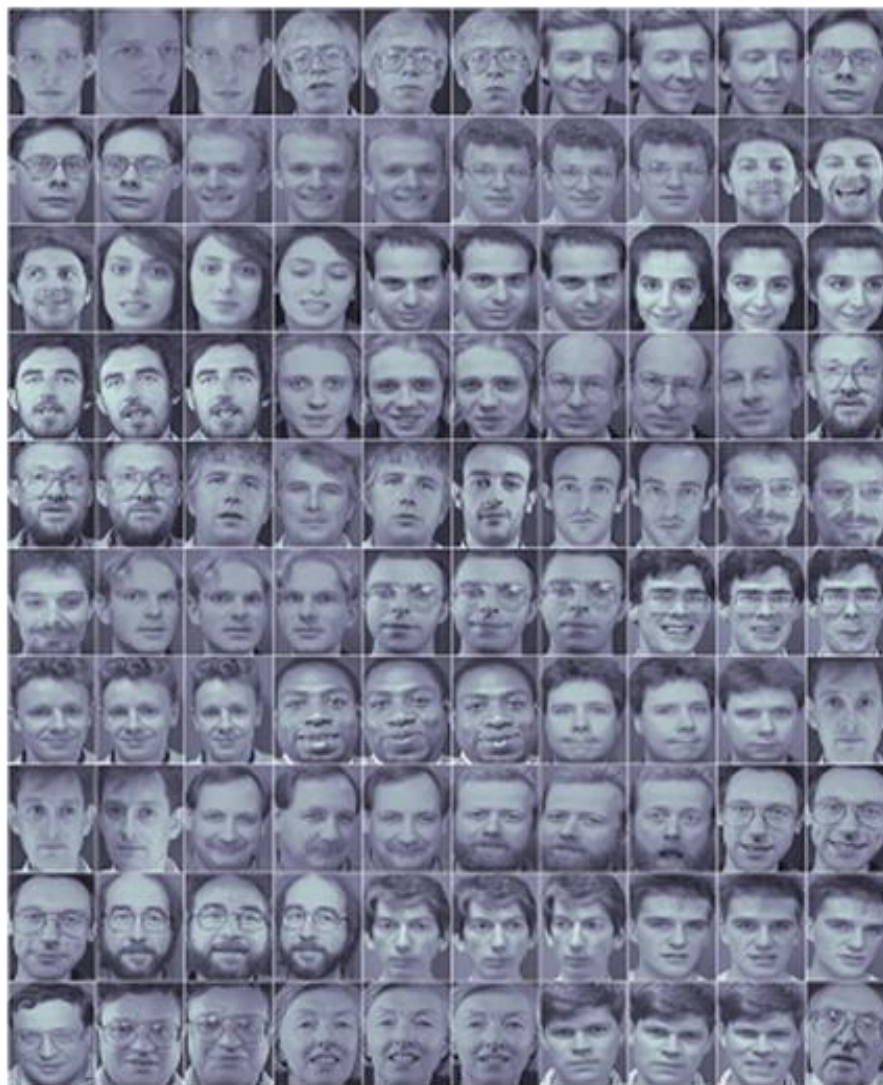
2.2 Alguns exemplos de mapeamentos multidimensionais

- São muitos os problemas da literatura que podem ser resolvidos a partir da síntese de mapeamentos multidimensionais. Seguem alguns exemplos, ainda sem

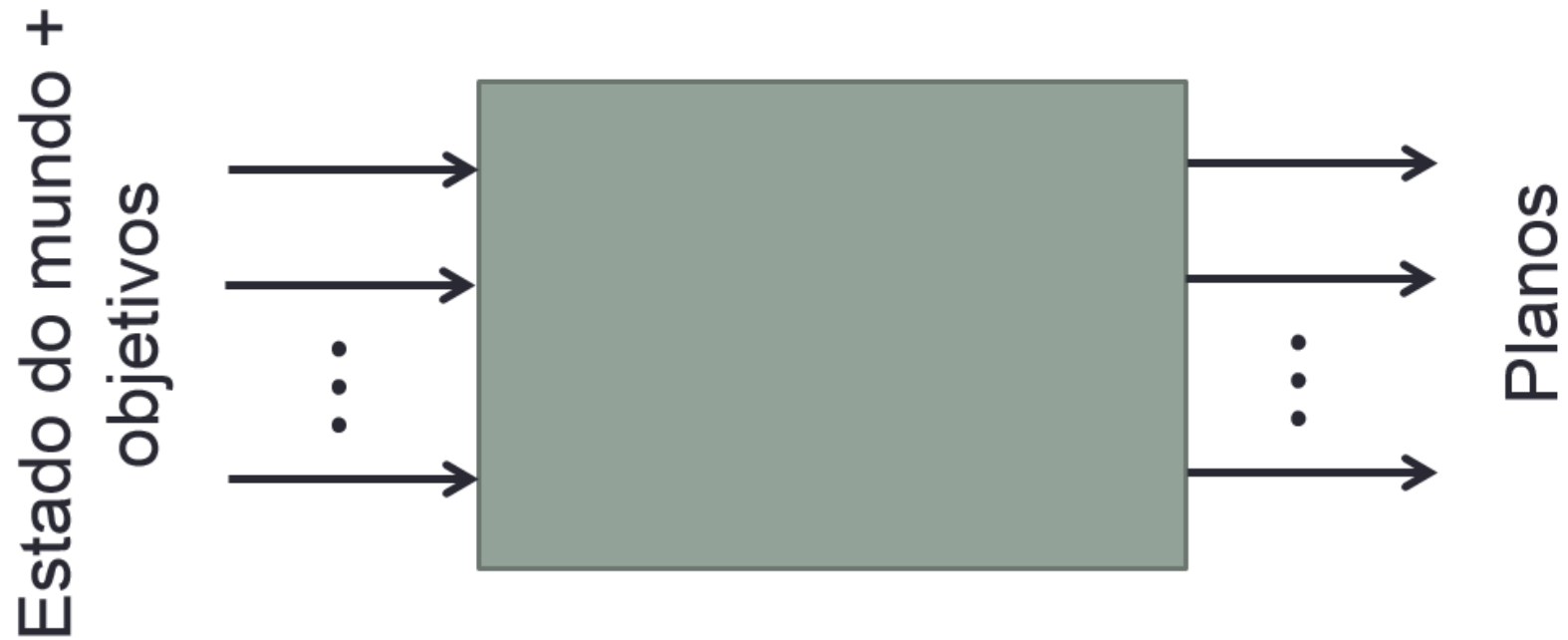
evidenciar a etapa de aprendizado da representação, que pode ser interpretado como a síntese de um mapeamento do espaço original dos dados para um espaço de representação que torna a tarefa de aprendizado mais fácil de ser resolvida.

Reconhecimento de padrões / Classificação:

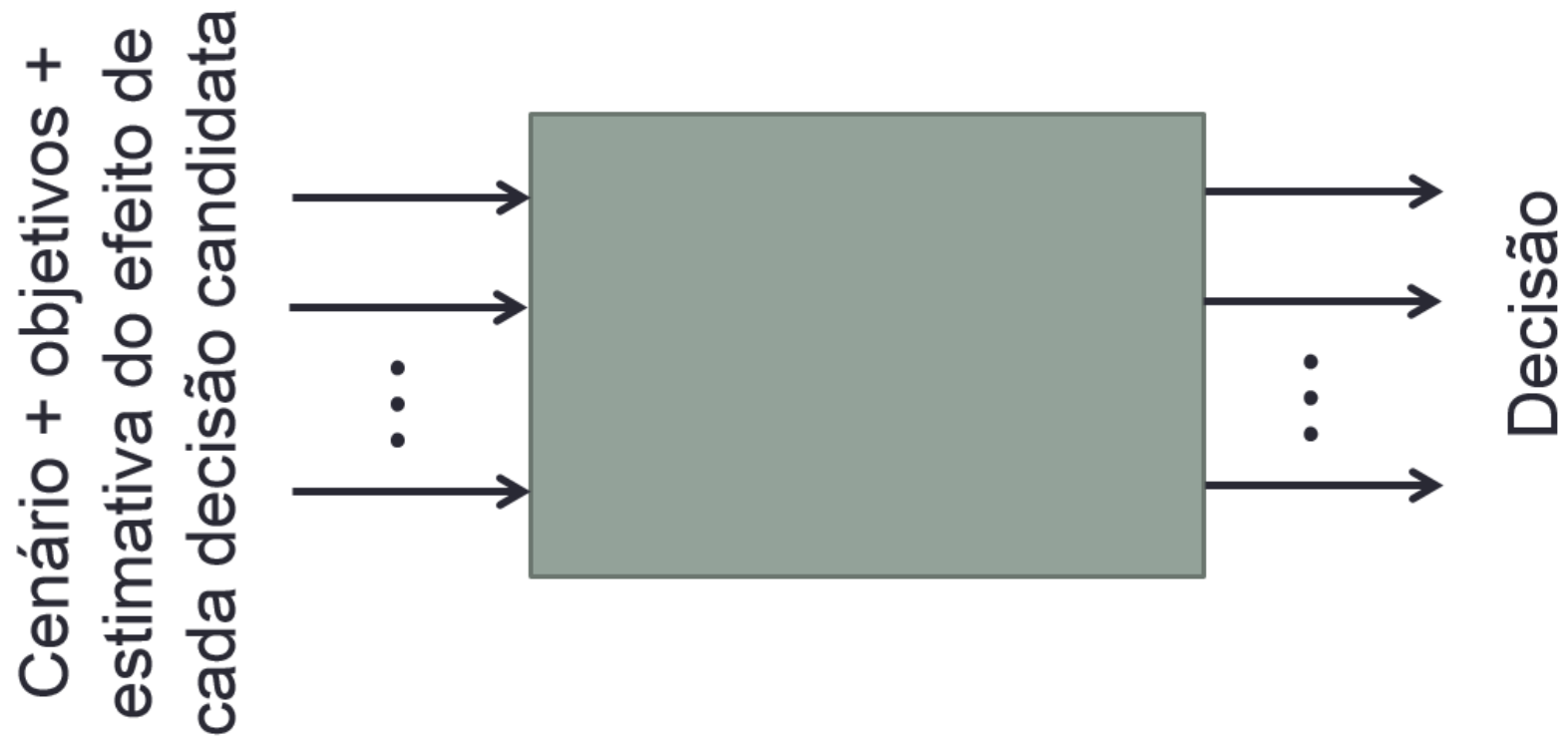




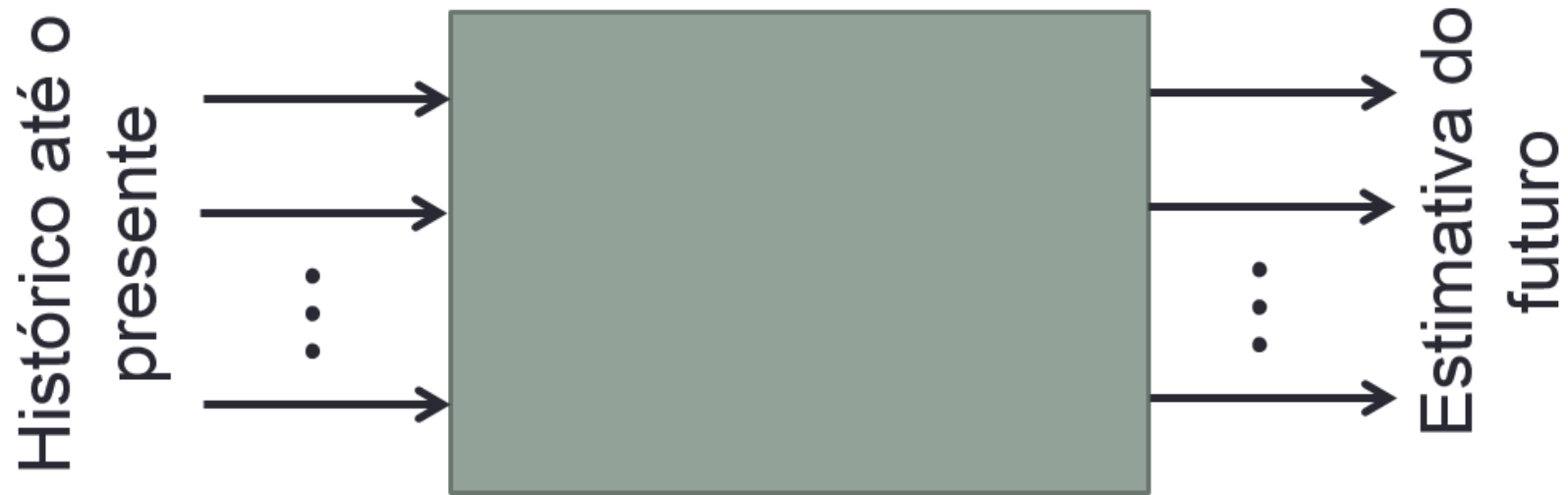
Planejamento:



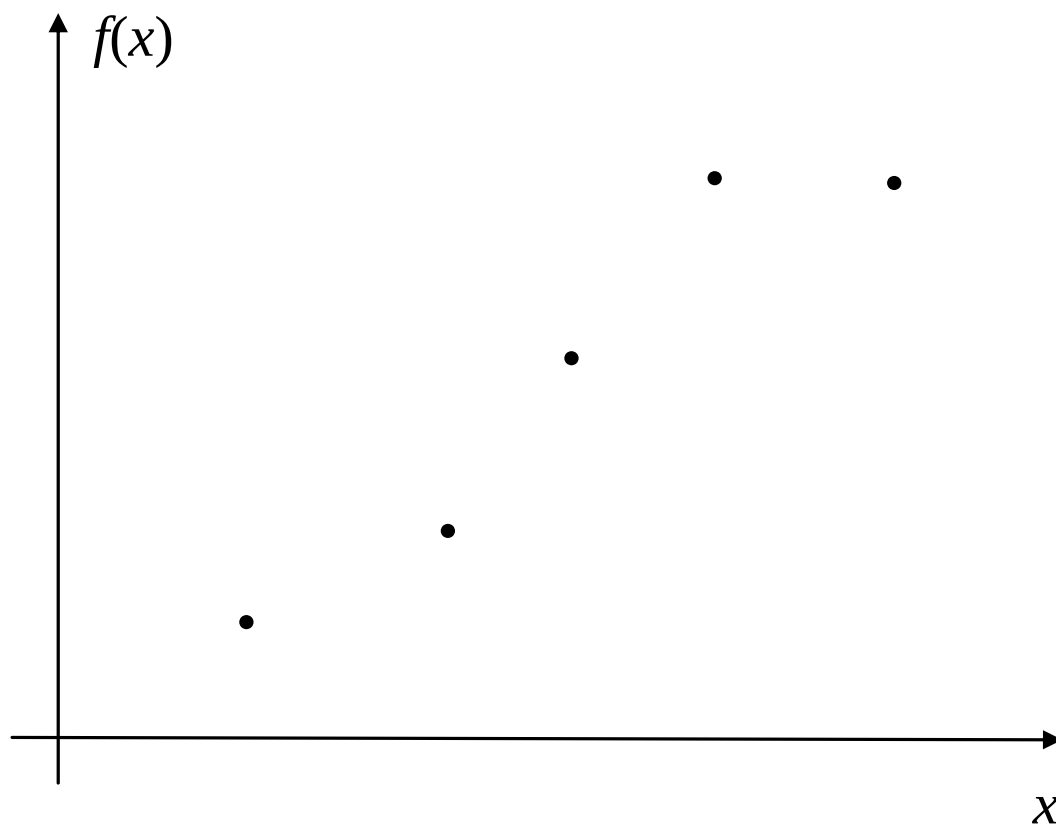
Tomada de decisão:

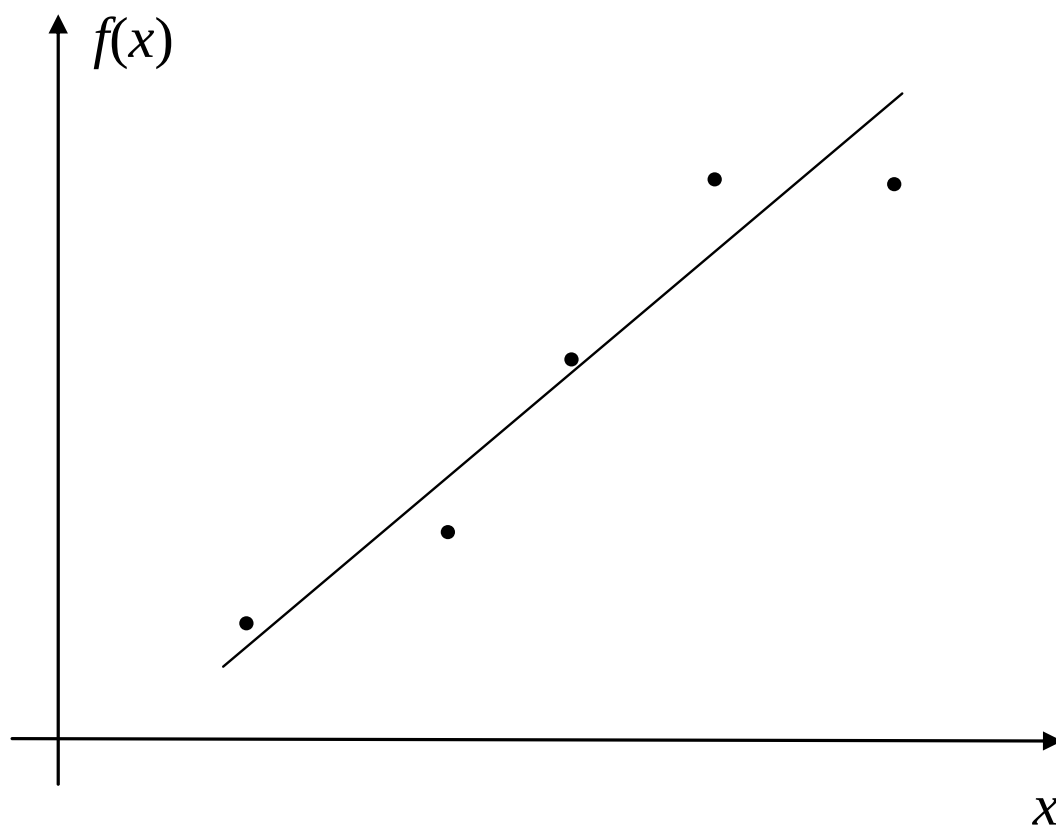


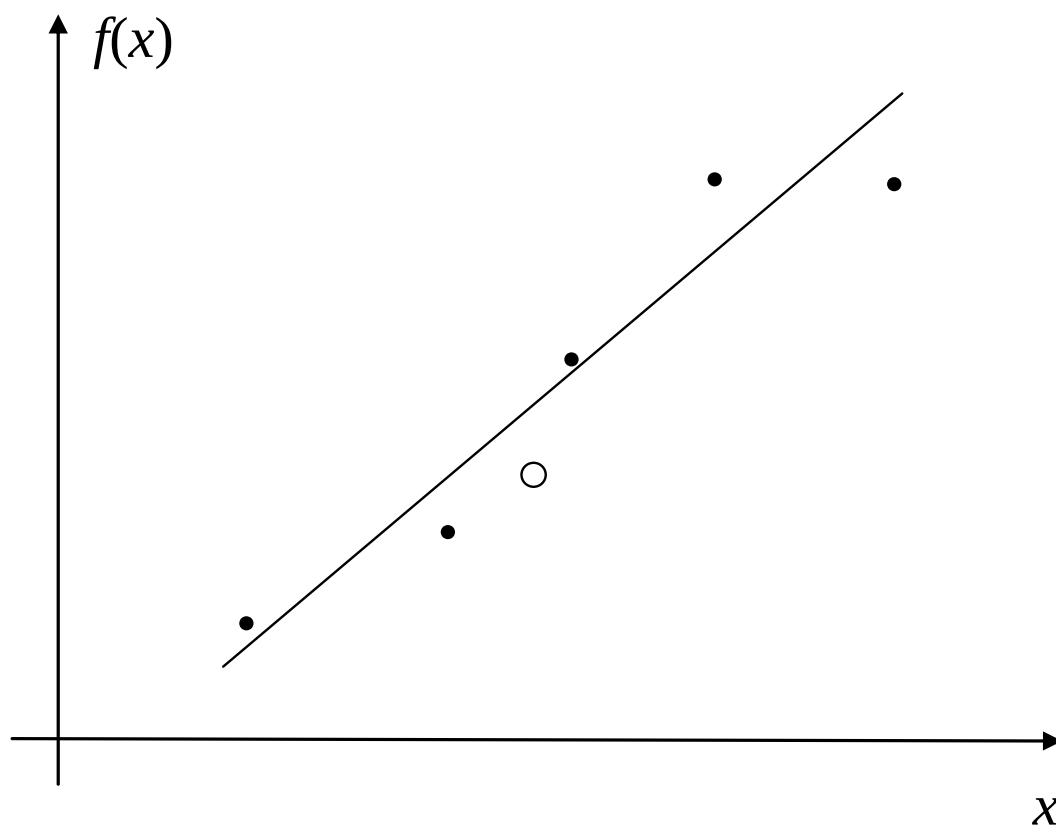
Predição:

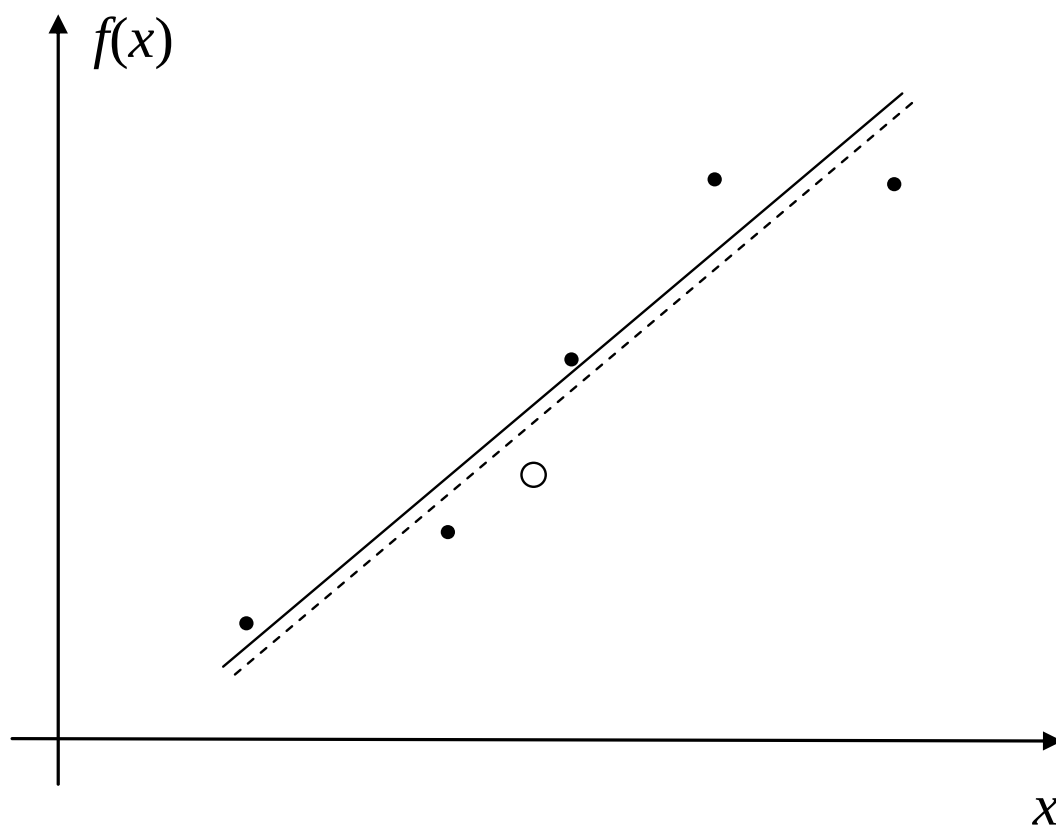


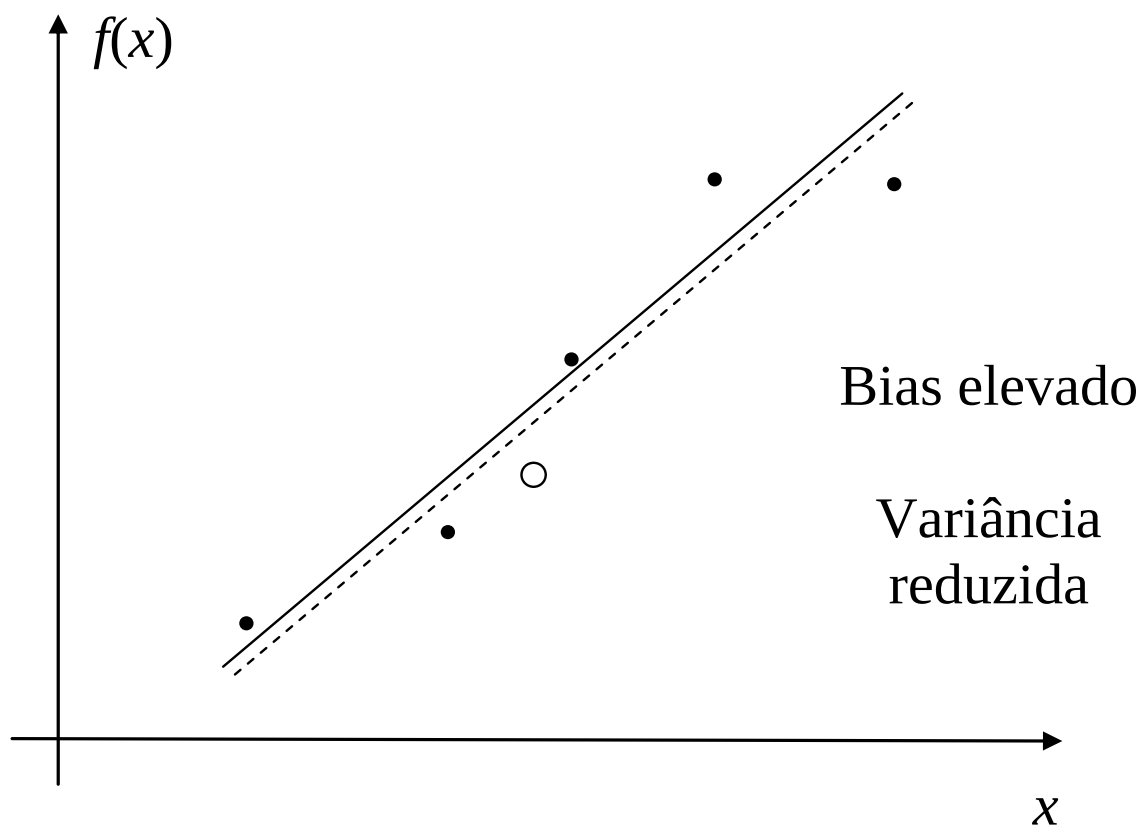
3 O dilema bias-variância

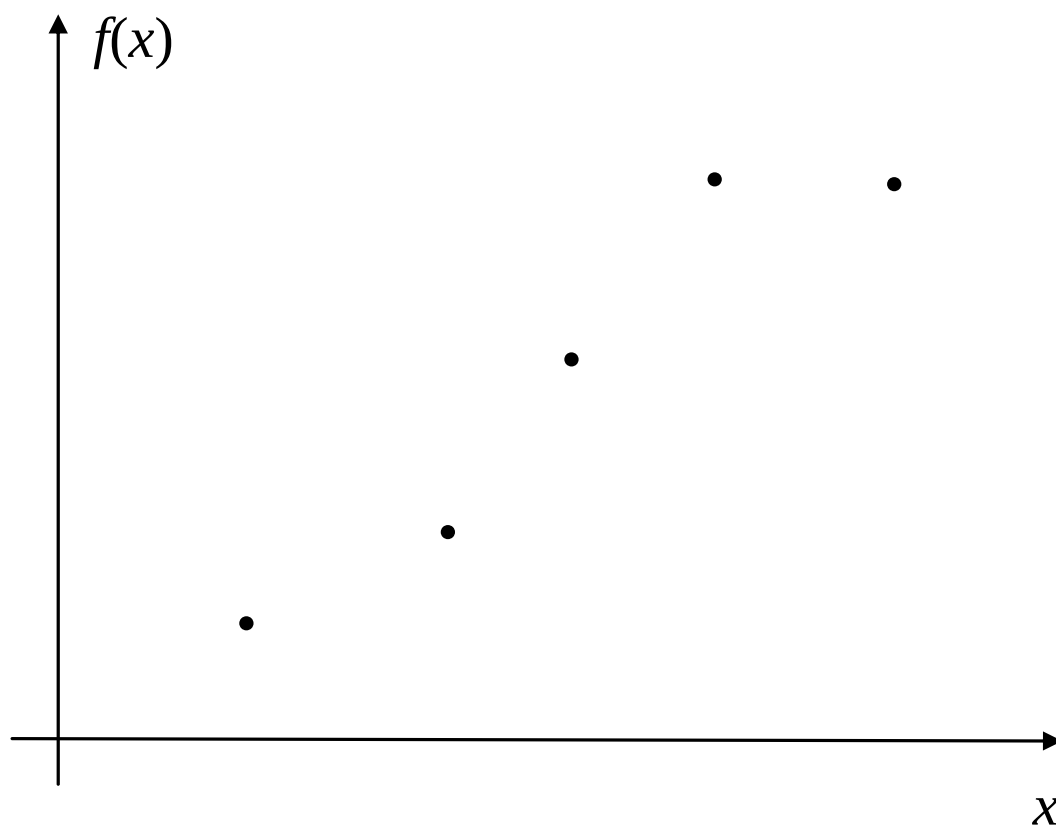


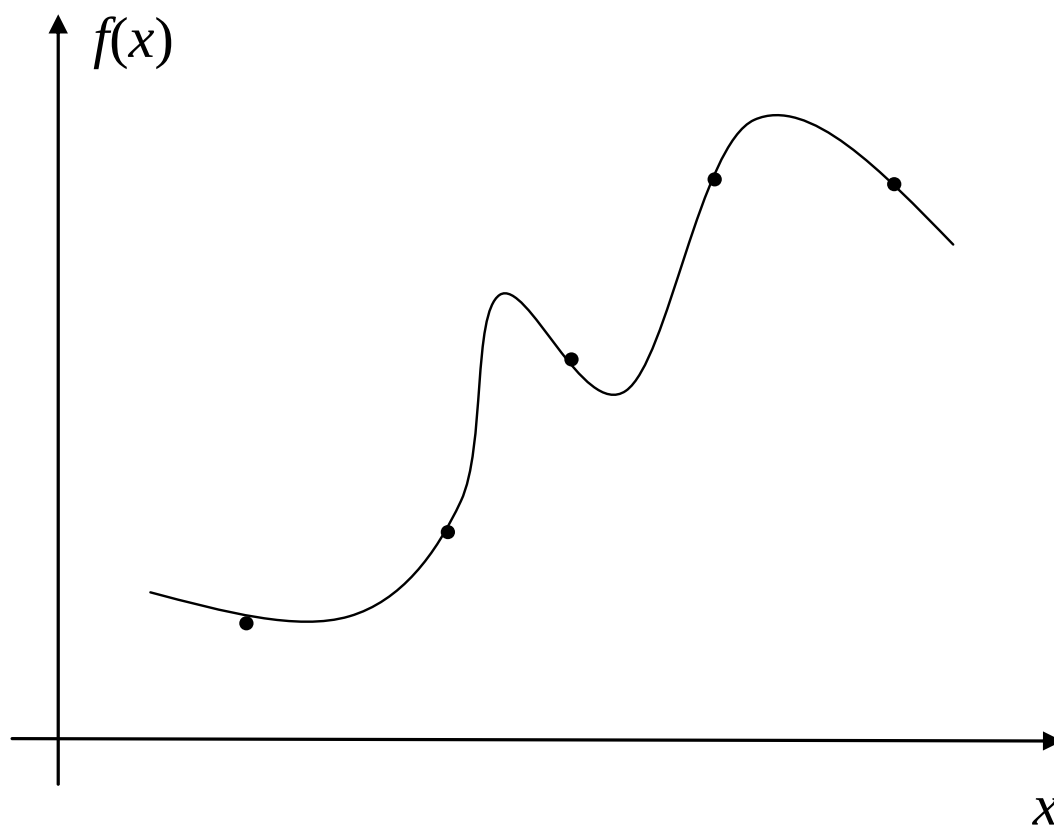


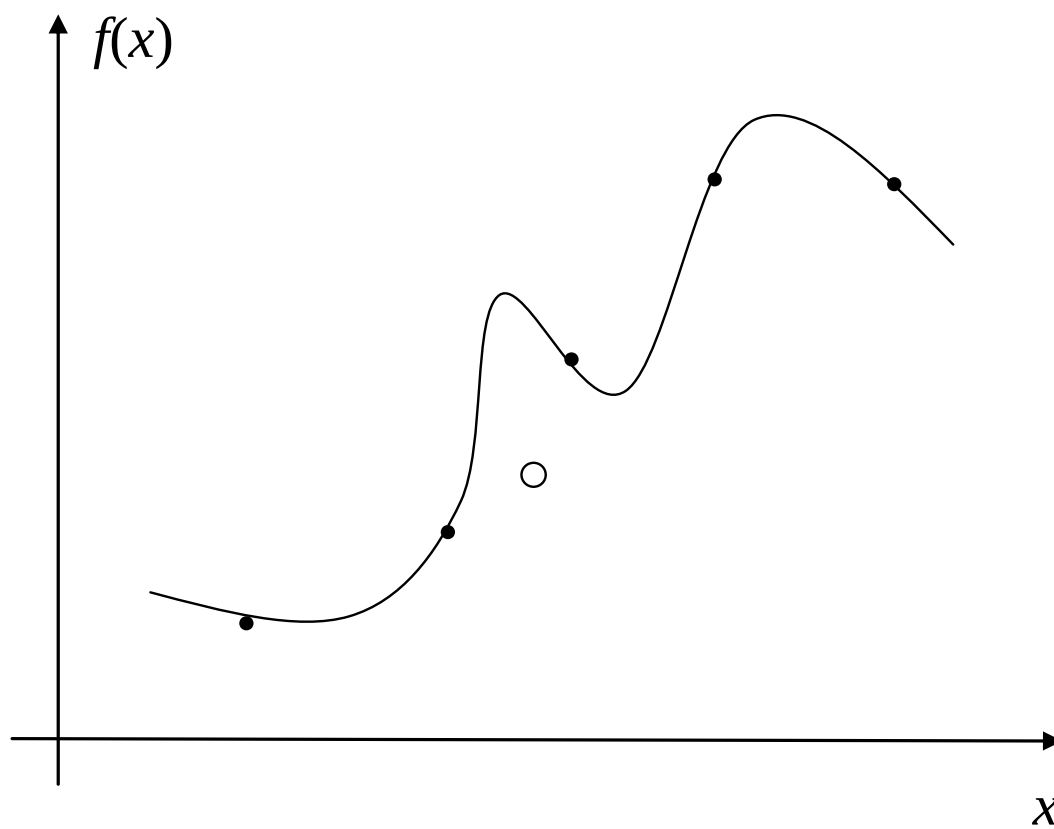


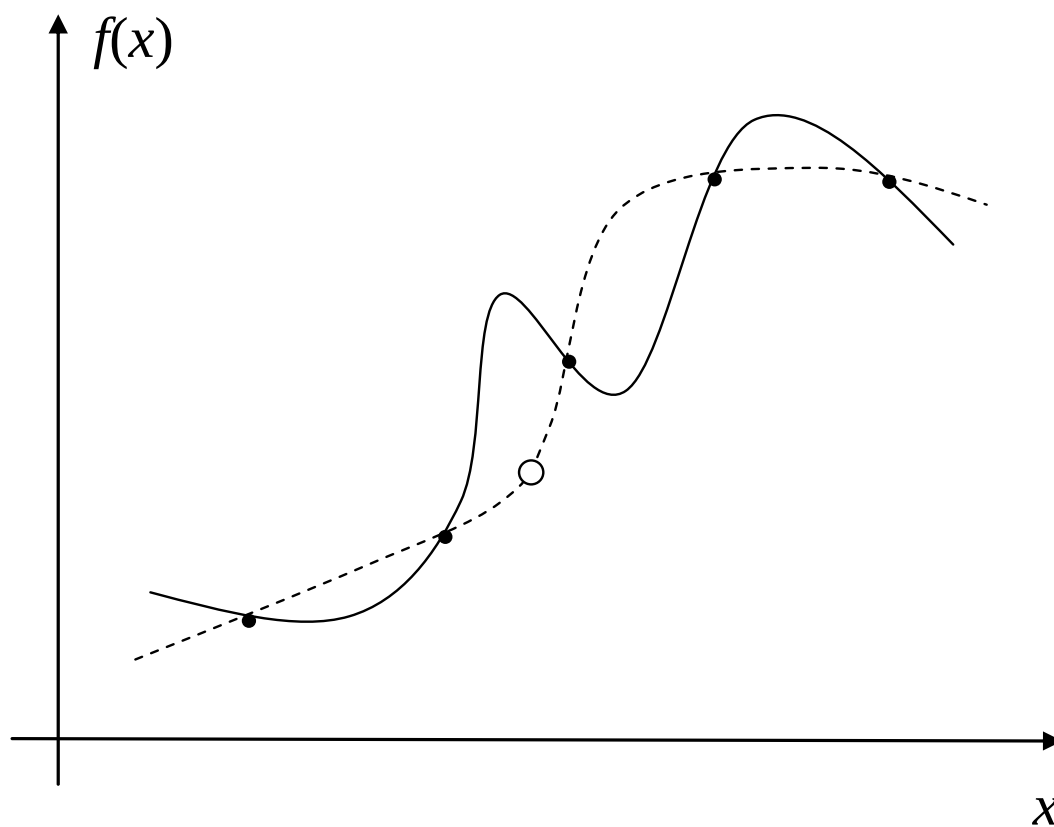


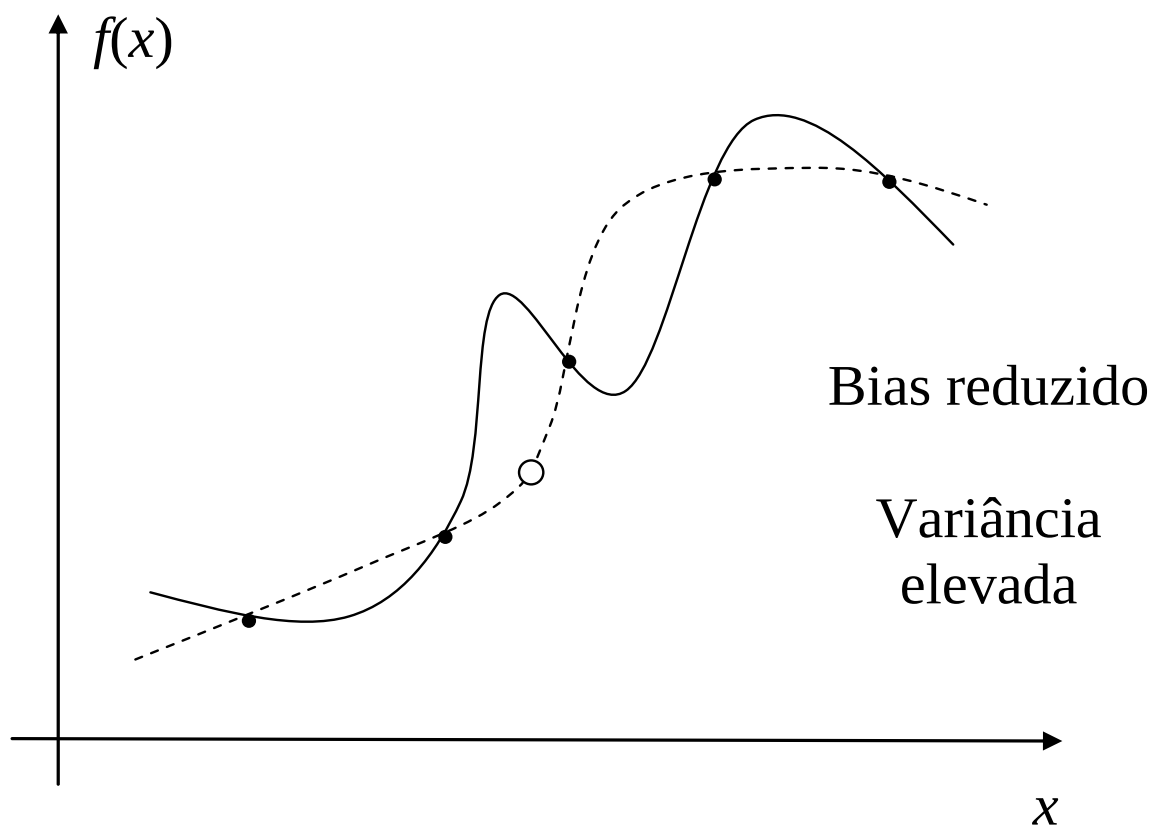


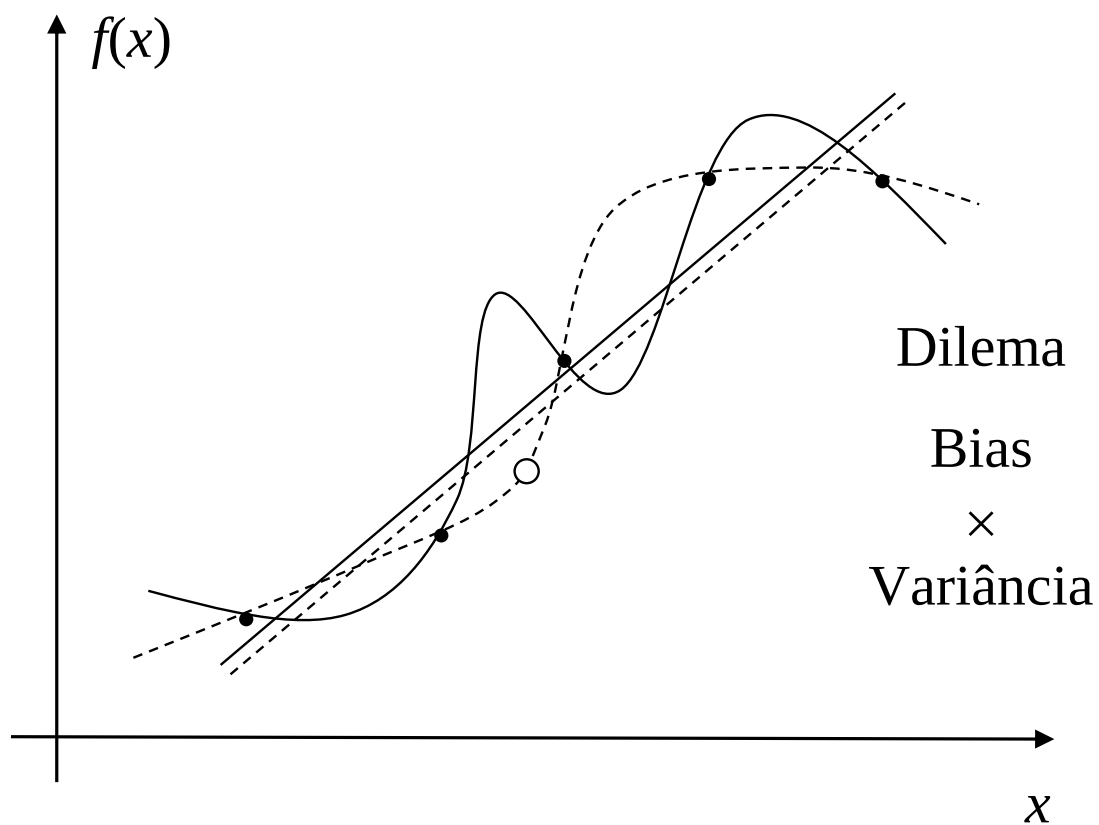








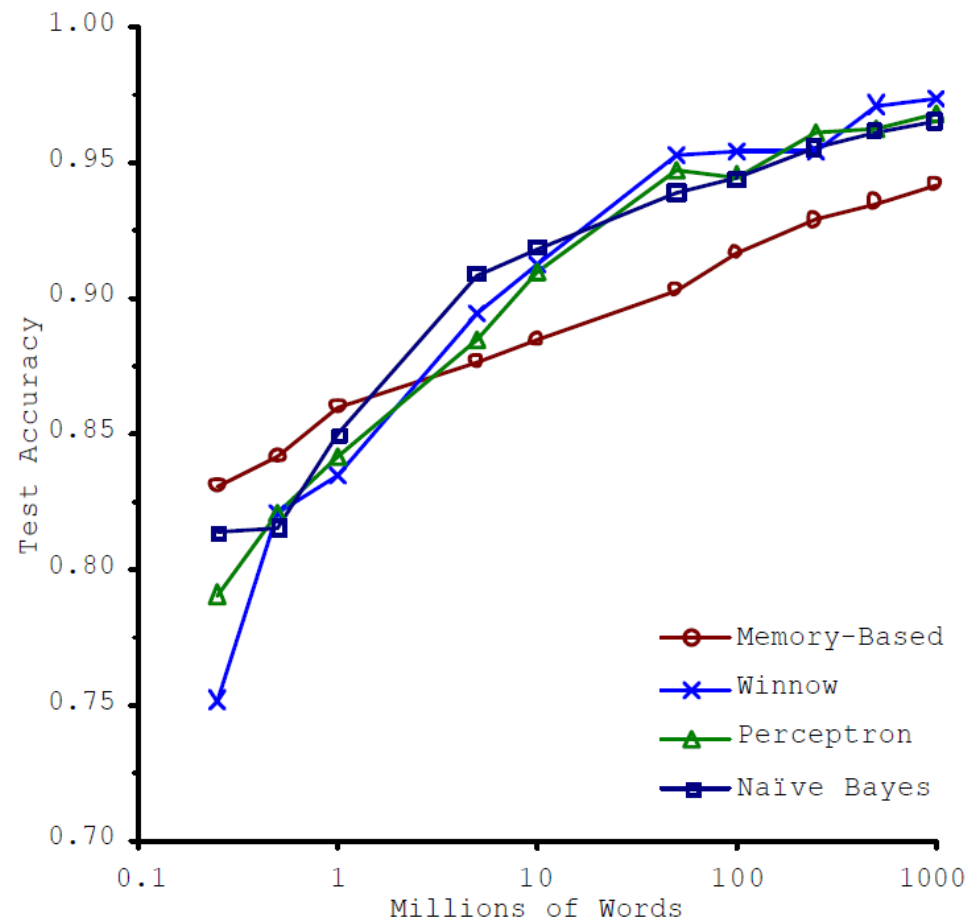




4 A efetividade não-intuitiva dos dados

- Pesquisadores da Microsoft Research chegaram, em 2001, a um resultado surpreendente em aprendizado de máquina: eles concluíram que, havendo dados suficientes, mesmo modelos de aprendizado muito simples produzem desempenho equivalente a modelos de aprendizado mais sofisticados, isso em problemas complexos, como em desambiguação de linguagem natural (Exemplo: ⟨pelo⟩ pode ser preposição ou substantivo).
- Passa a ser questionável então o investimento que se faz no desenvolvimento de modelos de aprendizado mais sofisticados, pois efeitos melhores podem ser produzidos investindo na coleta e preparação de mais dados de treinamento.
- A ideia de que dados importam mais que modelos de aprendizado, para problemas complexos, também foi defendida por HALEVY et al. (2009). No entanto, sabemos que bases de dados pequenas e médias são ainda muito comuns, justificando um maior investimento em modelos de aprendizado.

- Banko, M.; Brill, E. “Scaling to very very large corpora for natural language disambiguation”, Proceedings of the 39th Annual Meeting on Association for Computational Linguistics, pp. 26–33, 2001 (<https://doi.org/10.3115/1073012.1073017>).



5 Regularização de funções que aproximam mapeamentos

- Há outras formas de impor suavidade quando aproximando funções a partir de dados amostrados, além de *ridge regression* e LASSO, mas iremos dar ênfase a essas duas técnicas.
- Para funções unidimensionais, são indicadas algumas técnicas, como (HASTIE et al., 2009):
 - ✓ *k*-vizinhos mais próximos;
 - ✓ *Splines* polinomiais suavizantes;
 - ✓ Polinômios de Hermite com um número reduzido de funções-base;
 - ✓ Funções *kernel*.
- No entanto, a grande maioria dos mapeamentos que se quer sintetizar envolvem múltiplas dimensões e cabe investigar técnicas voltadas para o atendimento de demandas nesta frente.

5.1 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 $\mathbf{x}_i \in \mathbb{R}^n$, $i=1, \dots, N$. Suponha também que $N > n$.

- A regressão de quadrados mínimos busca um vetor $\mathbf{w} \in \mathbb{R}^n$ 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):

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

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

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

5.2 Ridge regression

- Tomando o mesmo cenário da Seção 5.1, é possível adicionar um termo de regularização, que penaliza o crescimento da norma do vetor $\mathbf{w}$, produzindo o problema regularizado de regressão de quadrados mínimos, também denominado de *ridge regression*, na forma:

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

- Aplicando a condição necessária de otimalidade, como feito na Seção 5.1, obtém-se como solução ótima:

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

- 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:

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

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

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

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

5.3 Regressão por função kernel

- Tomando o modelo de regressão em sua forma dual da Seção 5.2 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:

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

- Repare que $\mathbf{x}_i \in \mathcal{R}^n$, $i=1,\dots,N$ e também $\mathbf{x} \in \mathcal{R}^n$ aparecem sempre aos pares, em produtos internos.
- Suponha agora que exista um mapeamento não-linear $\Phi: \mathcal{R}^n \rightarrow S$, 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 $\kappa: \mathcal{R}^n \times \mathcal{R}^n \rightarrow \mathcal{R}$, 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:

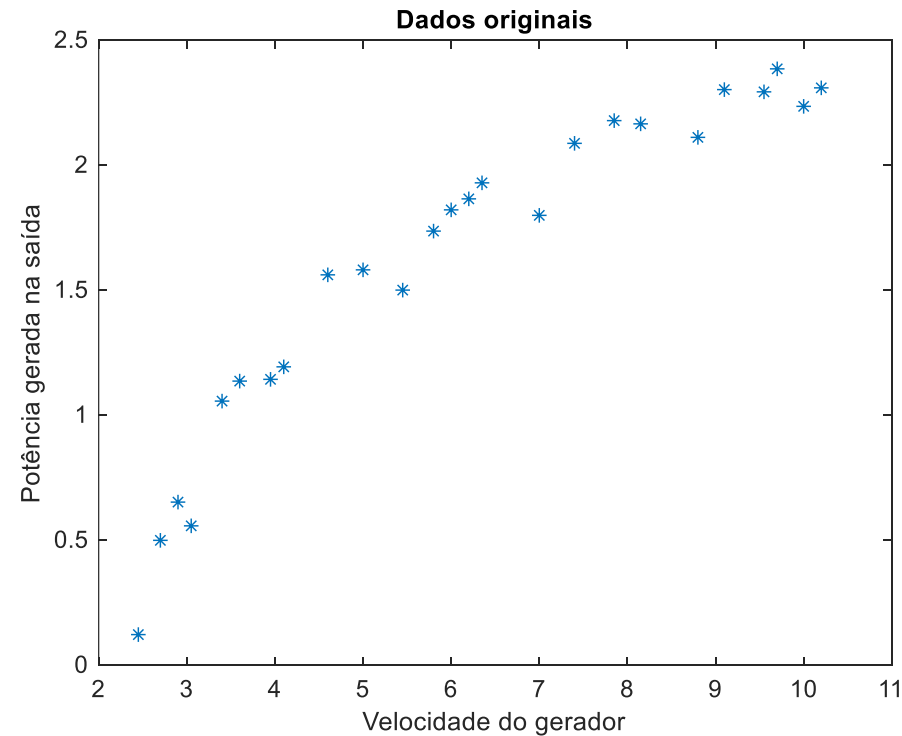
$$\mathbf{k}(\mathbf{x}) = \begin{bmatrix} \kappa(\mathbf{x}_1, \mathbf{x}) \\ \kappa(\mathbf{x}_2, \mathbf{x}) \\ \vdots \\ \kappa(\mathbf{x}_N, \mathbf{x}) \end{bmatrix},$$

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

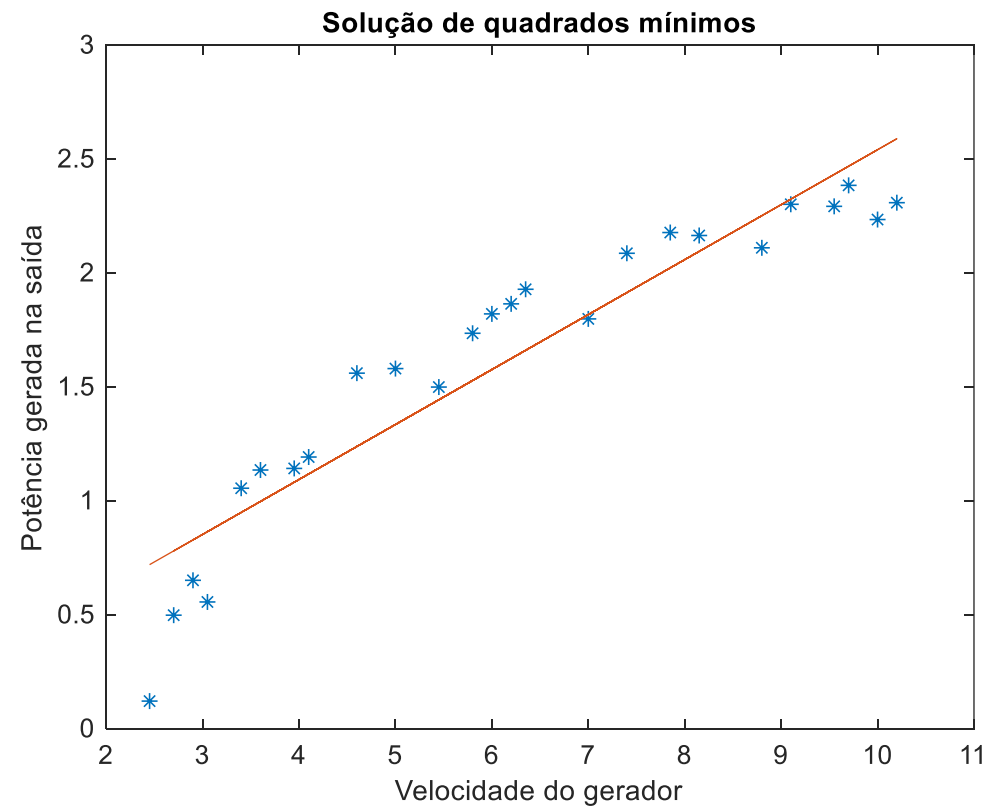
$$RR(\mathbf{x}) = \mathbf{y}^T (K + \lambda I)^{-1} \mathbf{k}(\mathbf{x}).$$

- Este último resultado representa um salto de qualidade em regressão não-linear, pois se ganha 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*.

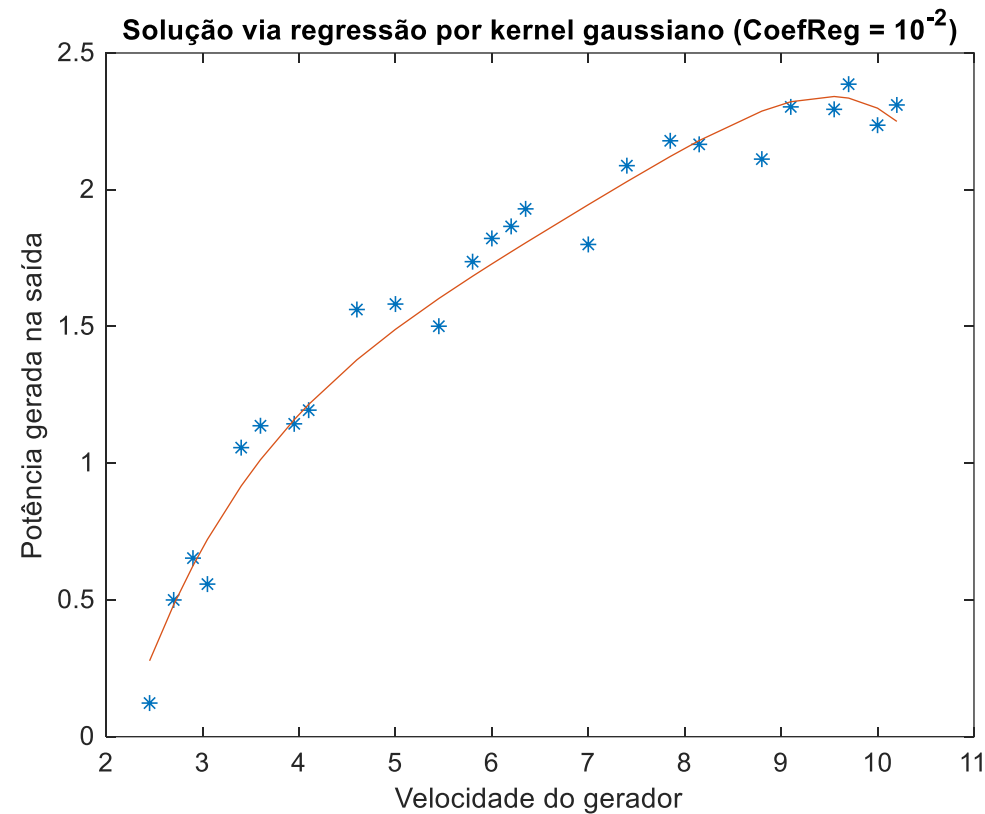
- Outras formas de conceber máquinas de aprendizado baseadas no truque do *kernel* estão associadas ao conceito de máquinas de vetores-suporte (VAPNIK, 1999b), que serão vistos mais adiante.
- Caso de estudo (MONTGOMERY et al., 2012):



- 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:



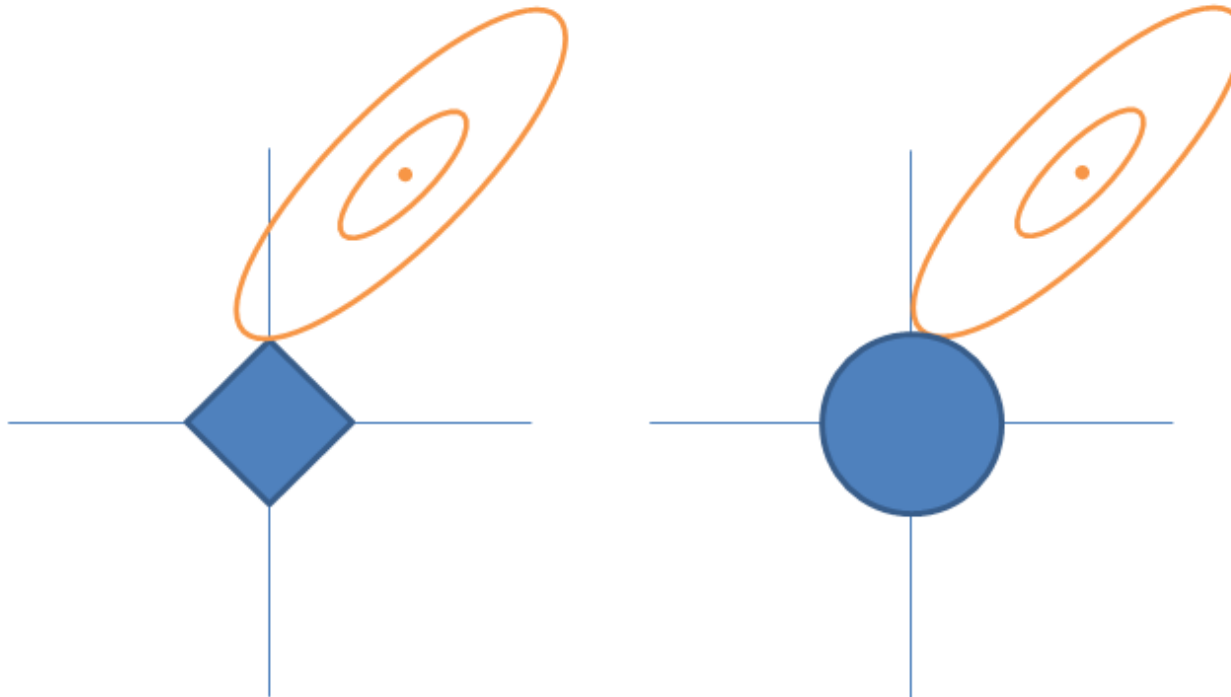
6 LASSO

- Uma formulação alternativa considera a penalização com a norma l_1 , produzindo:

$$\min_{\theta \in \mathbb{R}^n} J(\theta) + \lambda \|\theta\|_1, \text{ com } \|\theta\|_1 = \sum_{i=1}^n |\theta_i| \text{ e } \lambda \geq 0.$$

- Esta estratégia de regularização é conhecida como LASSO, do inglês *Least Absolute Shrinkage and Selection Operator*, e foi proposta por Tibshirani em 1996: Tibshirani, R. “Regression shrinkage and selection via the LASSO”, *Journal of the Royal Statistical Society B*, vol. 58, pp. 267–288, 1996.
- A vantagem do LASSO está principalmente no fato de que múltiplos termos do vetor θ acabam sendo anulados, o que representa um processo de **seleção de variáveis** e leva a modelos de aproximação mais parcimoniosos.
- A motivação geométrica que permite justificar a anulação de um subconjunto de parâmetros ajustáveis é apresentada na figura a seguir. Nesta figura, apresentam-se

os lugares geométricos de valores constantes para $J(\theta)$ e $\|\theta\|_1$, à esquerda, e os lugares geométricos de valores constantes para $J(\theta)$ e $\|\theta\|_2$, à direita.



Inspirado na interpretação geométrica presente em HASTIE et al. (2009).

- O problema de otimização associado é convexo e existem *solvers* eficientes para LASSO.

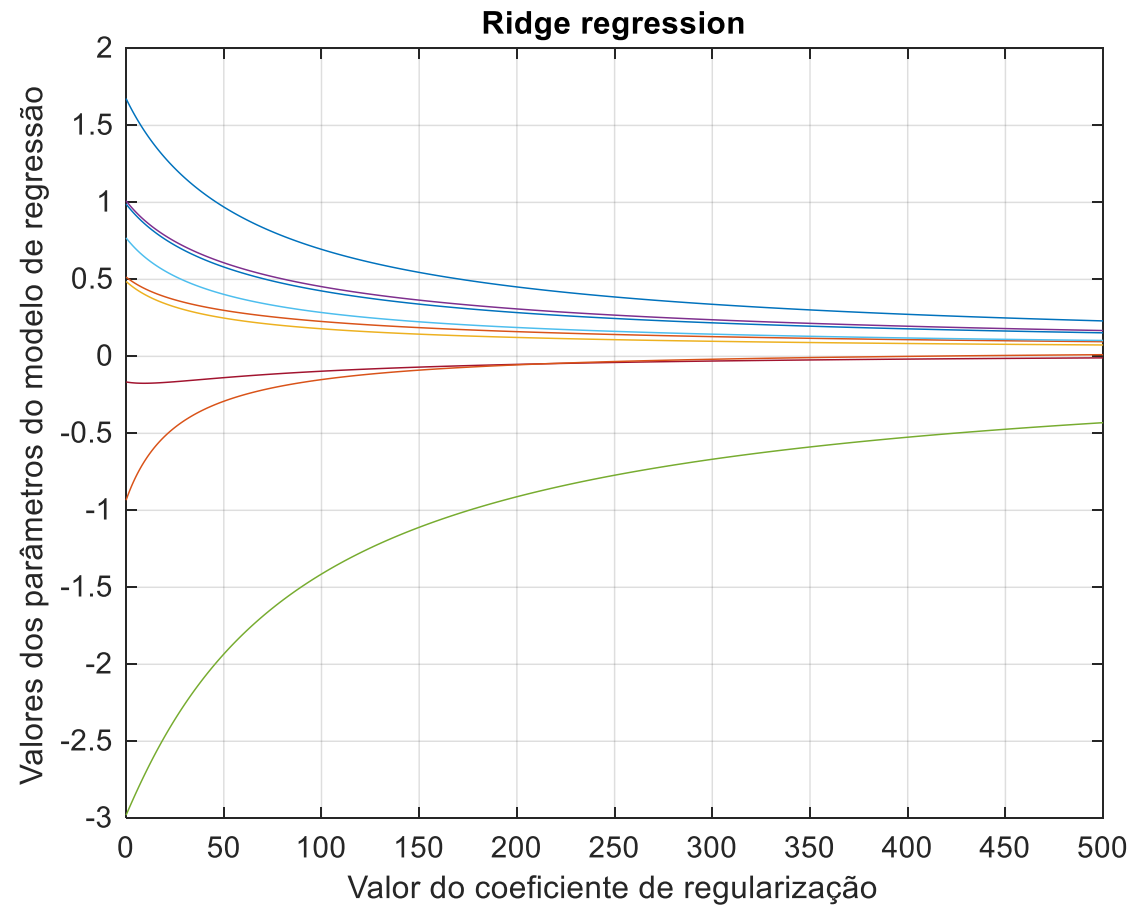


Figura 2 – Efeito sobre o valor dos parâmetros do modelo de regressão quando se aumenta o valor do coeficiente de regularização em *ridge regression*.

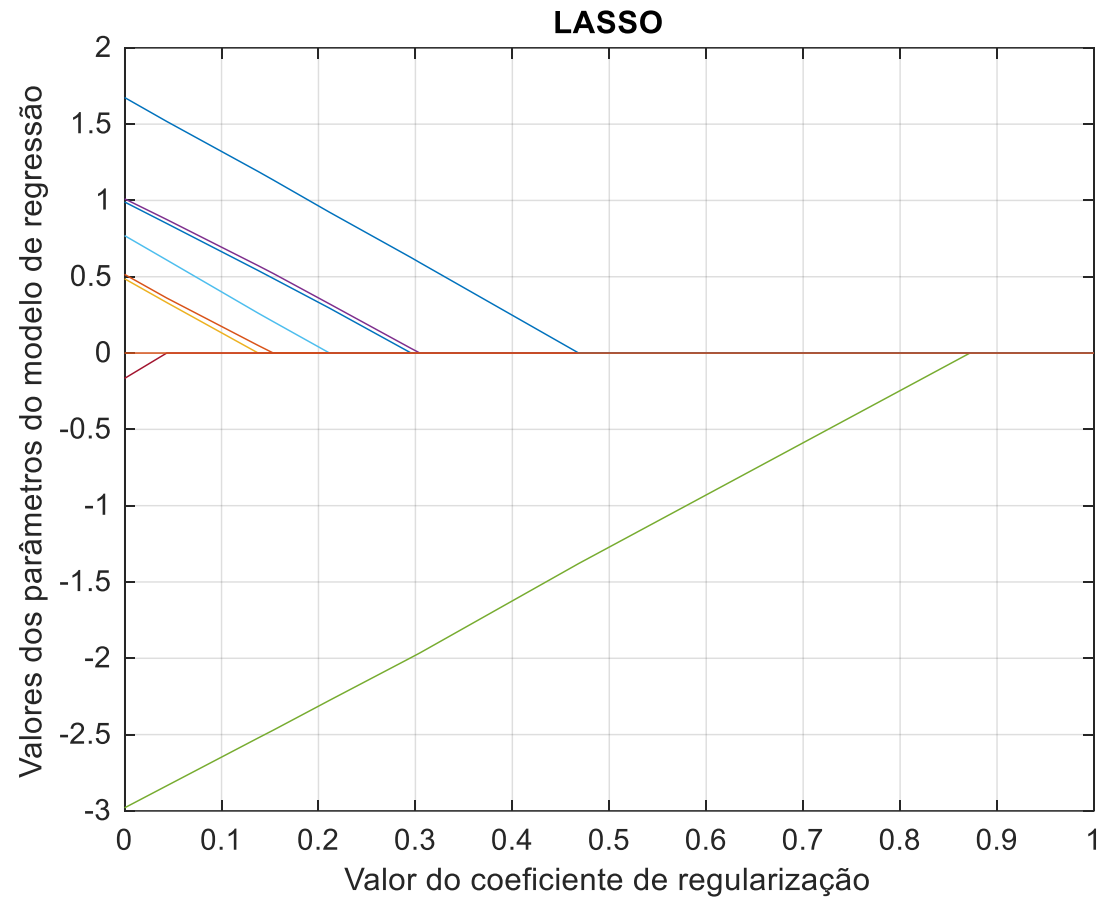


Figura 3 – Efeito sobre o valor dos parâmetros do modelo de regressão quando se aumenta o valor do coeficiente de regularização em LASSO.

- Comportamento conveniente para implementar seleção de variáveis.

7 Elastic Net

- Uma solução intermediária entre *ridge regression* e LASSO é a Elastic Net, proposta em: Zou, H., Hastie, T. “Regularization and variable selection via the elastic net”, *Journal of the Royal Statistical Society B*, vol. 67, pp. 301–320, 2005.
- No caso da *Elastic Net*, o problema de otimização regularizado assume a forma:

$$\min_{\theta \in \mathbb{R}^n} J(\theta) + \lambda \left[\alpha \|\theta\|_1 + (1 - \alpha) \|\theta\|_2^2 \right], \text{ com } \lambda \geq 0 \text{ e } \alpha \in [0, +1].$$

- O cálculo dos coeficientes λ e α requer uma busca visando minimizar o erro junto a um conjunto de dados de validação.
- Repare que a *Elastic Net* contempla todas as demais propostas, para valores específicos de λ e α :
 - ✓ Quadrados mínimos irrestrito: $\lambda = 0$;
 - ✓ *Ridge Regression*: $\lambda > 0$ e $\alpha = 0$;
 - ✓ LASSO: $\lambda > 0$ e $\alpha = 1$;
 - ✓ *Elastic net*: $\lambda > 0$ e $0 < \alpha < 1$.

8 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.

8.1 Risco esperado × Risco empírico

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

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

em que $\mathbf{z}_i \in Z$, $i=1,\dots,N$, 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 $\mathbf{x}_i \in \mathcal{R}^m$ 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 Ψ .

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

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

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:

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

- 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 $\mathbf{z}$.
- 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 Ψ .

✓ Por ser um problema de classificação binária, y e $F(\mathbf{x}, \alpha)$, para qualquer α , só podem assumir os valores -1 e $+1$. Logo, $Q(\mathbf{z}, \alpha)$ será uma função indicadora, assumindo apenas os valores 0 e $+4$ (segundo a Eq. (2)), os quais podem ser adequadamente redefinidos para 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:

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

em que $\mathbf{z}_i \in \Psi$, $i=1, \dots, N$, 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 α , mas se deve garantir que o mínimo do risco empírico $I_{emp}(\alpha)$ se aproxime do mínimo do risco esperado $I(\alpha)$. Ou seja, o α que minimiza $I_{emp}(\alpha)$ deve estar próximo do α que minimiza $I(\alpha)$. No entanto, não se sabe a priori onde este α se encontra, justificando buscar a aproximação desses dois funcionais para todo α .

- 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_∞ , 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 $\rho_{L_2}(f_1(\mathbf{x}), f_2(\mathbf{x}))$ 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 4 a seguir, numa situação arbitrária, mas possível.

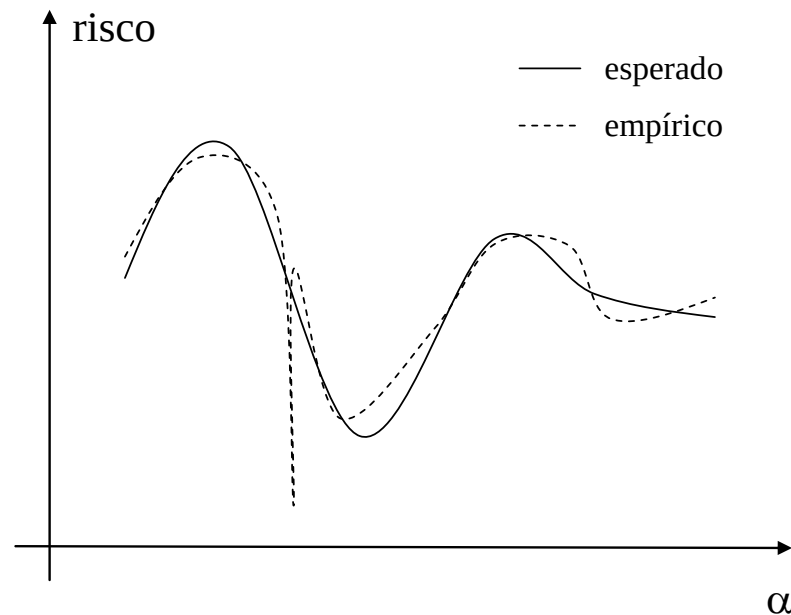


Figura 4 – Situação possível em que o mínimo do risco empírico difere significativamente do mínimo do risco esperado.

- Na Figura 4, $\rho_{L_2}(f_1(\mathbf{x}), f_2(\mathbf{x}))$ assume um valor reduzido quando comparado ao valor assumido por $\rho_{L_\infty}(f_1(\mathbf{x}), f_2(\mathbf{x}))$.
- Para evitar situações como a ilustrada na Figura 4, deve-se exigir que $I_{emp}(\alpha)$ não se distancie de $I(\alpha)$ mais que um valor dado δ , para todo α , produzindo:

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

- Sendo assim, valendo a relação:

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

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.

8.2 Aprendizagem em um conjunto finito de funções

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

$$\mathfrak{F} = \{F(\mathbf{x}, \alpha_1), F(\mathbf{x}, \alpha_2), \dots, F(\mathbf{x}, \alpha_l)\} \quad (8)$$

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 $\mathfrak{F}$ 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 α , 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:

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

- Como l e η são constantes, então:

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

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

8.3 Aprendizagem em um conjunto infinito de funções

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

8.4 Dimensão Vapnik-Chervonenkis (Dimensão VC)

- Em linhas gerais e em termos práticos, a dimensão Vapnik-Chervonenkis (VC) de um espaço de hipóteses $\mathcal{H}$ pode ser interpretada como o número **efetivo** de parâmetros de modelos pertencentes a este espaço de hipóteses $\mathcal{H}$, sendo denotada por $h = d_{VC}(\mathcal{H})$.
- Outro resultado prático muito relevante, o qual será apresentado mais adiante, é que quando $h = d_{VC}(\mathcal{H})$ é finita, então modelos pertencentes a $\mathcal{H}$ 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, $h = d_{VC}(\mathcal{H})$ está associada ao maior número de pontos que um modelo pertencente a $\mathcal{H}$ pode classificar corretamente, para toda proposta de rotulação binária possível desses pontos. Diz-se então que $\mathcal{H}$ **particiona** esses pontos.

- Sendo assim, quando se tem $d_{VC}(\mathcal{H}) \geq Q$, então existe algum conjunto de Q pontos que é particionado por $\mathcal{H}$. Por outro lado, quando $d_{VC}(\mathcal{H}) < Q$, então é possível afirmar que não existe nenhum conjunto de Q pontos que pode ser particionado por $\mathcal{H}$.

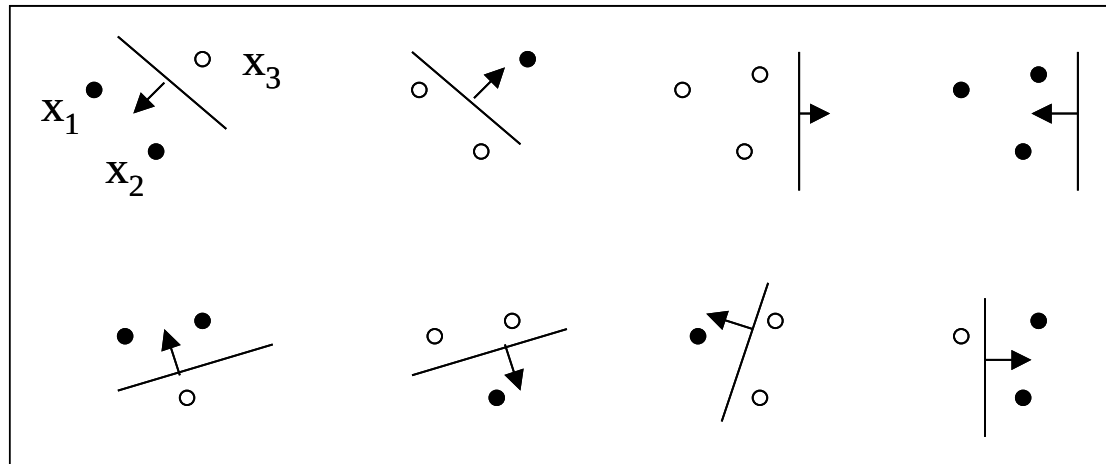


Figura 5 – Possibilidades de rotulação de três amostras no $\mathcal{R}^2$ e a classificação realizada por uma função linear (LIMA, 2004).

- Considerando um neurônio do tipo McCulloch & Pitts, com duas entradas, então a Figura 5 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.

8.5 Dimensão Vapnik-Chervonenkis e aprendizado

- Supondo que dois modelos de classificação são equivalentes quando eles produzem a mesma classificação para a amostra Ψ , 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 Ψ .

- 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 Ψ 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:

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

onde $\varepsilon = b_1 \frac{\left[\ln\left(b_2 \frac{N}{h}\right) + 1 \right] - \ln\left(\frac{\eta}{4}\right)}{N/h}$. 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 ε cresce rápido quando se aumenta o grau de confiança. De fato, quando $\eta \rightarrow 0$ tem-se que $\varepsilon \rightarrow \infty$, 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:

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

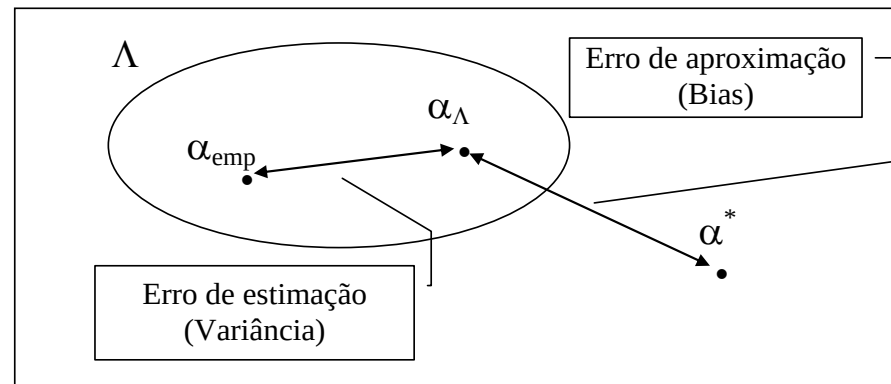
8.6 Minimização do risco estrutural

- 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 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 também que não há uma dependência explícita da dimensão m dos dados de entrada.

8.7 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).

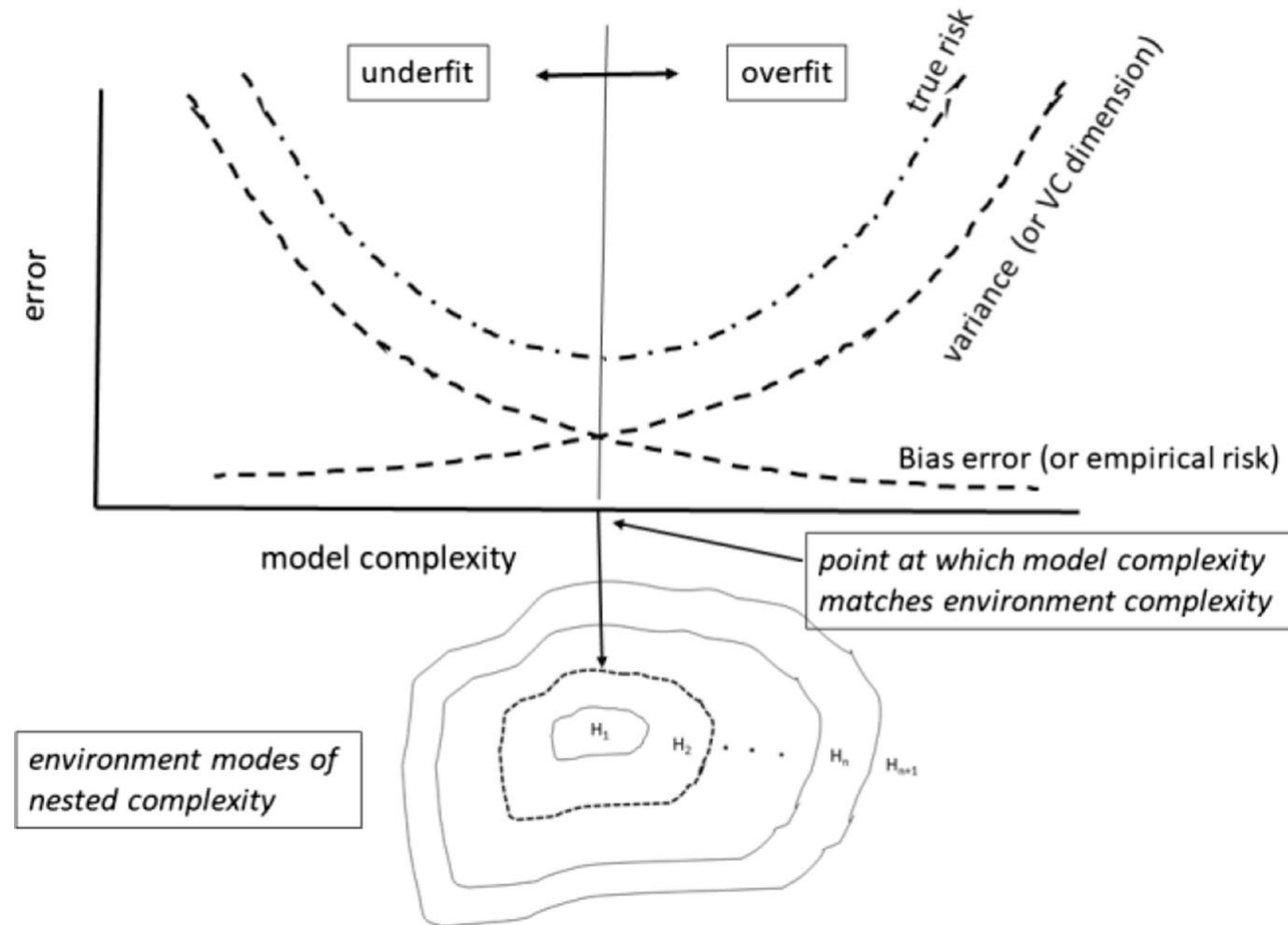


Figura 6 – Limitante superior para o risco esperado e risco empírico em função da complexidade do modelo. Fonte: Land W.H., Schaffer J.D. “The Support Vector Machine”, Chapter as part of The Art and Science of Machine Intelligence, Springer, 2020.

9 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 (classificação binária).
- Quando os dados de treinamento são separáveis, o hiperplano ótimo no espaço de características é aquele que apresenta a máxima margem de separação ρ (ver Figura 7).

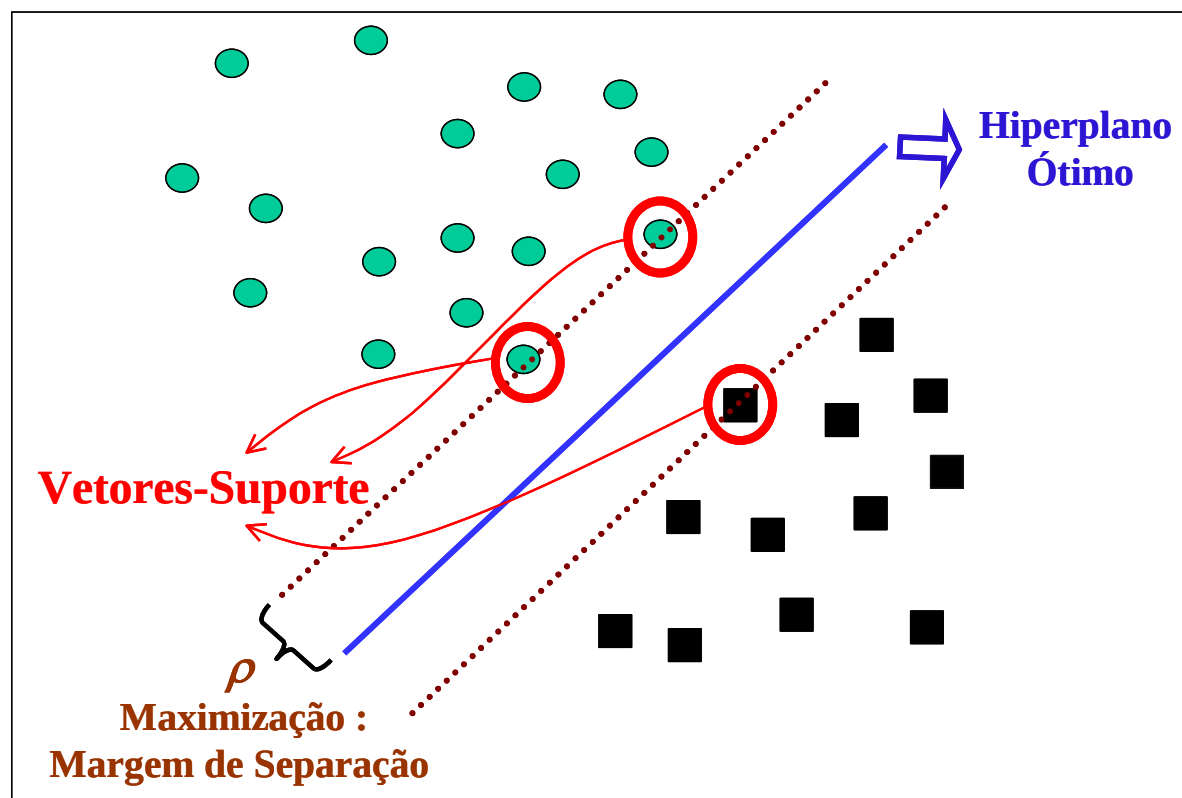


Figura 7 – Hiperplano de máxima margem

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

9.1 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:

$$(x_i, y_i)_{1 \leq i \leq N}, \quad x \in \mathbb{R}^m, \quad y \in \{+1, -1\}$$

onde 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 $g : X \subseteq \mathbb{R}^m \rightarrow \mathbb{R}$ 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:

$$g(x) = (w^T x) + b = 0 \quad (12)$$

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

- Assim, podemos aplicar a seguinte estratégia de decisão:

$$(w^T x) + b \geq 0 \quad \text{para } y = +1$$

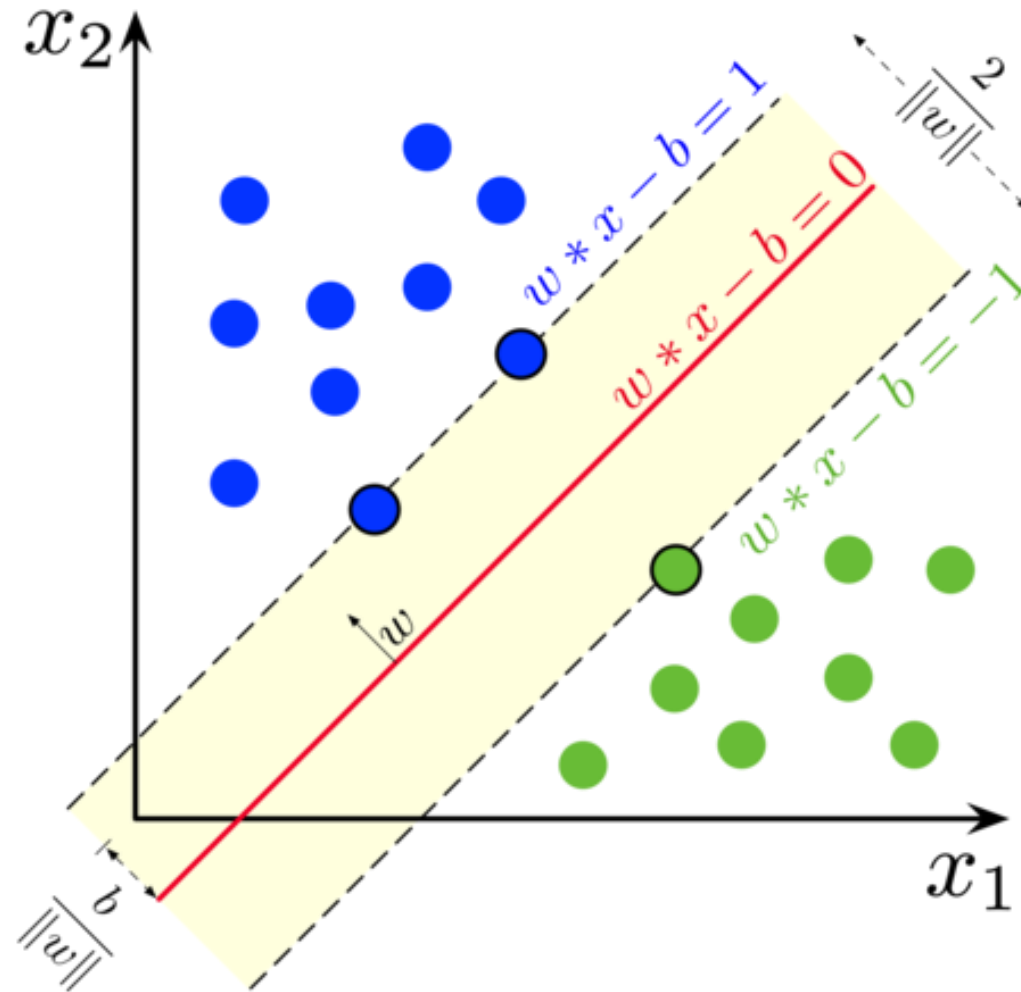
$$(w^T x) + b < 0 \quad \text{para } y = -1$$

- 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):

$$\begin{aligned} (w^T x) + b &\geq +1 && \text{para } y = +1 \\ (w^T x) + b &\leq -1 && \text{para } y_i = -1 \end{aligned} \quad (13)$$

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

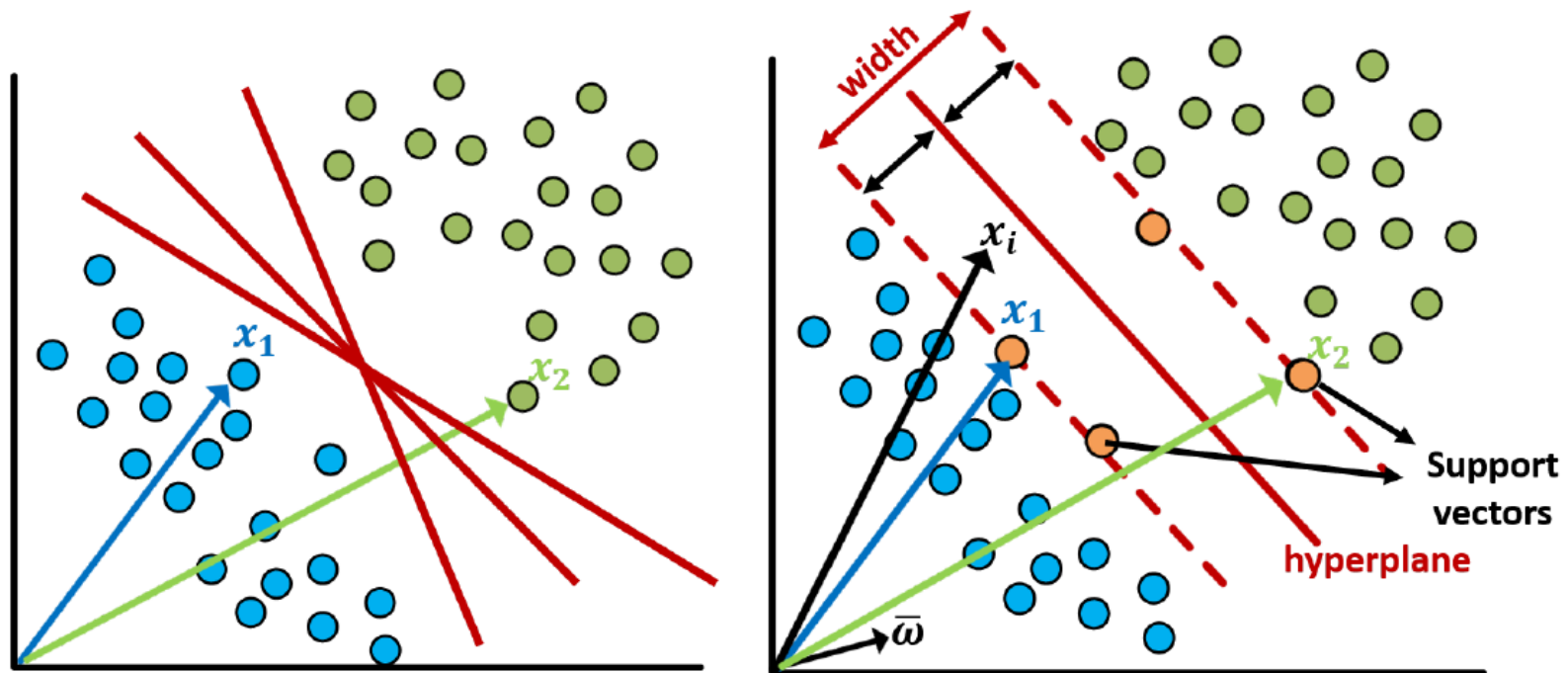
$$y [(w^T x) + b] \geq 1 \quad (14)$$



Nota: O b desta figura corresponde a $-b$ da formulação que está sendo adotada.

Fonte: https://en.wikipedia.org/wiki/Support-vector_machine

- Para um dado vetor de pesos w e intercepto b , a separação entre o hiperplano $g(x) = (w^T x) + b = 0$ e o dado de entrada mais próximo é chamada de *margem de separação* e é denotada por ρ .
- 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* ρ 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 ρ do hiperplano são chamados vetores-suporte (*support vectors*).
- O conceito de hiperplano ótimo foi desenvolvido por Vapnik e Chervonenkis em 1965 (VAPNIK & CHERVONENKIS, 1974). A figura a seguir ilustra a geometria de construção do hiperplano ótimo, dentre os infinitos hiperplanos possíveis.



(a) Separating hyper-planes.

(b) Best hyper-plane.

Fonte: Savas, C. & Dosis, F. “The Impact of Different Kernel Functions on the Performance of Scintillation Detection Based on Support Vector Machines”, Sensors, vol. 19, no. 23, paper no. 5219, 2019.

- Os dados para os quais o resultado da Eq. (14) é igual a 1 são os vetores-suporte, pois são aqueles que se encontram à distância ρ do hiperplano ótimo.
- 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 desafiadora. Como tal, eles têm uma relação direta com a localização da superfície de decisão.
- Considere a forma canônica (13) para os vetores-suporte (hiperplanos com a margem igual a 1 são conhecidos como hiperplanos canônicos):

$$(w^T x^{sv+}) + b = +1, \text{ sendo } x^{sv+} \text{ um vetor-suporte pertencente à classe } y = +1 \quad (15)$$

$$(w \cdot x^{sv-}) + b = -1, \text{ sendo } x^{sv-} \text{ um vetor-suporte pertencente à classe } y = -1$$

- Para calcular a distância algébrica dos vetores-suporte ao hiperplano ótimo, ou seja, o tamanho da margem ρ , é preciso primeiro normalizar o vetor de pesos w , e usando a equação do hiperplano canônico (15), tem-se:

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

- A Eq. (16) mostra que maximizar a margem de separação entre as classes é equivalente a minimizar a norma euclidiana do vetor de pesos w .
- Em resumo, o hiperplano ótimo, definido pela Eq. (14), apresenta o vetor de pesos w ótimo proporcionando 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 linearmente separáveis $(x_i, y_i)_{1 \leq i \leq N}$, $x \in \mathbb{R}^m$, $y \in \{+1, -1\}$, onde x são os dados de entrada e y é a correspondente resposta desejada, encontre o valor do vetor de pesos w e do intercepto b que resolve o seguinte problema:

$$\begin{aligned} \text{Minimizar} \quad & V(w, b) = \frac{1}{2} \|w\|^2 \\ \text{Sujeito a} \quad & \forall_{i=1}^N : y_i [w^T x_i + b] \geq 1 \end{aligned} \tag{17}$$

- O escalar $\frac{1}{2}$ foi incluído por conveniência da representação.
- O problema (17) pode ser resolvido utilizando o método dos multiplicadores de Lagrange, para transformar o problema de otimização primal em seu correspondente problema dual.
- Considere a função lagrangeana referente ao problema (17):

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

onde os multiplicadores de Lagrange α_i são todos não-negativos. Sendo a função-objetivo do problema (17) convexa e todas as suas restrições funções afins, o correspondente problema de otimização dual é encontrado aplicando as condições de Karush-Kuhn-Tucker junto à função lagrangeana (18), produzindo:

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

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

$$\begin{aligned}L(w, b, \alpha) &= \frac{1}{2} \sum_{i,j=1}^N y_i y_j \alpha_i \alpha_j (x_i^T x_j) - \sum_{i,j=1}^N y_i y_j \alpha_i \alpha_j (x_i^T x_j) + \sum_{i=1}^N \alpha_i \\ &= \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j=1}^N y_i y_j \alpha_i \alpha_j (x_i^T x_j)\end{aligned}\tag{20}$$

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

$$\begin{aligned} \text{Maximizar} \quad & W(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j=1}^N y_i y_j \alpha_i \alpha_j (x_i^T x_j) \\ \text{Sujeito a} \quad & \sum_{i=1}^N y_i \alpha_i = 0 \\ & \forall_{i=1}^N : \alpha_i \geq 0 \end{aligned} \tag{21}$$

- 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 $(x_i^T x_j)_{1 \leq i \leq N ; 1 \leq j \leq N}$.
- 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 outros problemas de otimização.

- Determinando os multiplicadores de Lagrange ótimos α^* , pode-se calcular o vetor de pesos ótimo w^* utilizando a primeira equação em (19), produzindo:

$$w^* = \sum_{i=1}^N y_i \alpha_i^* x_i \text{ onde } y \in \{+1, -1\}, \alpha_i^* \in \mathbb{R} \text{ e } x \in \mathbb{R}^m \quad (22)$$

- Assim, w^* é o vetor que encontra o hiperplano ótimo com a máxima margem de separação ρ (veja Eq. (16)).
- O valor do intercepto ótimo b^* é encontrado utilizando a Eq. (22), com o auxílio das restrições primais em (17), levando a:

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

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

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

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

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

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

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

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

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

- 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 α^* 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:

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

- 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 $f(x)$ for negativo, o ponto x pertence à classe negativa; se o resultado de $f(x)$ for positivo, o ponto x pertence à classe positiva.

9.2 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 $K(x_i, x_j)$, ou seja, $K(x_i, x_j) = K(x_j, x_i)$, com $x_i, x_j \in \mathfrak{R}^m$, pode ser expressa como um produto interno

$$K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$$

para alguma função $\phi(\cdot): \mathfrak{R}^m \rightarrow \mathfrak{R}^q$, se e somente se $K(x_i, x_j)$ é semidefinida positiva, ou seja:

$$\int K(x_i, x_j) g(x_i) g(x_j) dx_i dx_j \geq 0 \quad \forall g(\cdot)$$

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

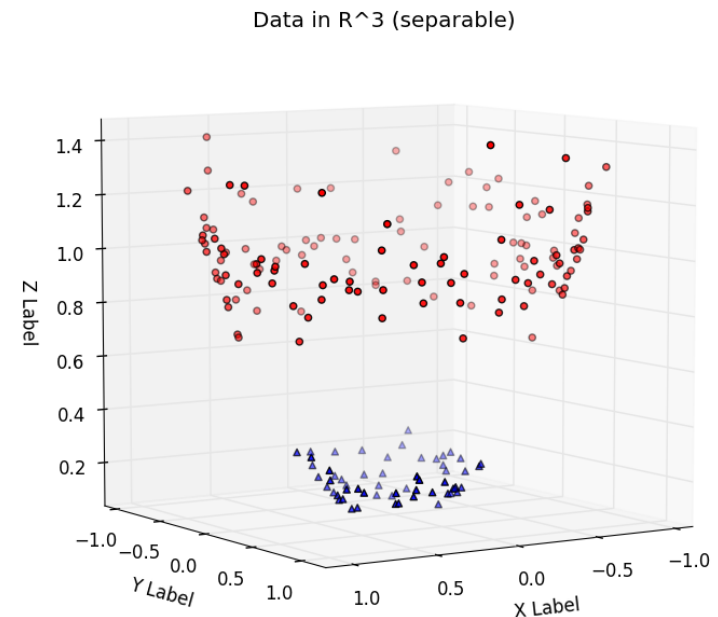
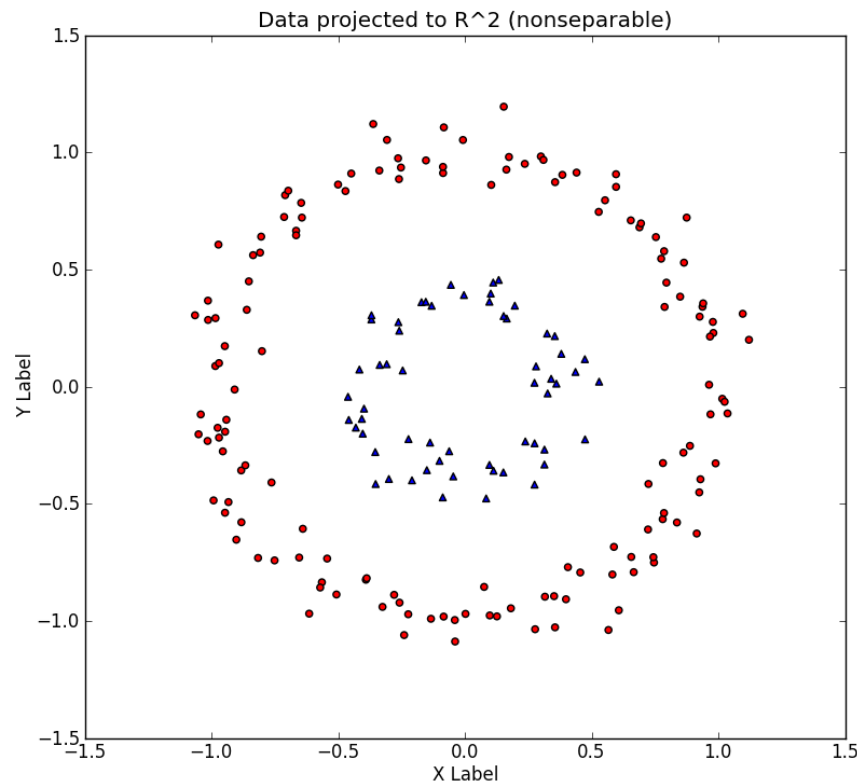
$$\begin{bmatrix} K(x_1, x_1) & K(x_1, x_2) & \cdots & K(x_1, x_N) \\ K(x_2, x_1) & \ddots & & K(x_2, x_N) \\ \vdots & & \ddots & \vdots \\ K(x_N, x_1) & K(x_N, x_2) & & K(x_N, x_N) \end{bmatrix}$$

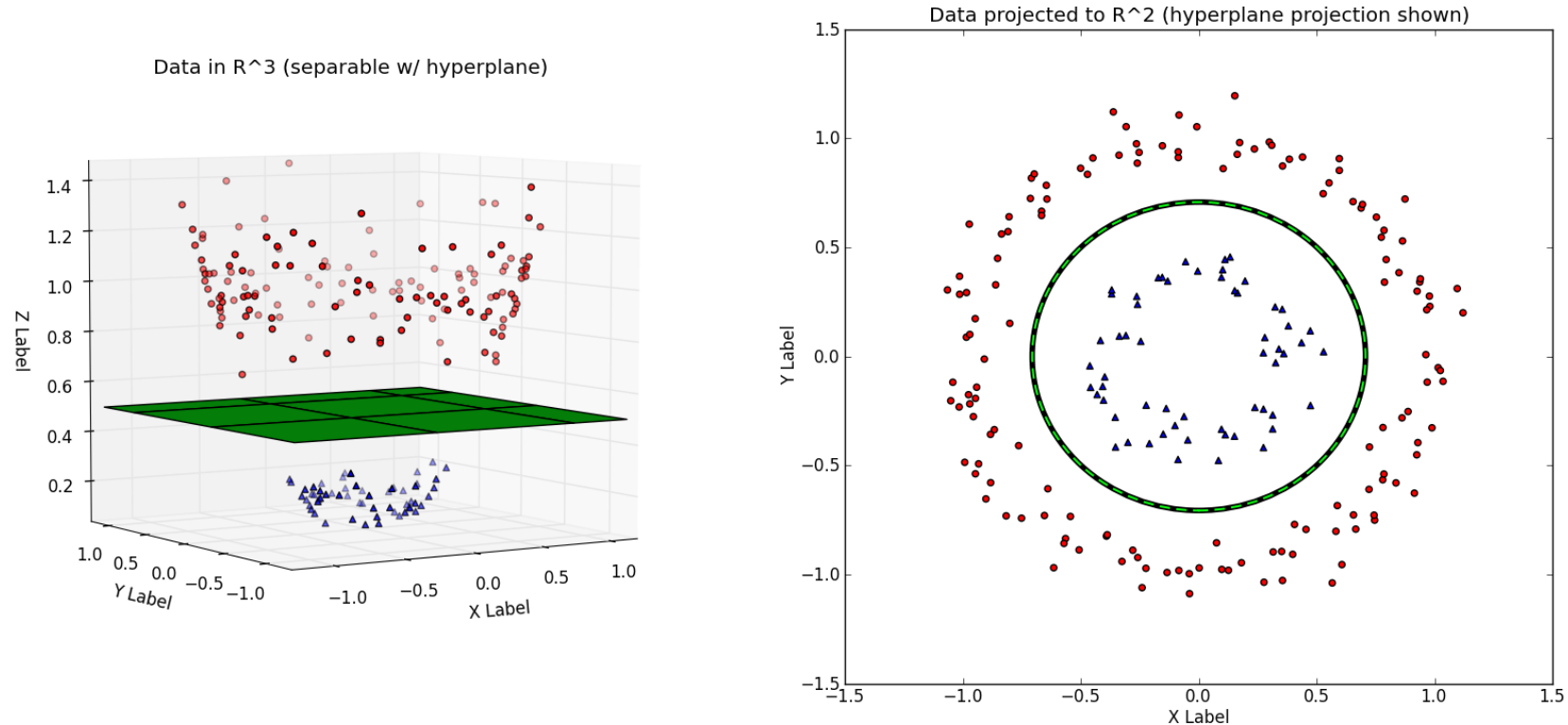
é semidefinida positiva para qualquer coleção de pontos $\{x_1, \dots, x_N\}$ no $\mathcal{R}^m$.

- Nos casos em que se tem q bem maior que m , em lugar de realizar o produto interno $\phi(x_i)^T \phi(x_j)$, é possível aplicar diretamente $K(x_i, x_j)$, 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 $\phi(\cdot): \mathcal{R}^m \rightarrow \mathcal{R}^q$.

9.3 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.





- 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.

9.4 Extensão para o espaço de características (*feature space*)

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

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

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

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

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

- Formalmente, se $\phi(\cdot): \mathcal{R}^m \rightarrow \mathcal{R}^q$ for a função que mapeia do espaço original (de dimensão m) para o espaço de características (de dimensão q), então vale a seguinte relação: $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$.
- Basicamente, $K(x_i, x_j): \mathcal{R}^m \times \mathcal{R}^m \rightarrow \mathcal{R}$ calcula a similaridade entre os vetores $x_i, x_j \in \mathcal{R}^m$ quando esses são mapeados por $\phi(\cdot): \mathcal{R}^m \rightarrow \mathcal{R}^q$ 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 isto, restará apenas ajustar seus parâmetros.

9.5 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 (ξ_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:

$$\begin{aligned} \text{Minimizar} \quad & V(w, b, \xi) = \frac{1}{2} \|w\|^2 + C_+ \sum_{i: y_i = +1} \xi_i + C_- \sum_{j: y_j = -1} \xi_j \\ \text{Sujeito a} \quad & \forall_{i=1}^N : \quad y_i [w^T x_i + b] \geq 1 - \xi_i \\ & \forall_{i=1}^N : \quad \xi_i \geq 0 \end{aligned}$$

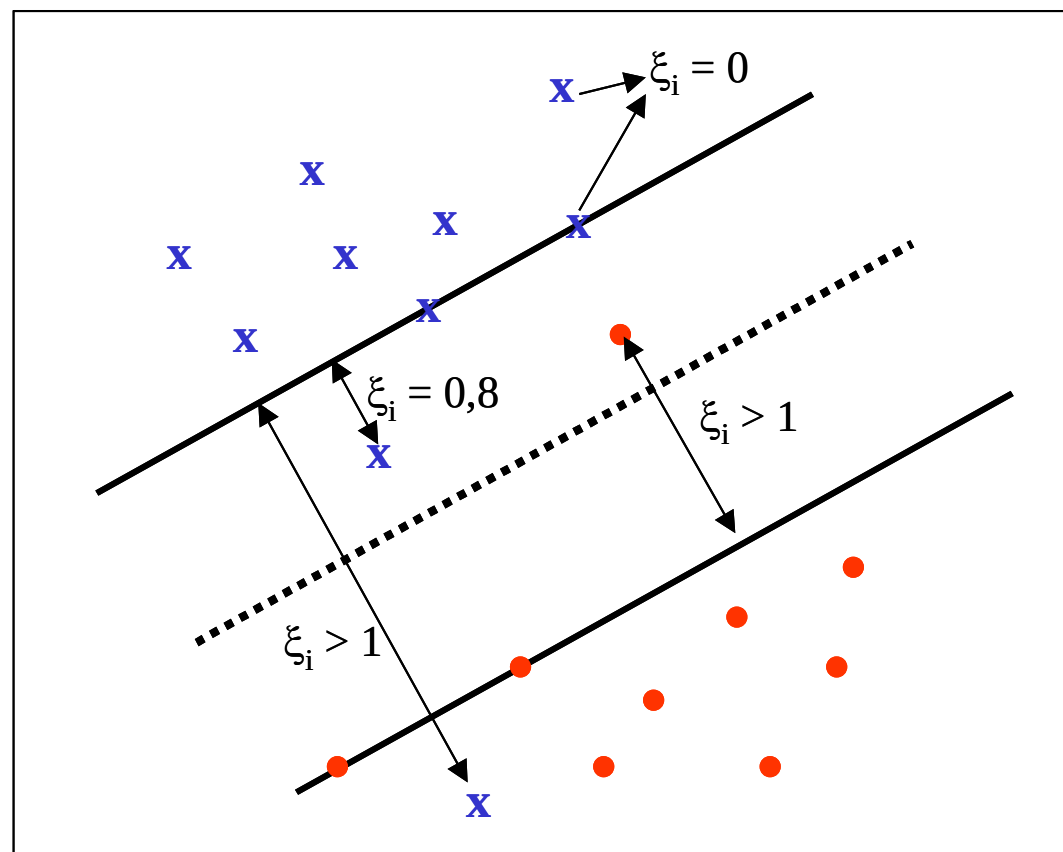
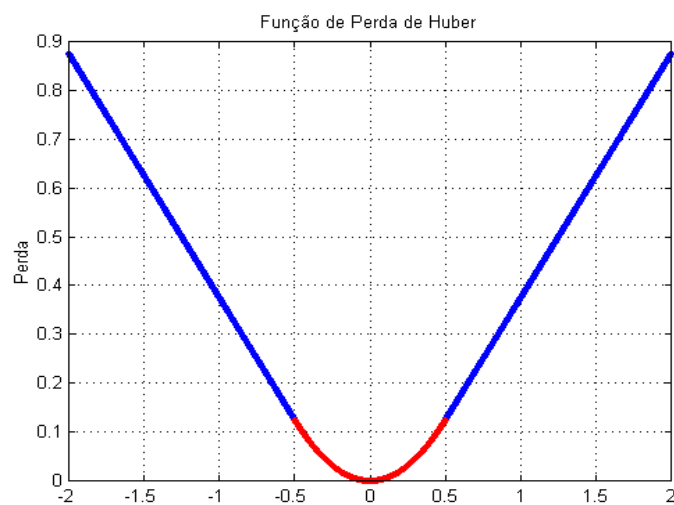
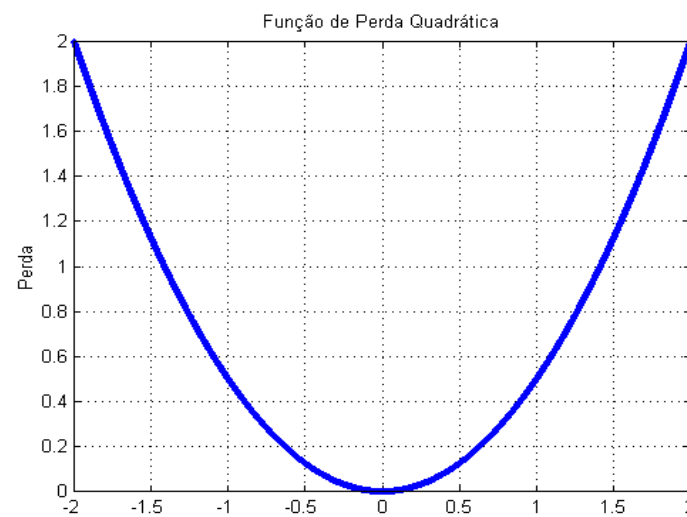
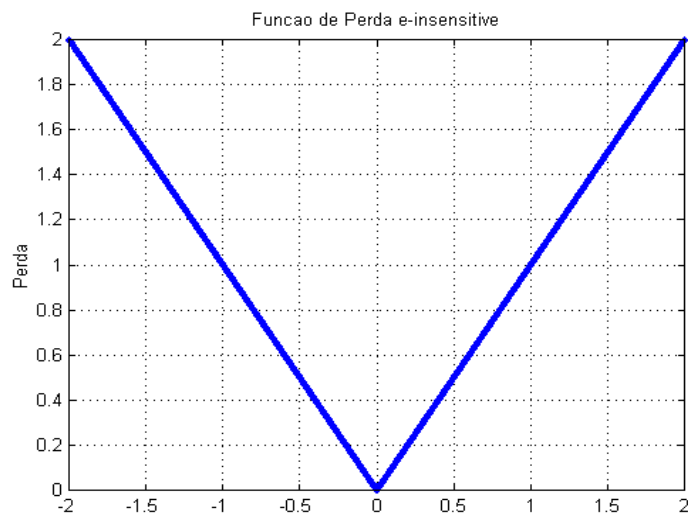


Figura 8 – Papel exercido pelas variáveis de folga

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



- A formulação dual é dada como segue:

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

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

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

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

- É preciso definir os custos C_+ e C_- , que influenciam no desempenho de generalização do classificador. Para tanto, pode-se empregar validação cruzada, por exemplo.

9.6 Alguns tipos de funções kernel

- Linear:

$$K(x, y) = x^T y$$

- Polinomial:

$$K(x, y) = \left(b(x^T y) + c \right)^d$$

- Função de base radial gaussiana:

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

- Função de base radial laplaciana:

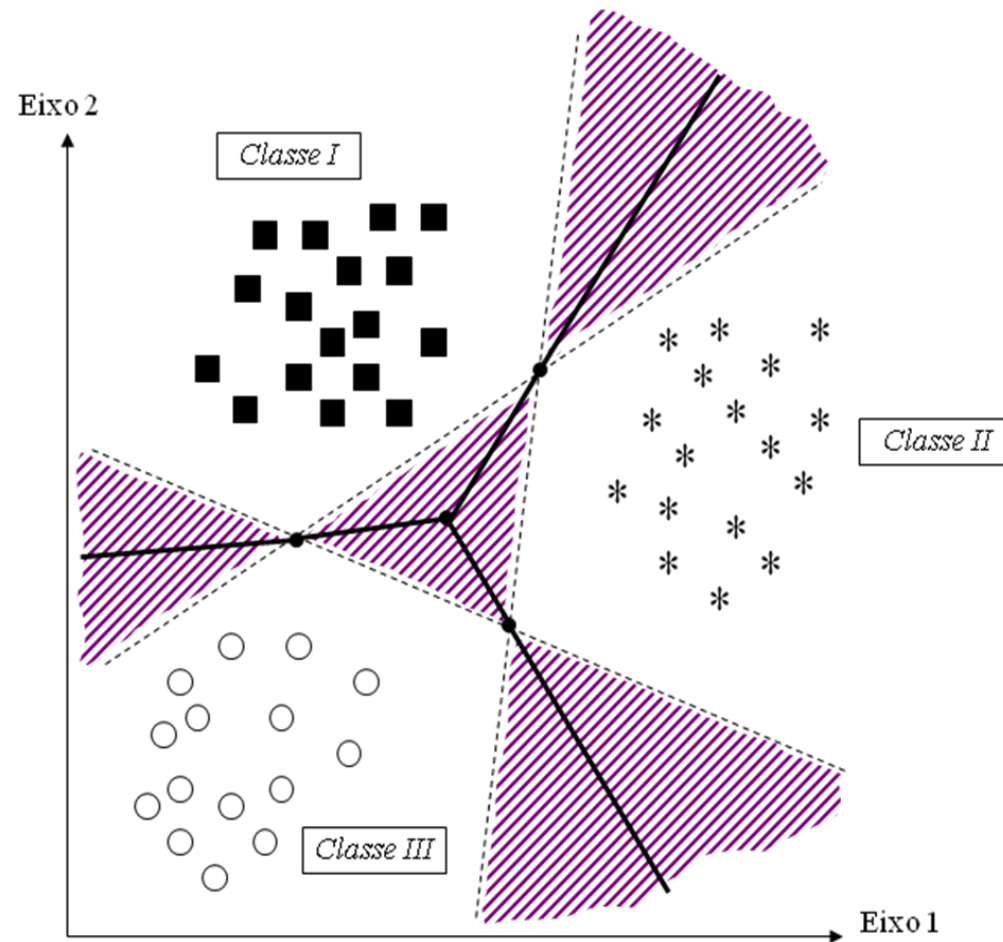
$$K(x, y) = \exp\left(-\frac{|x - y|}{2\sigma^2} \right)$$

- Tangente hiperbólica:

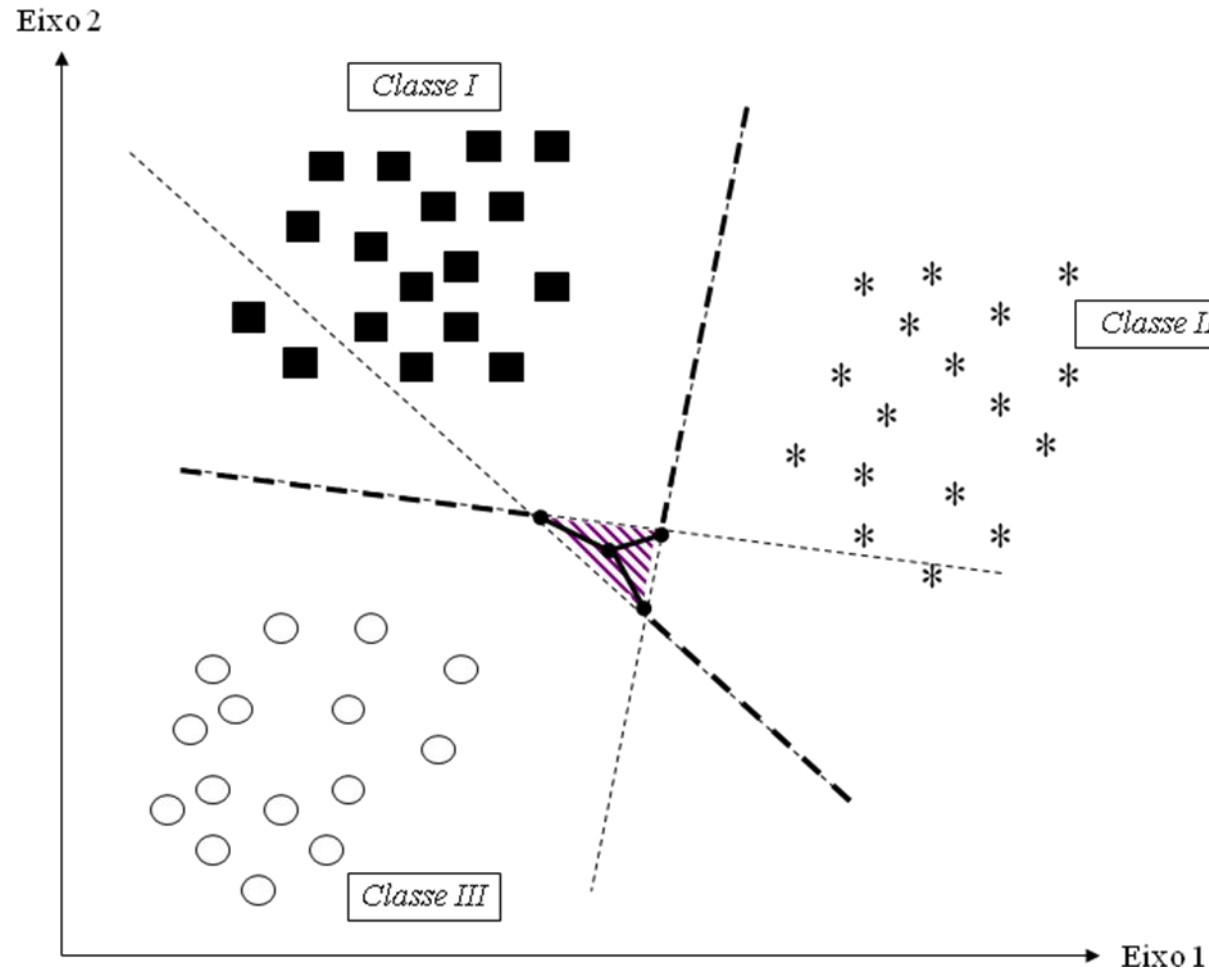
$$K(x, y) = \tanh\left(b(x^T y) + c \right)$$

9.7 Extensões para o tratamento de múltiplas classes

9.7.1 Um contra todos



9.7.2 Um contra um



9.7.3 Grafo direcionado acíclico (DAGSVM)

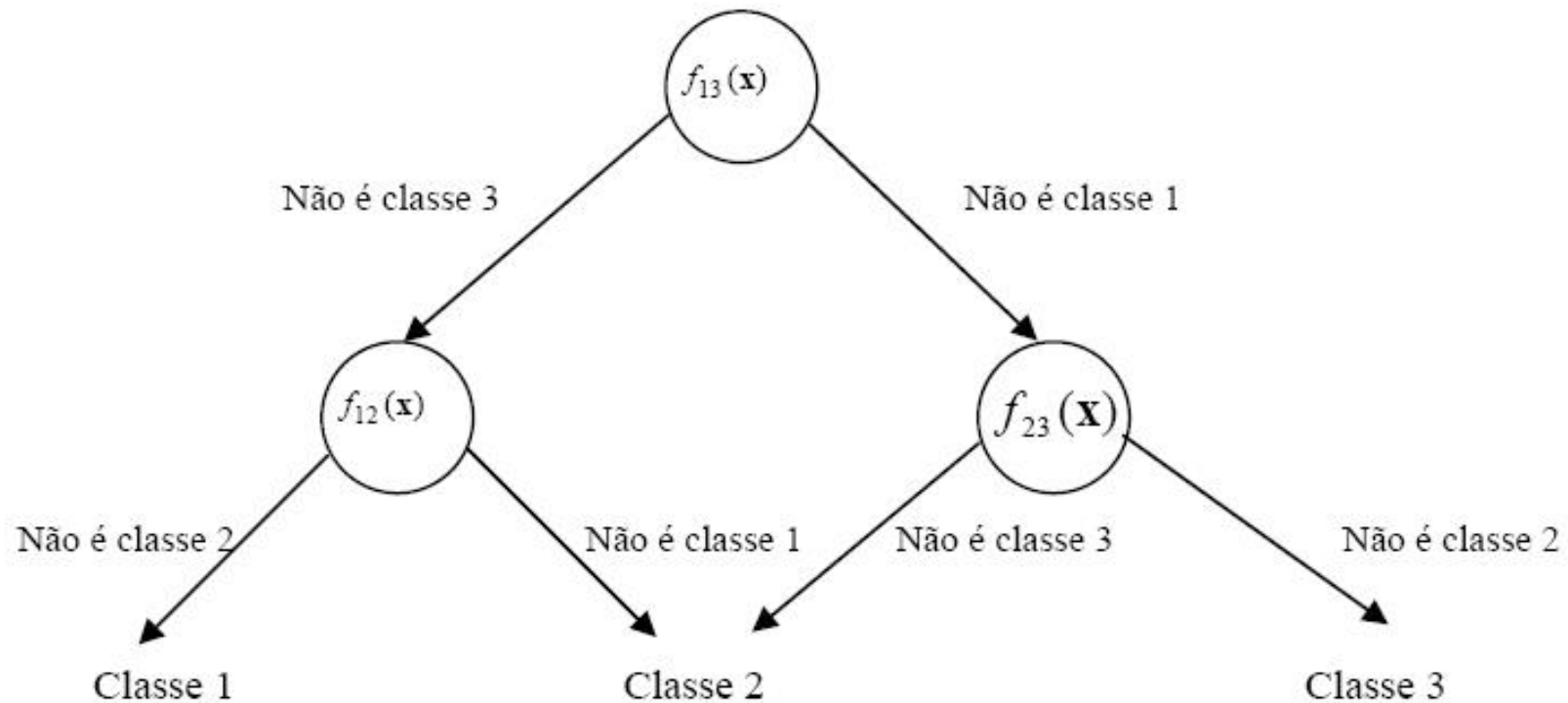
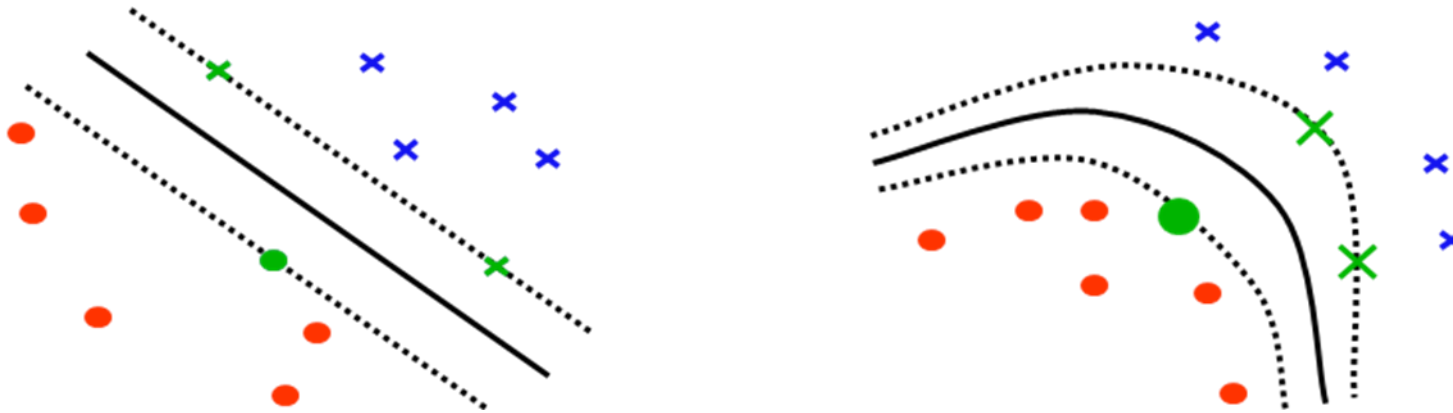


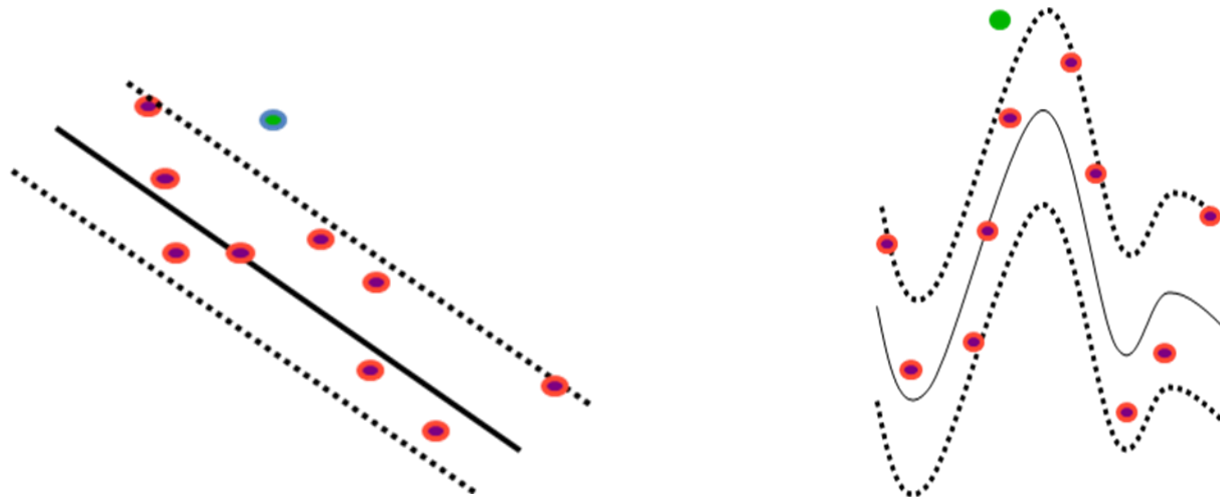
Figura 9 – Ilustração de um grafo direcionado acíclico para um problema com 3 classes no método DAGSVM.

9.8 Extensão para o tratamento de problemas de regressão

- Enquanto para classificação de dados tem-se:



- Para regressão de dados são necessárias adaptações, produzindo:

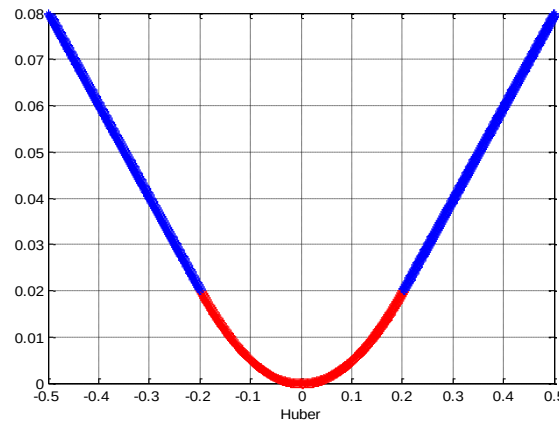
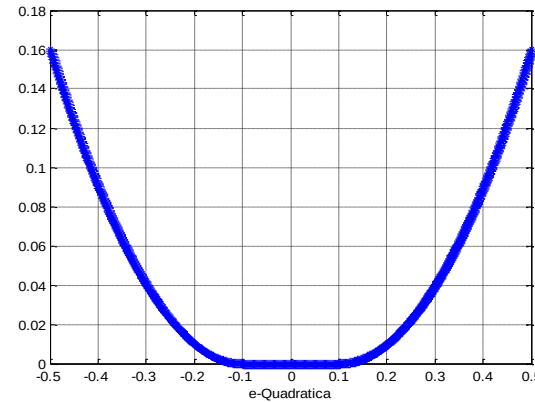
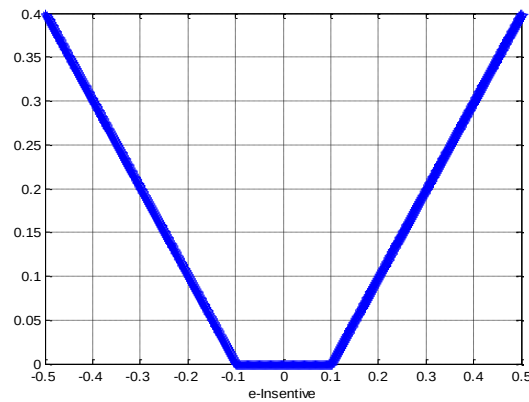


- Cria-se, assim, um compromisso entre a minimização da margem e a minimização do somatório dos erros admissíveis, conforme indicado na formulação a seguir, referente ao problema primal:

$$\begin{aligned} \text{Minimizar} \quad & V(w, b, \xi^+, \xi^-) = \frac{1}{2} \|w\|^2 + C \left(\sum_{i: y_i - [w^T x_i + b] \geq 0} \xi_i^+ + \sum_{i: y_i - [w^T x_i + b] \leq 0} \xi_i^- \right) \\ \text{Sujeito a} \quad & \forall_{i=1}^N : \quad y_i - [w^T x_i + b] \leq \varepsilon + \xi_i^+ \\ & \forall_{i=1}^N : \quad [w^T x_i + b] - y_i \leq \varepsilon + \xi_i^- \\ & \forall_{i=1}^N : \quad \xi_i^+ \geq 0 \\ & \forall_{i=1}^N : \quad \xi_i^- \geq 0 \end{aligned}$$

- 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):



9.9 Exemplos de comportamento para diferentes ε 's

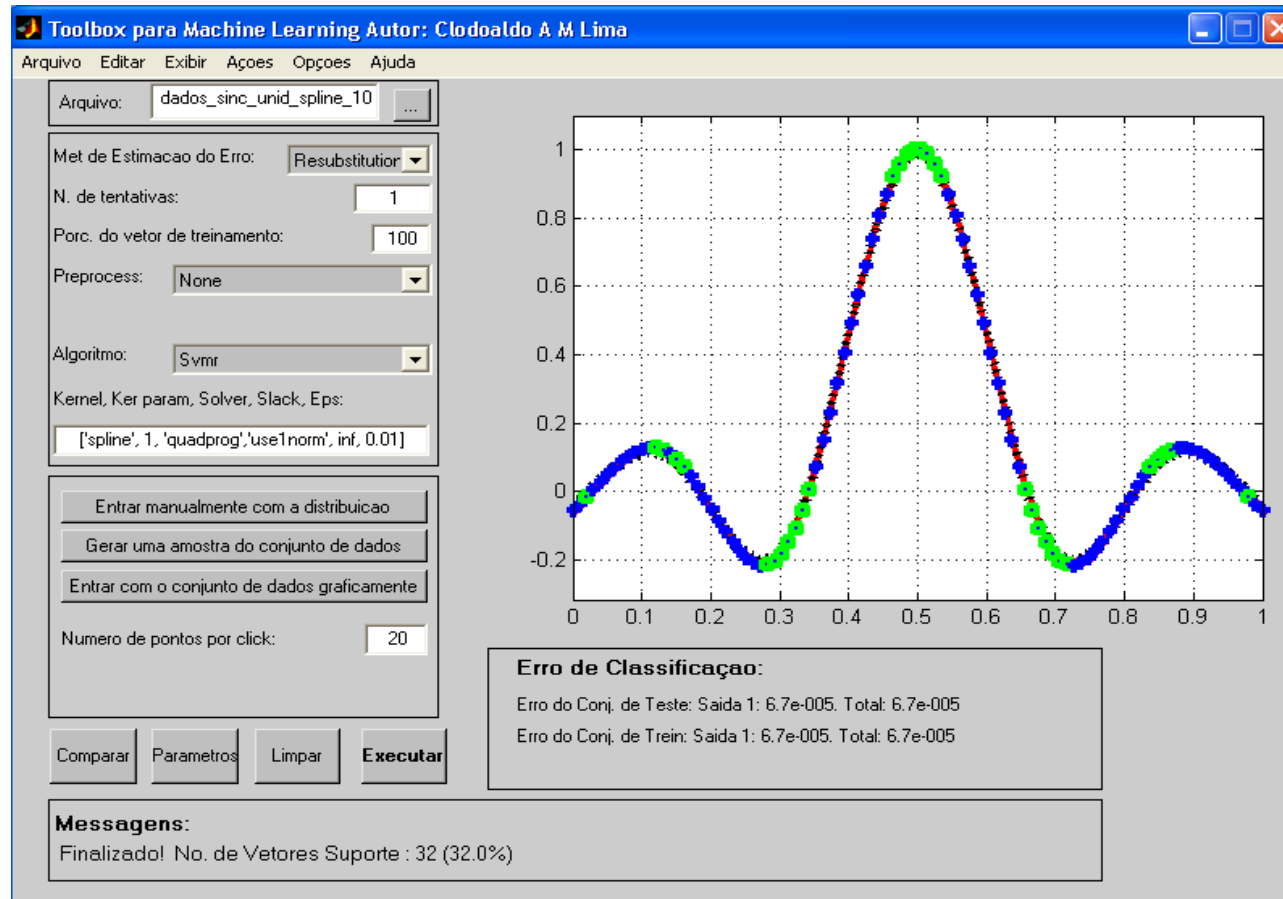


Figura 10 – $\varepsilon = 0,01$; No. de amostras = 100; No. de vetores-suporte = 32

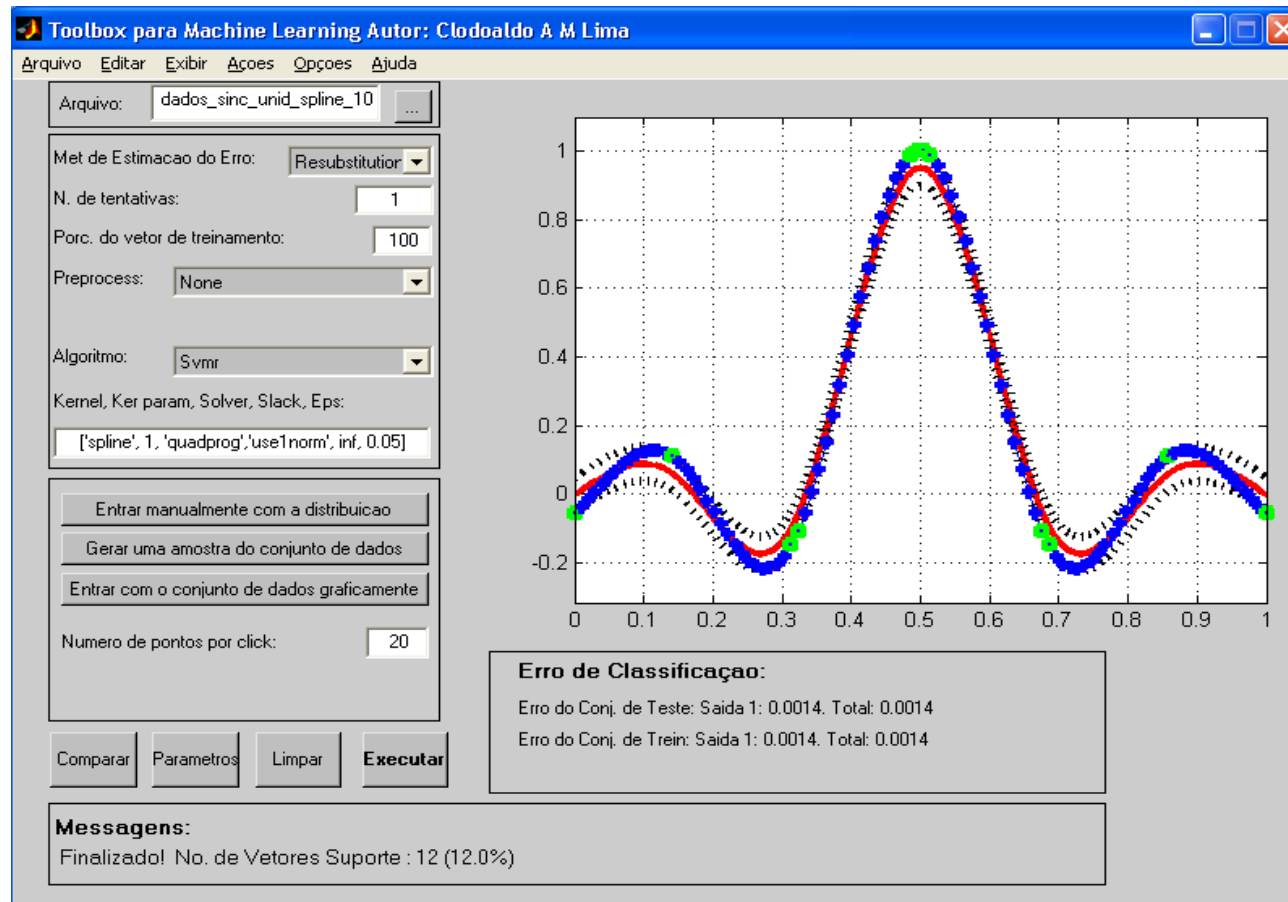


Figura 11 – $\epsilon = 0,05$; No. de amostras = 100; No. de vetores-suporte = 12

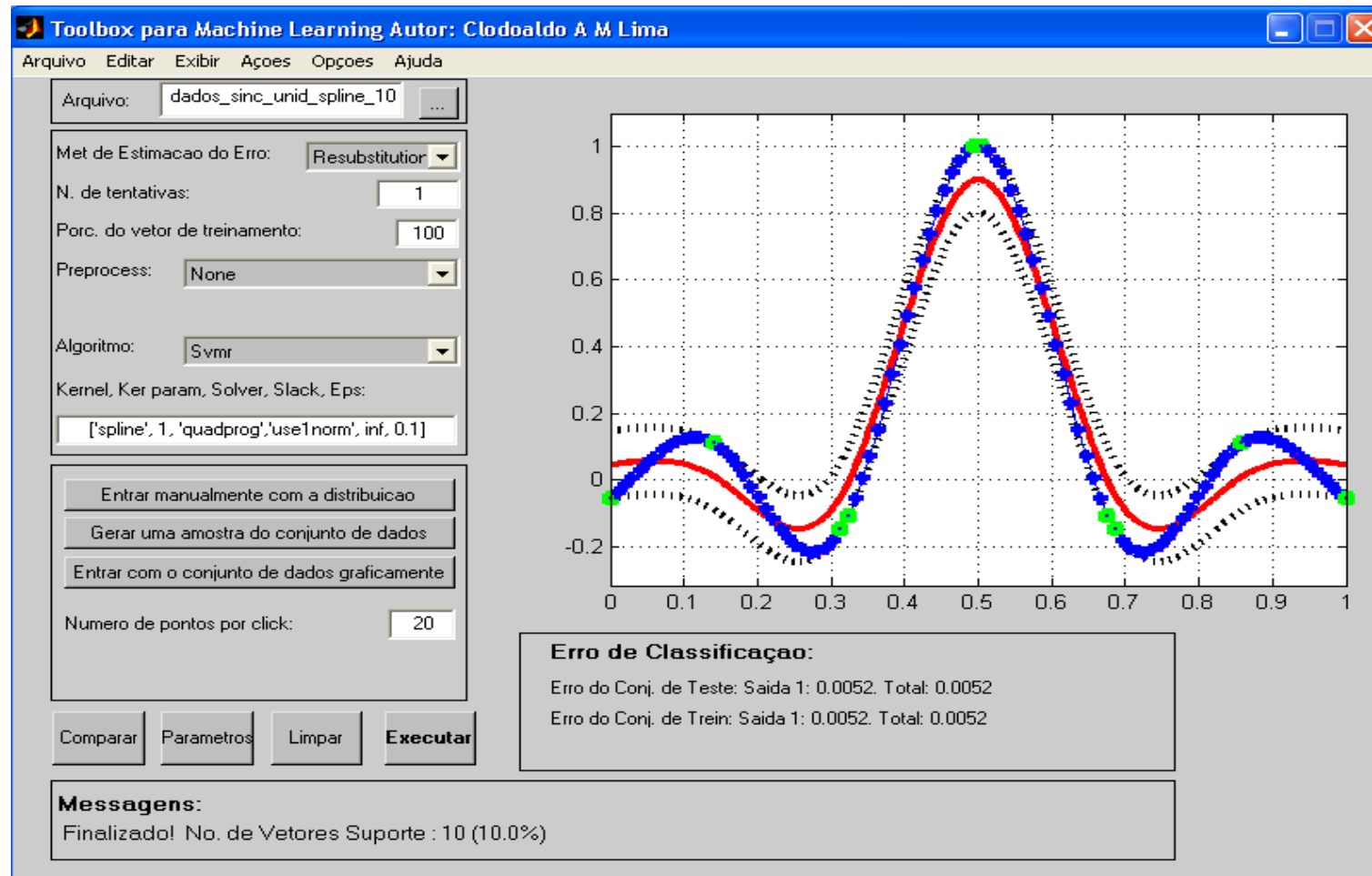


Figura 12 – $\varepsilon = 0,1$; No. de amostras = 100; No. de vetores-suporte = 10

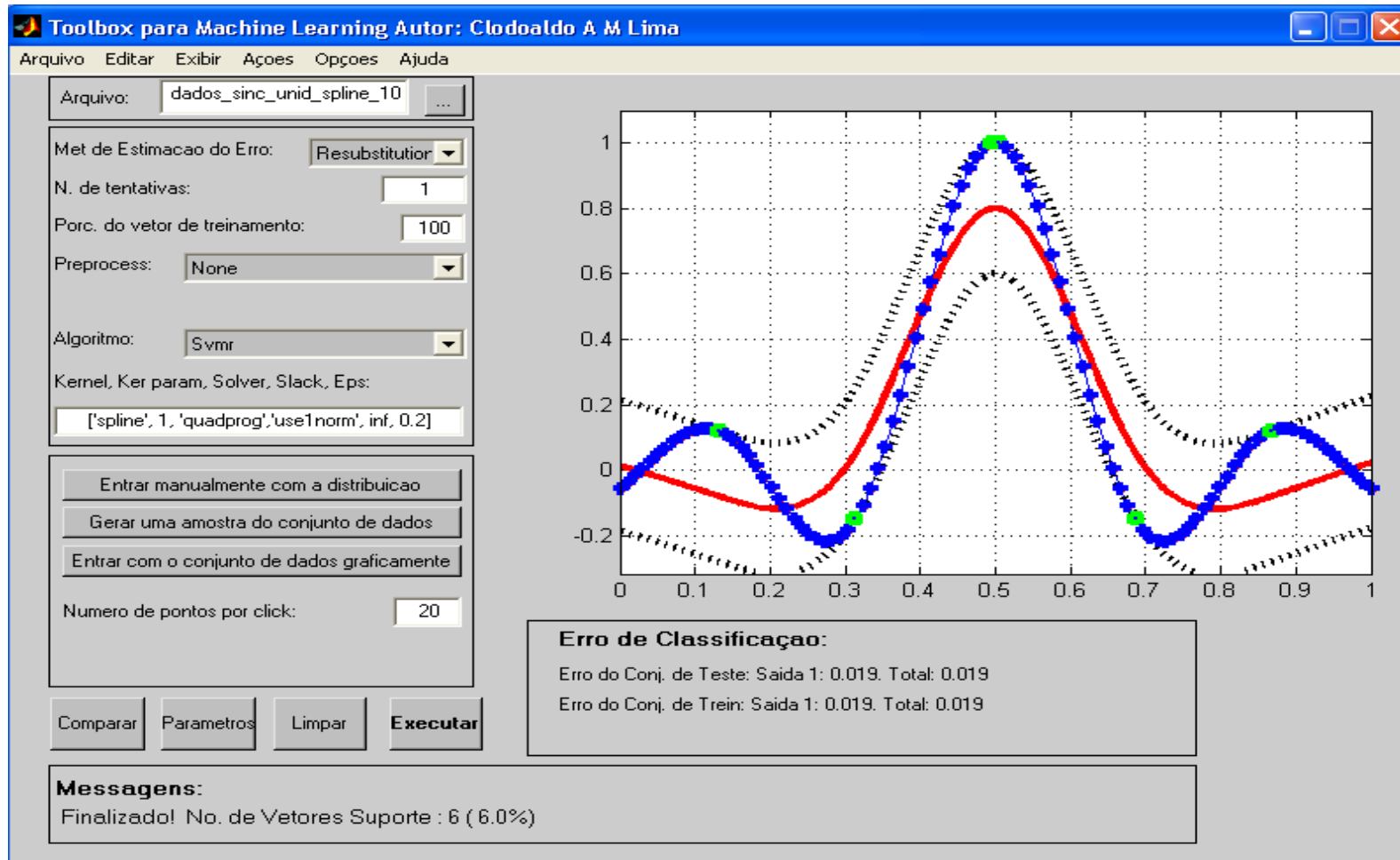


Figura 13 – $\varepsilon = 0,2$; No. de amostras = 100; No. de vetores-suporte = 6

9.10 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.

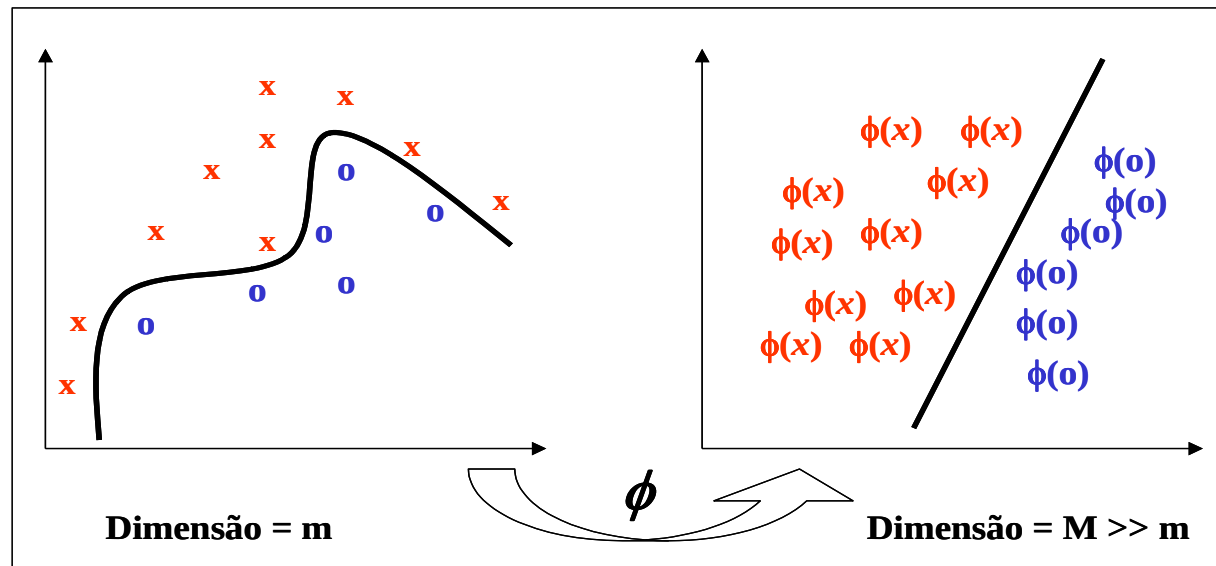


Figura 14 – Mapeamento para o espaço de características

- 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.

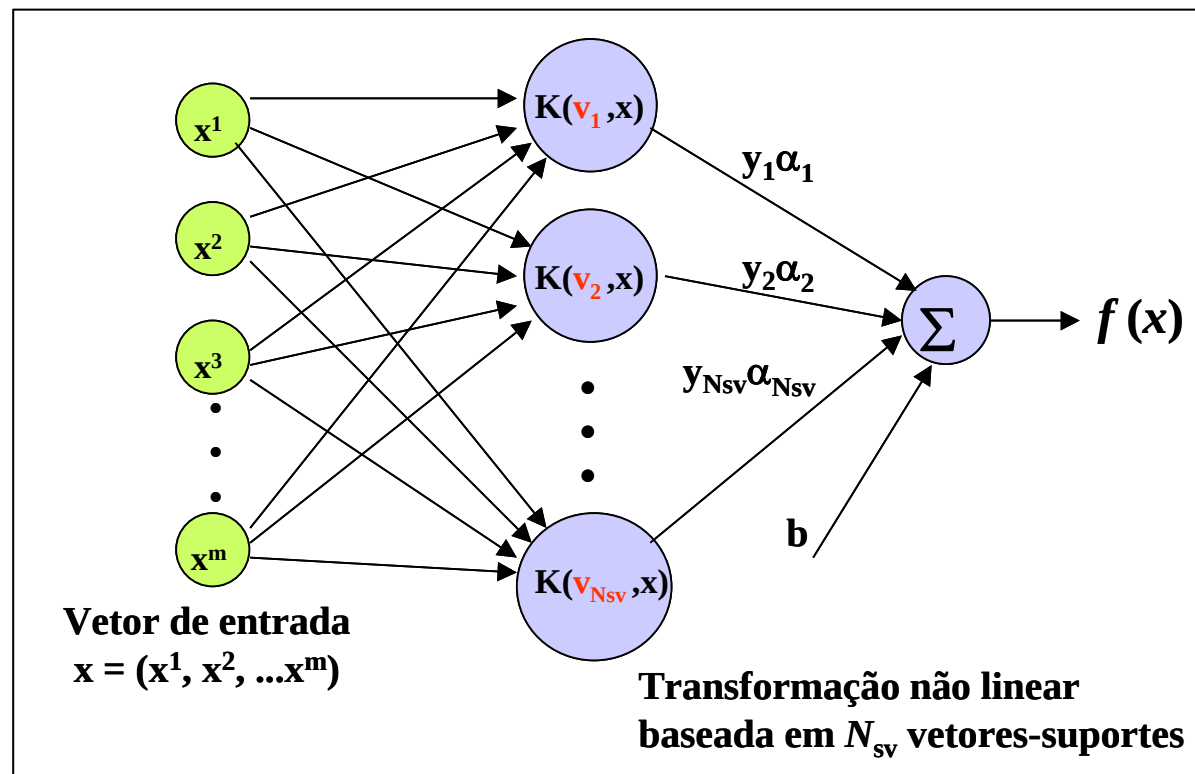


Figura 15 – Resultado final de uma SVM

$$\begin{array}{|c|} \hline \text{"Erro de Validação"} \\ \text{ou} \\ \text{"Incapacidade de} \\ \text{Generalização"} \\ \hline \end{array} \leq \begin{array}{|c|} \hline \text{"Erro de} \\ \text{Treinamento"} \\ \hline \end{array} + \begin{array}{|c|} \hline \text{Termo dependente} \\ \text{da Dimensão VC} \\ \hline \end{array}$$

Figura 16 – Princípio de minimização do risco estrutural

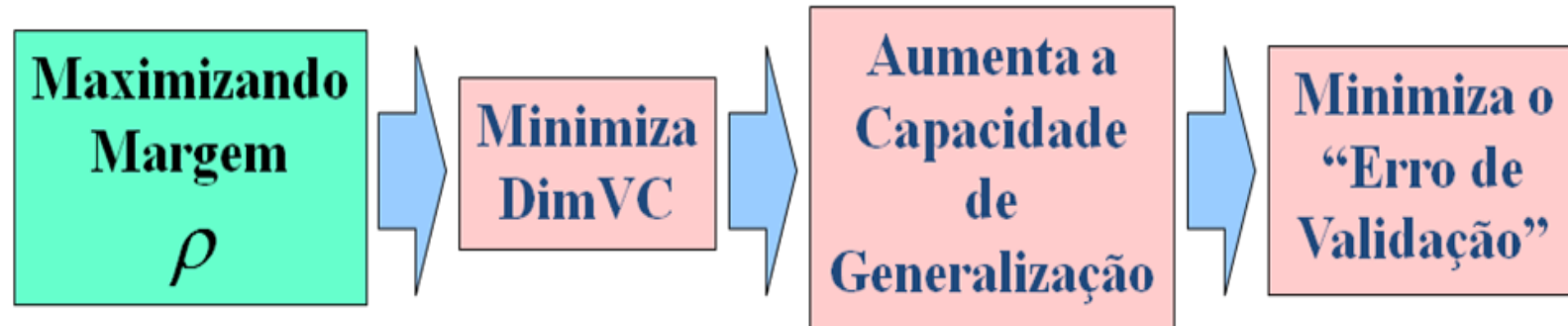


Figura 17 – Princípio dos classificadores de máxima margem

$$\text{Erro de Validação} \leq \frac{\text{Qtd. Vetores-Suportes}}{\text{Qtd. Amostras Treinamento}}$$

Figura 18 – Aspectos fundamentais da teoria do aprendizado estatístico

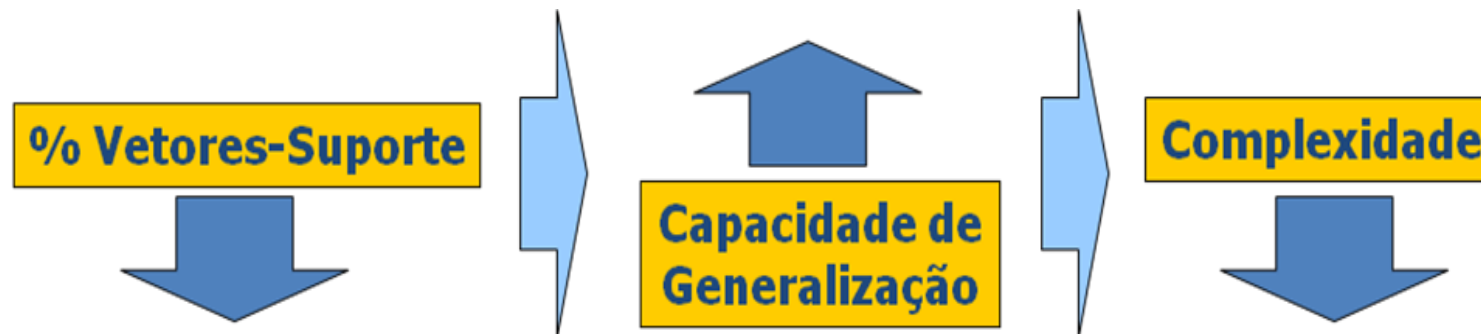
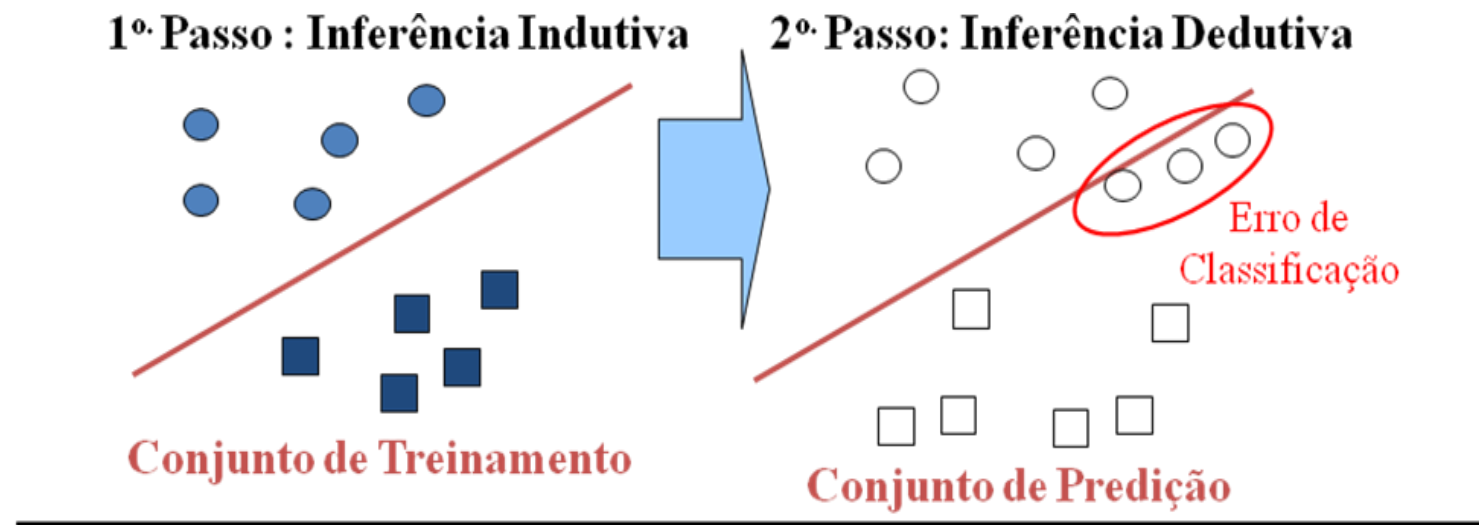


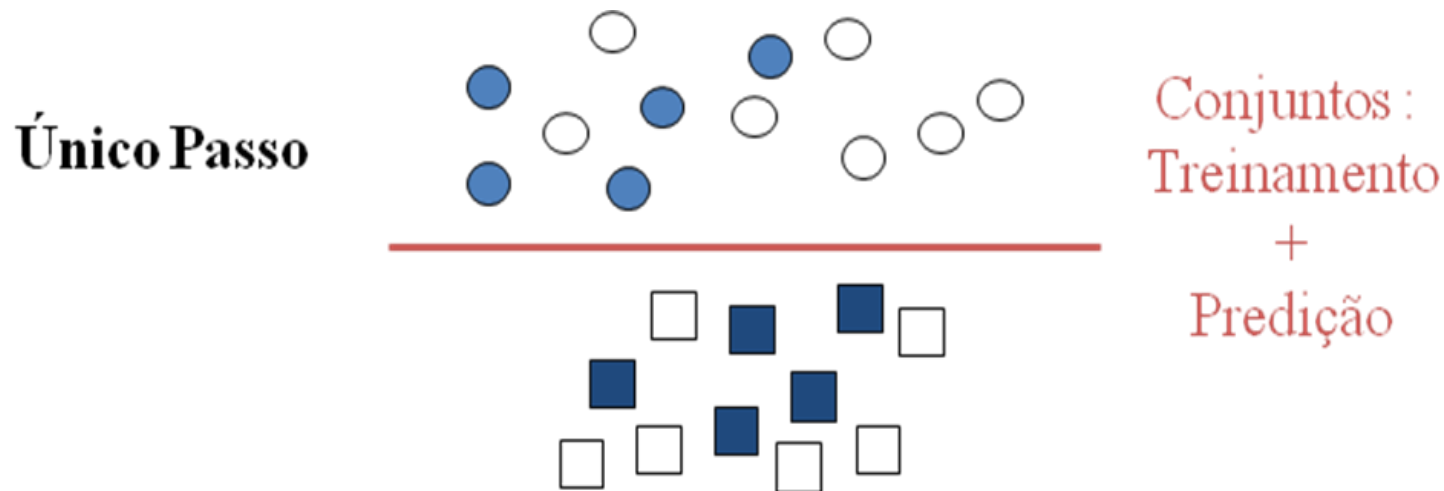
Figura 19 – Aspectos fundamentais da teoria do aprendizado estatístico quando aplicados a máquinas de vetores-suporte

9.11 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 $(x_i, y_i)_{1 \leq i \leq N}$, $x \in \mathcal{R}^m$, $y \in \{+1, -1\}$, e dos dados de predição $(x_j^*)_{1 \leq j \leq K}$, encontre o valor do vetor de pesos w , intercepto b , variáveis de folga $(\xi_i)_{1 \leq i \leq N}$, $(\xi_j)_{1 \leq j \leq K}$ e os valores da classificação do conjunto de predição $(y_j^*)_{1 \leq j \leq K}$, $y^* \in \{+1, -1\}$ que resolvem o seguinte problema:

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

$$\begin{aligned} \text{Sujeito a } \quad & \forall_{i=1}^N : \quad y_i [w^T x_i + b] \geq 1 - \xi_i \\ & \forall_{j=1}^K : \quad y_j^* [w^T x_j^* + b] \geq 1 - \xi_j^* \\ & \forall_{i=1}^N : \quad \xi_i \geq 0 \\ & \forall_{j=1}^K : \quad \xi_j^* \geq 0 \end{aligned}$$

- 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.

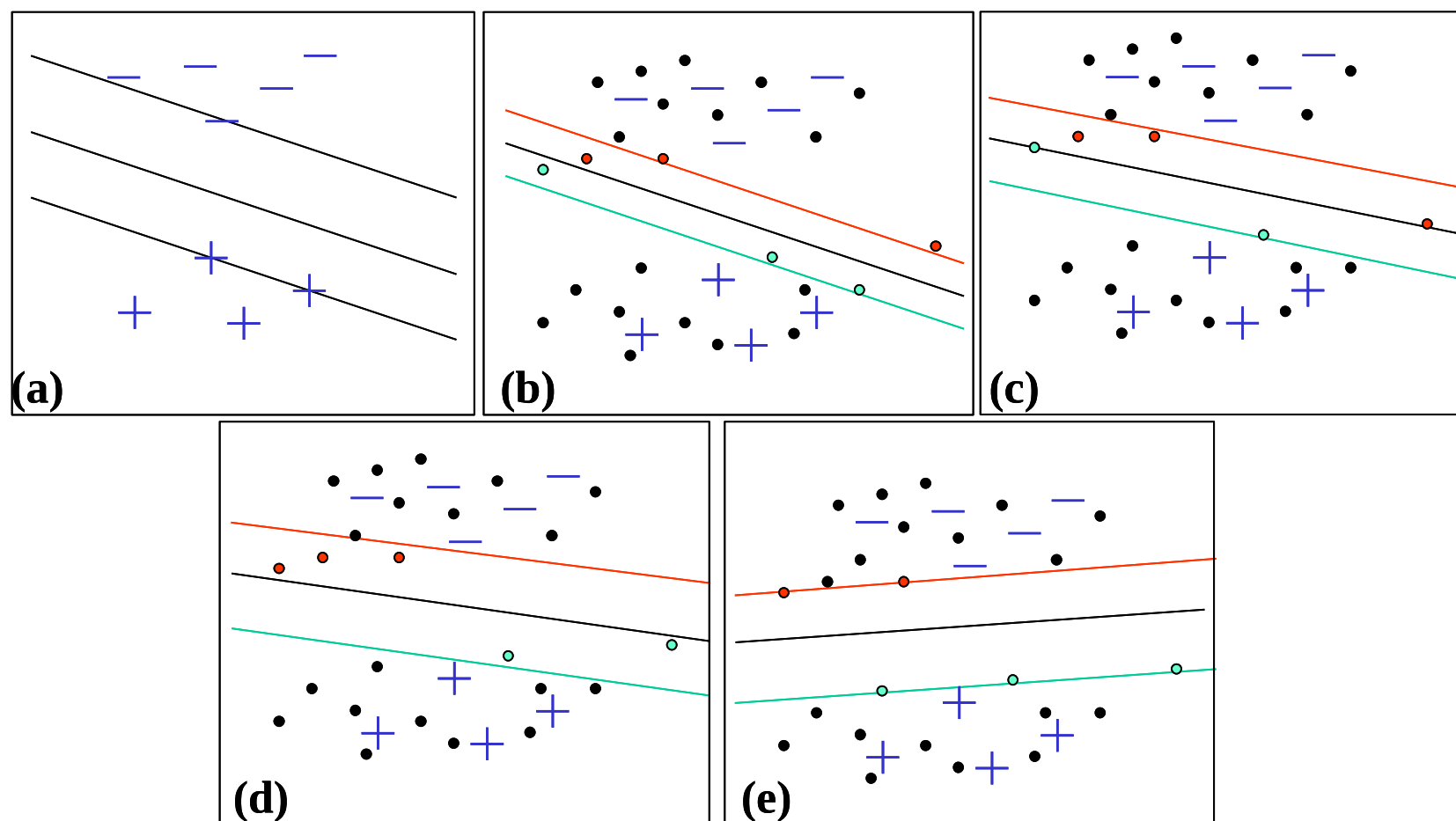
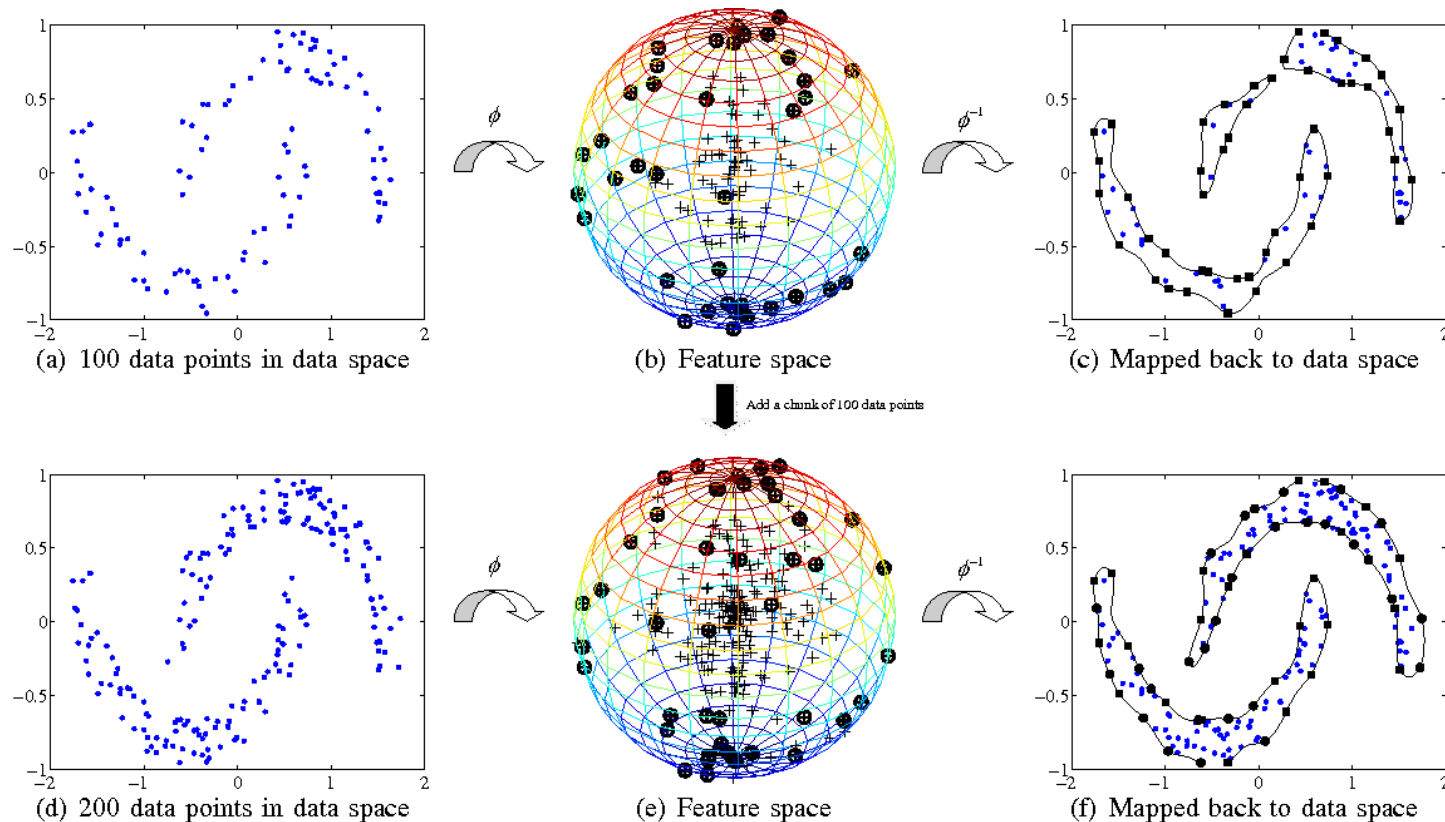


Figura 20 – Análise gráfica do funcionamento do algoritmo de JOACHIMS (1999)

9.12 Support Vector Clustering (SVC)



WANG, C.-D., LAI, J.-H., HUANG, D. “Incremental Support Vector Clustering”, IEEE 11th International Conference on Data Mining Workshops, pp. 839-846, 2011.

10 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.
- GOODFELLOW, I.; BENGIO, Y. & COURVILLE, A. “Deep Learning”, The MIT Press, 2016.
- HALEVY, A.; NORVIG, P. & PEREIRA, F. “The Unreasonable Effectiveness of Data”, IEEE Intelligent Systems, vol. 24, no. 2, pp. 8-12, 2009.
- HASTIE, T.; TIBSHIRANI, R. & FRIEDMAN, J.H. “The Elements of Statistical Learning: Data Mining, Inference, and Prediction”, 2nd edition, Springer, 2009.
- 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.