Introdução a Ciências de Dados

Aula 8: Classificação: SVM, Avaliação de modelos

Francisco A. Rodrigues
ICMC/USP
francisco@icmc.usp.br

Aula 8: Classificação

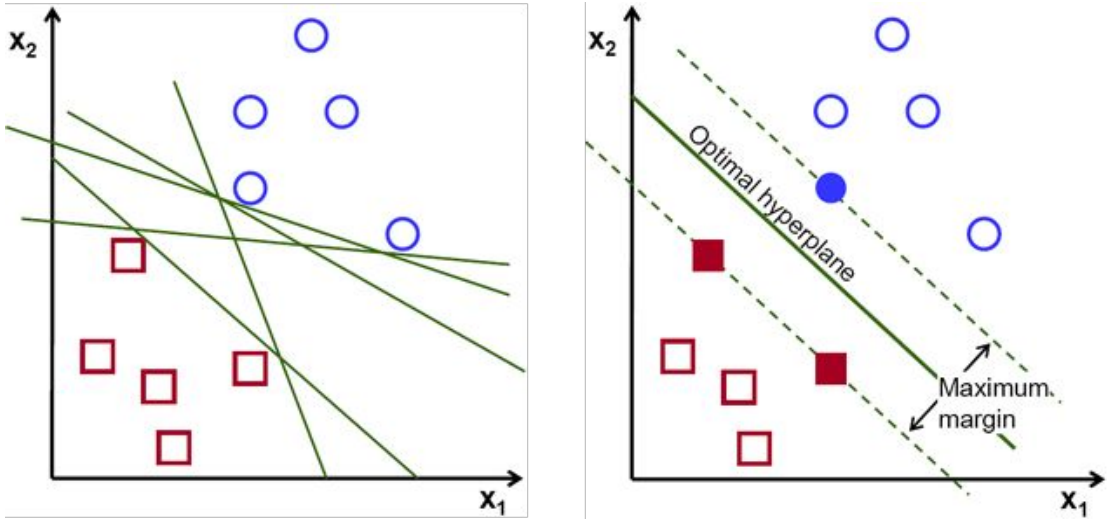
- Support Vector Machines
- Avaliando Modelos de Classificação

Support Vector Machines

Support Vector Machines

Ideia básica:

- Dada duas classes, como separá-las linearmente?

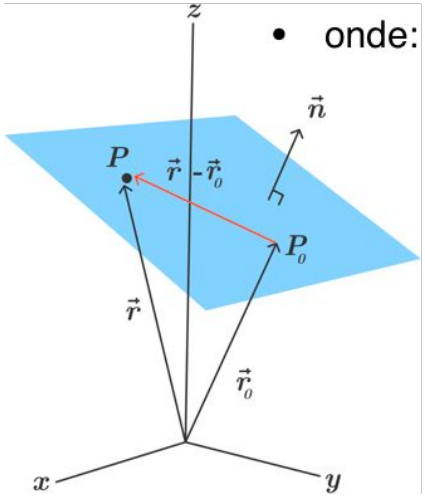


Support Vector Machines

Equação de um plano:

$$\overrightarrow{PP_0} \cdot \vec{n} = (\vec{r} - \vec{r_0}) \cdot \vec{n} = (x - x_0, y - y_0, z - z_0) \cdot (a, b, c) = 0$$

$$a(x - x_0) + b(y - y_0) + c(z - z_0) = 0$$



- onde:

$$ax + by + cz + d = 0$$

$$d = -(ax_0 + by_0 + cz_0)$$

- Usando a notação de ML:

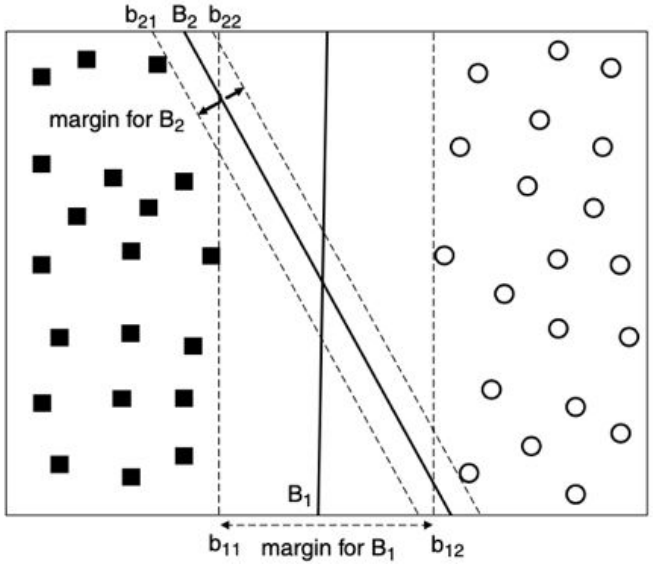
$$w_0 + w_1 x_1 + \ldots + w_d x_d + b = 0$$

$$\mathbf{w} \cdot \mathbf{x} + b = 0$$

Support Vector Machines

Ideia básica:

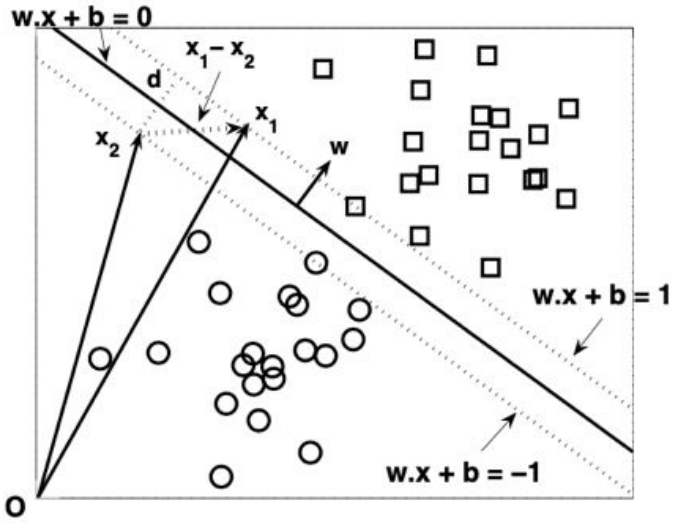
- Objetivamos maximizar a margem e a separação entre as classes.



Support Vector Machines

Método

- Para classes linearmente separáveis, a região de separação:



$$\mathbf{w} \cdot \mathbf{x} + b = 0,$$

w e **b** são os parâmetros do modelo.

Se x_a e x_b são pontos na superfície de decisão:

$$\mathbf{w} \cdot \mathbf{x}_a + b = 0,$$
$$\mathbf{w} \cdot \mathbf{x}_b + b = 0.$$

Subtraindo:

$$\mathbf{w} \cdot (\mathbf{x}_b - \mathbf{x}_a) = 0,$$

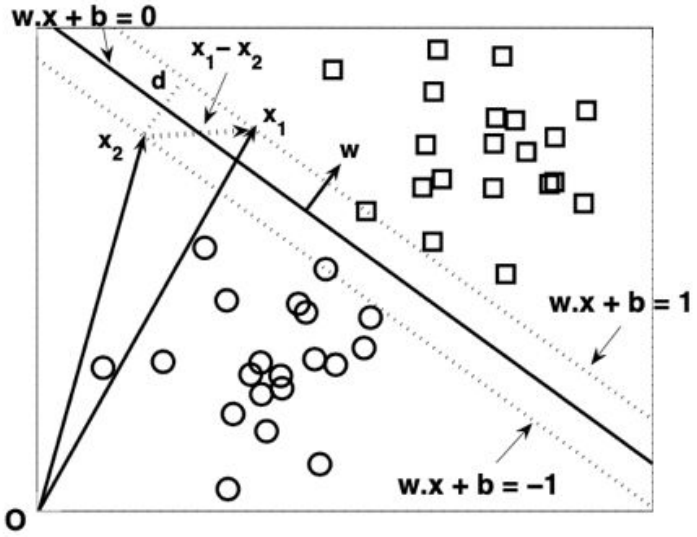
Ou seja, como o produto escalar é nulo, temos que w deve ser perpendicular à superfície de decisão.

Support Vector Machines

Método

- A regra de decisão:

$$y = \begin{cases} 1, & \text{if } \mathbf{w} \cdot \mathbf{z} + b > 0; \\ -1, & \text{if } \mathbf{w} \cdot \mathbf{z} + b < 0. \end{cases}$$



- Podemos estimar a margem de separação. Sejam dois hiperplanos paralelos:

$$b_{i1} : \ \mathbf{w} \cdot \mathbf{x} + b = 1,$$
$$b_{i2} : \ \mathbf{w} \cdot \mathbf{x} + b = -1.$$

- Subtraindo:

$$\mathbf{w} \cdot (\mathbf{x}_1 - \mathbf{x}_2) = 2$$
$$\|\mathbf{w}\| \times d = 2$$
$$\therefore d = \frac{2}{\|\mathbf{w}\|}.$$

Support Vector Machines

Método

- Ou seja, os parâmetros do modelo devem ser escolhidos tal que:

$$\mathbf{w} \cdot \mathbf{x_i} + b \geq \ 1 \text{ if } y_i = 1,$$
$$\mathbf{w} \cdot \mathbf{x_i} + b \leq -1 \text{ if } y_i = -1.$$

- Essa relação pode ser representada por uma única equação:

$$y_i(\mathbf{w} \cdot \mathbf{x_i} + b) \geq 1, \quad i = 1, 2, \ldots, N.$$

Support Vector Machines

Método

- Como queremos maximizar a margem de separação, podemos definir o problema para classes linearmente separáveis:
- **Definição**: A tarefa de aprendizado usando SVM pode ser sumarizado pelo problema de otimização:

$$\min_{\mathbf{w}} \frac{\|\mathbf{w}\|^2}{2}$$

$$\text{subject to} \qquad y_i(\mathbf{w} \cdot \mathbf{x_i} + b) \geq 1, \quad i = 1, 2, \ldots, N.$$

Support Vector Machines

$$\min_{\mathbf{w}} \frac{\|\mathbf{w}\|^2}{2}$$

$$\text{subject to} \qquad y_i(\mathbf{w} \cdot \mathbf{x_i} + b) \geq 1, \quad i = 1, 2, \ldots, N.$$

- Esse é um problema de otimização convexa e pode ser resolvido usando-se multiplicadores de Lagrange.

$$L_P = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{N} \lambda_i \Big(y_i(\mathbf{w} \cdot \mathbf{x_i} + b) - 1 \Big),$$

Otimização →

$$\frac{\partial L_p}{\partial \mathbf{w}} = 0 \Longrightarrow \mathbf{w} = \sum_{i=1}^{N} \lambda_i y_i \mathbf{x}_i,$$

$$\frac{\partial L_p}{\partial b} = 0 \Longrightarrow \sum_{i=1}^{N} \lambda_i y_i = 0.$$

Support Vector Machines

Método

- Como não sabemos os valores do multiplicadores de Lagrange, não podemos resolver para w e b.
- Restringindo ao caso em que os multiplicadores são não-negativos, podemos usar as condições de Karush-Kuhn-Tucker (KKT):

$$\lambda_i \geq 0,$$
$$\lambda_i \left[y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \right] = 0.$$

- Nesse caso, os multiplicadores são diferentes de zero apenas no caso em que i é um vetor de suporte (está no plano que define as margens).
- Usando esses resultados nas equações anteriores, obtemos uma nova versão da função:

$$L_D = \sum_{i=1}^{N} \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j y_i y_j \mathbf{x_i} \cdot \mathbf{x_j}.$$

Support Vector Machines

Método

$$L_D = \sum_{i=1}^{N} \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j y_i y_j \mathbf{x_i} \cdot \mathbf{x_j}.$$

- Resolvendo usando otimização quadrática, os multiplicadores de Lagrange.
- Inserindo em:

$$\frac{\partial L_p}{\partial \mathbf{w}} = 0 \Longrightarrow \mathbf{w} = \sum_{i=1}^{N} \lambda_i y_i \mathbf{x}_i,$$

$$\lambda_i \left[y_i (\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \right] = 0.$$

- E resolvendo, obtemos os parâmetros **w** e **b**.
- A superfície de decisão pode ser expressa por:

$$\left(\sum_{i=1}^{N} \lambda_i y_i \mathbf{x_i} \cdot \mathbf{x} \right) + b = 0.$$

Support Vector Machines

Método

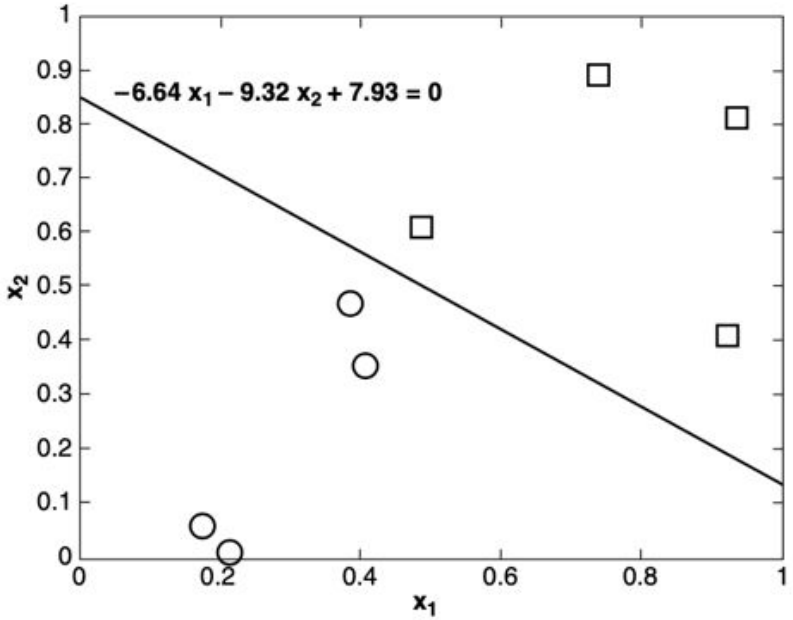
- Para classificar uma nova observação z, basta verificar o sinal da função:

$$f(\mathbf{z}) = sign\big(\mathbf{w} \cdot \mathbf{z} + b\big) = sign\Big(\sum_{i=1}^{N} \lambda_i y_i \mathbf{x_i} \cdot \mathbf{z} + b\Big).$$

- $f(\mathbf{z}) > 0$, z pertence à classe 1
- $f(\mathbf{z}) < 0$, z pertence à classe -1

Support Vector Machines

Exemplo:

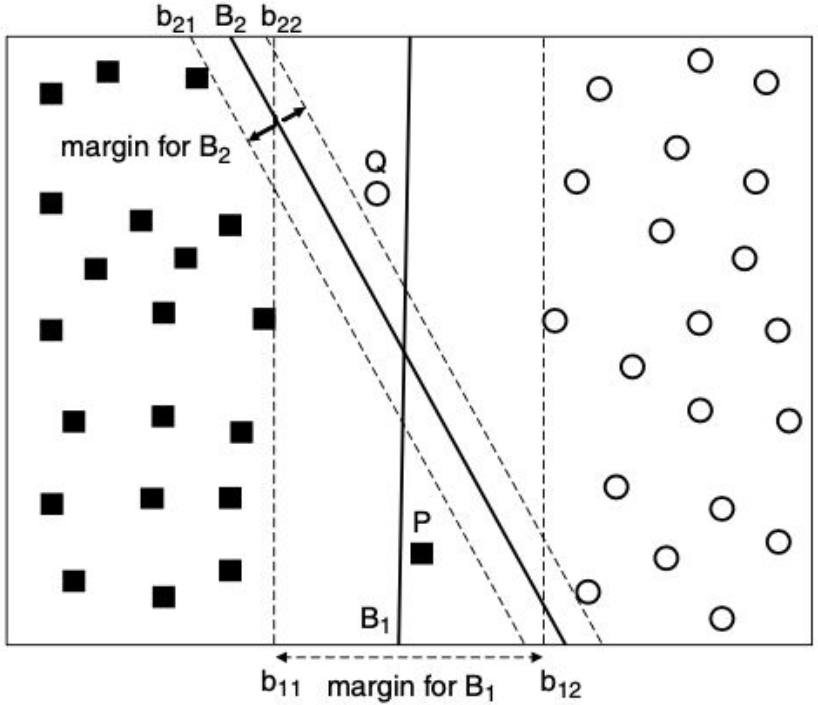


x_1	x_2	y	Lagrange Multiplier
0.3858	0.4687	1	65.5261
0.4871	0.611	–1	65.5261
0.9218	0.4103	–1	0
0.7382	0.8936	–1	0
0.1763	0.0579	1	0
0.4057	0.3529	1	0
0.9355	0.8132	–1	0
0.2146	0.0099	1	0

Support Vector Machines

Caso não separável

- Caso as classes não sejam perfeitamente separáveis linearmente, devemos estabelecer um equilíbrio entre o número de elementos classificados erroneamente e a largura da margem:



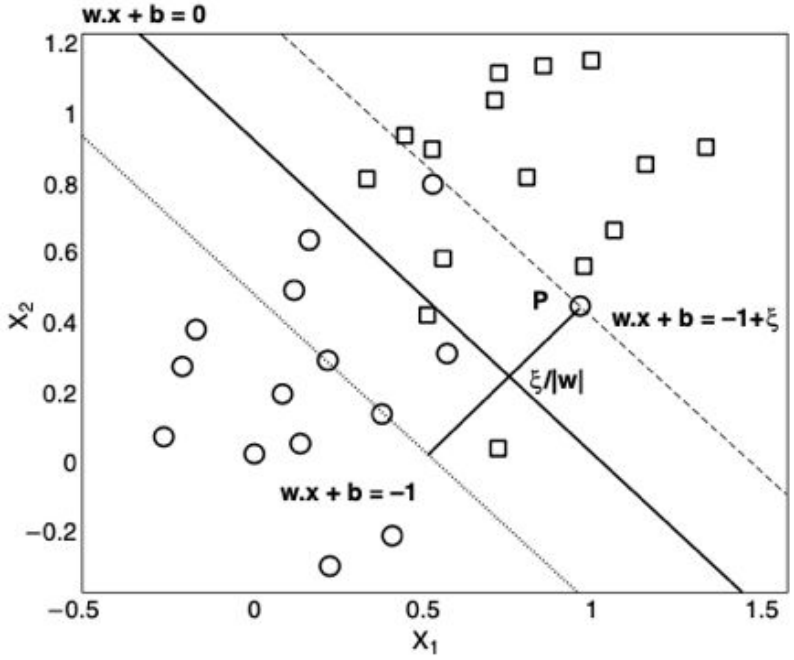
Support Vector Machines

Caso não separável

- Nesse caso, devemos definir uma margem de folga.

- Assim, temos que o problema de classificação se torna:

$$\mathbf{w} \cdot \mathbf{x_i} + b \geq 1 - \xi_i \quad \text{if } y_i = 1,$$
$$\mathbf{w} \cdot \mathbf{x_i} + b \leq -1 + \xi_i \quad \text{if } y_i = -1,$$
$$\xi_i > 0.$$



Support Vector Machines

Caso não separável

- Nesse caso, a nova função objetivo:

$$f(\mathbf{w}) = \frac{\|\mathbf{w}\|^2}{2} + C(\sum_{i=1}^{N} \xi_i)^k,$$

- E a função de Lagrange:

$$L_P = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\lambda_i\{y_i(\mathbf{w}\cdot\mathbf{x_i}+b) - 1 + \xi_i\} - \sum_{i=1}^{N}\mu_i\xi_i,$$

Support Vector Machines

Caso não separável

- Otimizando a Lagrangeana:

$$\frac{\partial L}{\partial w_j} = w_j - \sum_{i=1}^{N} \lambda_i y_i x_{ij} = 0 \implies w_j = \sum_{i=1}^{N} \lambda_i y_i x_{ij}.$$

$$\frac{\partial L}{\partial b} = -\sum_{i=1}^{N} \lambda_i y_i = 0 \implies \sum_{i=1}^{N} \lambda_i y_i = 0.$$

$$\frac{\partial L}{\partial \xi_i} = C - \lambda_i - \mu_i = 0 \implies \lambda_i + \mu_i = C.$$

Support Vector Machines

Caso não separável

- Usando as condições de Karush-Kuhn-Tucker:

$$\xi_i \geq 0, \quad \lambda_i \geq 0, \quad \mu_i \geq 0,$$
$$\lambda_i\{y_i(\mathbf{w}\cdot\mathbf{x}_i + b) - 1 + \xi_i\} = 0,$$
$$\mu_i\xi_i = 0.$$

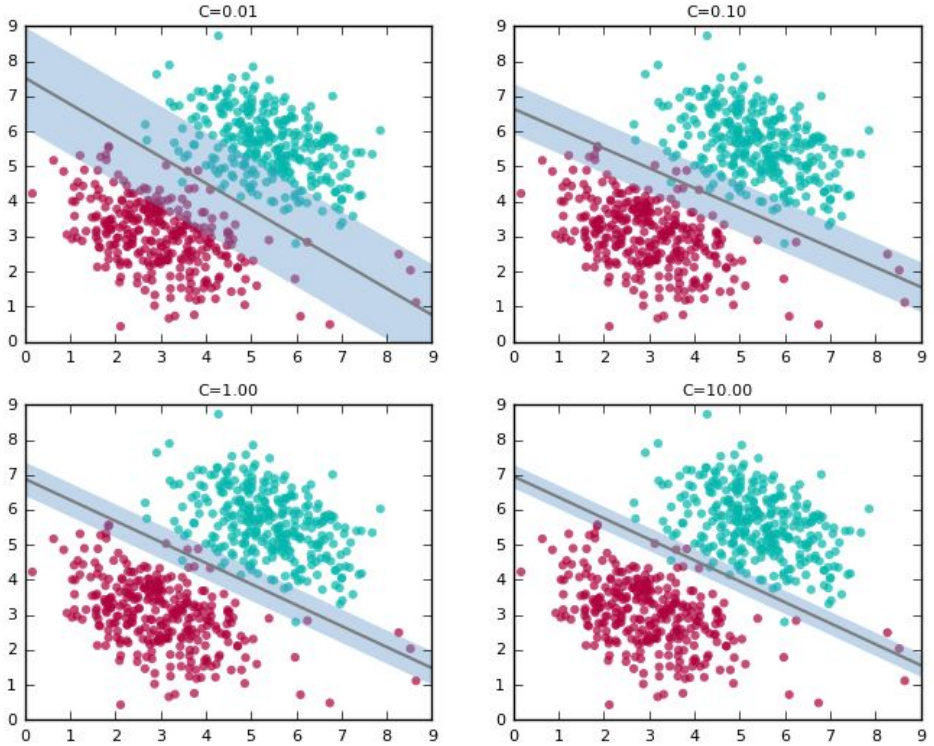
$$\begin{aligned} L_D &= \frac{1}{2}\sum_{i,j}\lambda_i\lambda_j y_i y_j \mathbf{x}_i\cdot\mathbf{x}_j + C\sum_i \xi_i \\ &\quad - \sum_i \lambda_i\{y_i(\sum_j \lambda_j y_j \mathbf{x}_i\cdot\mathbf{x}_j + b) - 1 + \xi_i\} \\ &\quad - \sum_i (C-\lambda_i)\xi_i \\ &= \sum_{i=1}^{N}\lambda_i - \frac{1}{2}\sum_{i,j}\lambda_i\lambda_j y_i y_j \mathbf{x_i}\cdot\mathbf{x_j}, \end{aligned}$$

- O procedimento restante é similar ao caso separável.

Support Vector Machines

Caso não separável

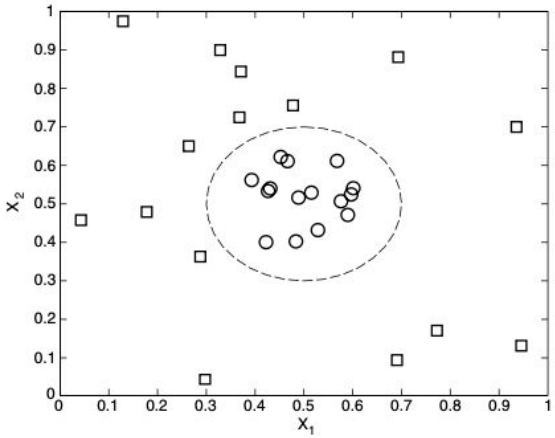
- O parâmetro C é definido pelo usuário.

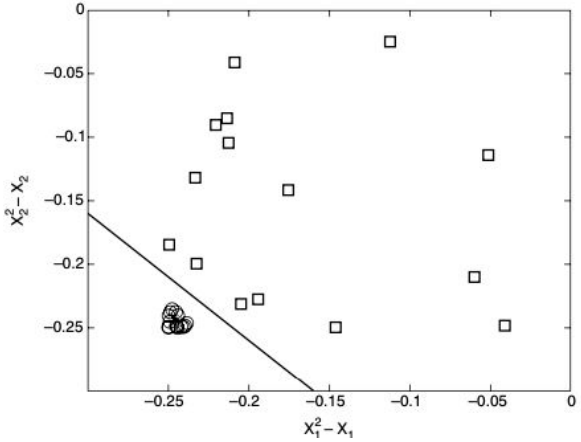


Support Vector Machines

Caso não linear:

- No caso não linear, a ideia é transformar os dados de forma que estes possam ser linearmente separados nesse novo espaço transformado.





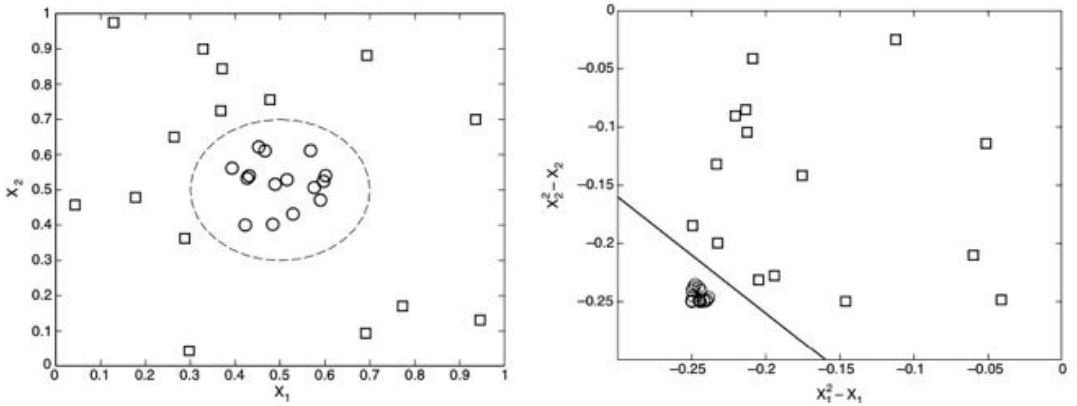
$$y(x_1, x_2) = \begin{cases} 1 & \text{if } \sqrt{(x_1 - 0.5)^2 + (x_2 - 0.5)^2} > 0.2, \\ -1 & \text{otherwise.} \end{cases}$$

Support Vector Machines

Caso não linear:

- Ou seja, escolhemos a transformação:

$$\Phi : (x_1, x_2) \longrightarrow (x_1^2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2, 1).$$



Support Vector Machines

Caso não linear:

- O problema de otimização se torna:

$$\min_{\mathbf{w}} \frac{\|\mathbf{w}\|^2}{2}$$

$$\text{subject to} \qquad y_i(\mathbf{w} \cdot \Phi(\mathbf{x}_i) + b) \geq 1, \quad i = 1, 2, \ldots, N.$$

Support Vector Machines

Caso não linear:

- Onde a Lagrangeana é dada por:

$$L_D = \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j y_i y_j \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$$

- Procedendo como anteriormente, os parâmetros w e b são obtidos a partir das equações:

$$\mathbf{w} = \sum_i \lambda_i y_i \Phi(\mathbf{x}_i)$$

$$\lambda_i \{ y_i (\sum_j \lambda_j y_j \Phi(\mathbf{x}_j) \cdot \Phi(\mathbf{x}_i) + b) - 1 \} = 0,$$

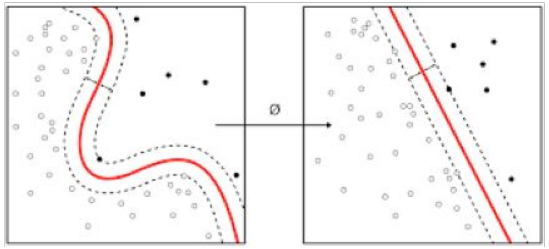
Support Vector Machines

Caso não linear:

- Para classificar uma nova observação z, basta verificar o sinal da função:

$$f(\mathbf{z}) = sign\big(\mathbf{w} \cdot \Phi(\mathbf{z}) + b\big) = sign\left(\sum_{i=1}^{n} \lambda_i y_i \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{z}) + b\right).$$

- $f(\mathbf{z}) > 0$, z pertence à classe 1
- $f(\mathbf{z}) < 0$, z pertence à classe -1



Support Vector Machines

Função kernel:

- O cálculo do produto escalar Φ(xi)·Φ(xj) pode ser complicado e sofre com o problema da maldição da dimensionalidade.
- Esse produto escalar pode ser entendido como uma medida de similaridade entre as observações xi e xj no espaço transformado.
- Por exemplo, para a transformação:

$$\Phi : (x_1, x_2) \longrightarrow (x_1^2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2, 1).$$

Temos:

$$\begin{aligned} \Phi(\mathbf{u}) \cdot \Phi(\mathbf{v}) &= (u_1^2, u_2^2, \sqrt{2}u_1, \sqrt{2}u_2, 1) \cdot (v_1^2, v_2^2, \sqrt{2}v_1, \sqrt{2}v_2, 1) \\ &= u_1^2 v_1^2 + u_2^2 v_2^2 + 2u_1 v_1 + 2u_2 v_2 + 1 \\ &= (\mathbf{u} \cdot \mathbf{v} + 1)^2. \end{aligned}$$

A função kernel:

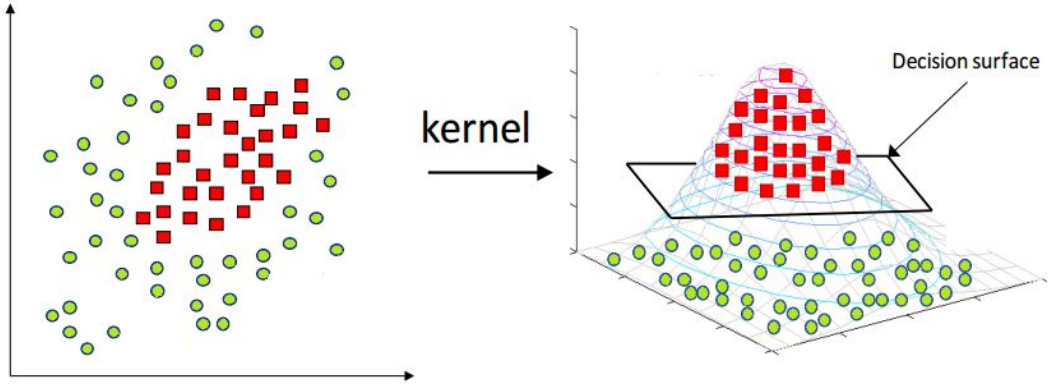
$$K(\mathbf{u}, \mathbf{v}) = \Phi(\mathbf{u}) \cdot \Phi(\mathbf{v}) = (\mathbf{u} \cdot \mathbf{v} + 1)^2.$$

Support Vector Machines

Função kernel:

- A função kernel permite operar no espaço de características original sem calcular as coordenadas dos dados no espaço de dimensão maior.
- Ex:

$$K(\mathbf{u}, \mathbf{v}) = \Phi(\mathbf{u}) \cdot \Phi(\mathbf{v}) = (\mathbf{u} \cdot \mathbf{v} + 1)^2.$$

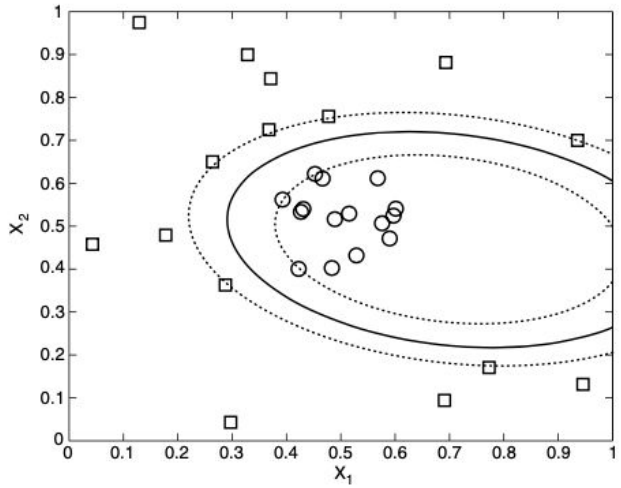


Support Vector Machines

Função kernel:

- Ex: Região de separação obtida com kernel não-linear:

$$K(\mathbf{u}, \mathbf{v}) = \Phi(\mathbf{u}) \cdot \Phi(\mathbf{v}) = (\mathbf{u} \cdot \mathbf{v} + 1)^2.$$



$$\begin{aligned} f(\mathbf{z}) &= sign(\sum_{i=1}^{n} \lambda_i y_i \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{z}) + b) \\ &= sign(\sum_{i=1}^{n} \lambda_i y_i K(\mathbf{x}_i, \mathbf{z}) + b) \\ &= sign(\sum_{i=1}^{n} \lambda_i y_i (\mathbf{x}_i \cdot \mathbf{z} + 1)^2 + b), \end{aligned}$$

Support Vector Machines

Função kernel:

- Exemplos de funções kernel:
- Kernel polinomial:

$$K(\mathbf{x}_i, \mathbf{x_j}) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^d$$

- Kernel gaussiano:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2\sigma^2})$$

- Gaussian radial basis function (RBF)

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma||\mathbf{x}_i - \mathbf{x}_j||^2)$$

- Hyperbolic tangent kernel

$$K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\kappa \mathbf{x}_i \cdot \mathbf{x}_j + c)$$

Support Vector Machines

Propriedades:

- SVM pode ser formulado como um problema de otimização convexa, sendo que há diversos softwares eficientes para encontrar o mínimo global.
- SVM pode ser usado com variáveis categóricas, desde que façamos a sua transformação para inteiros ou usar one-hot-encoding.
- SVM pode ser estendido para mais de duas classes.

Avaliação de modelos

Avaliação de modelos

- Não existe técnica de AM universal, que se saia melhor em qualquer tipo de problema (*No free lunch theorem*).
- Mesmo que um único algoritmo seja escolhido, variações de parâmetros produzem diferentes modelos.
- Como comparar modelos e parâmetros dos modelos?
- **Métricas para classificação:**
  - Taxa de erro
  - Acurácia
- **Métricas para regressão:**
  - Erro quadrático médio
  - Distância absoluta média

Avaliação de modelos

Taxa de erro de um classificador:

$$E(f) = \frac{1}{n} \sum_{i=1}^{n} I(y_i \neq f(\mathbf{x}_i))$$

- Onde I(.) é a função indicadora, sendo igual a 1 se a entrada for verdadeira.
- E(f) varia entre zero e um, sendo melhor quando for próximo de zero.

Avaliação de modelos

Taxa de acerto (acurácia)

$$\mathrm{Ac}(f) = 1 - E(f) = 1 - \frac{1}{n}\sum_{i=1}^{n} I(y_i \neq f(\mathbf{x}_i))$$

- Proporção de exemplos classificados corretamente em um conjunto com n objetos.
- Varia entre 0 e 1 e valores próximos de 1 são melhores.

Avaliação de modelos

Matriz de confusão

- Linhas representam classes verdadeiras.
- Colunas representam classes preditas.
  - Elemento Aij: número de exemplos da classe ci classificados como pertencentes à classe cj.
- Diagonal da matriz: acertos do classificador.
- Elementos fora da diagonal: erros cometidos.

Dados da Iris

Clase Predita	Classe Correcta		
	Setosa	Versicolor	Virgínica
Setosa	15	0	0
Versicolor	0	14	1
Virginica	0	1	4

Avaliação de modelos

Problema de duas classes:

- Tipos de erros:

Avaliação de modelos

Problema de duas classes:

- Medidas de desempenho:

		Classe Predita	
		Positivo	Negativo
Classe Verdadeira	Positivo	Verdadeiro Positivo (VP)	Falso Negativo (FN)
	Negativo	Falso Positivo (FP)	Verdadeiro Negativo (VN)

Sensibilidade: $VP/(VP + FN)$

Precisão: $VP/(VP + FP)$

Especificidade: $VN/(VN + FP)$

Avaliação de modelos

Problema de duas classes:

- Medidas de desempenho:
  - **Erro total:**

$$E(f) = \frac{FP + FN}{n}$$

  - **Acurácia total:**

$$\mathrm{Ac}(f) = \frac{VP + VN}{n}$$

  - **Precisão:**

$$\mathrm{Prec}(f) = \frac{VP}{VP + FP}$$

Verdadeiro Positivo (VP)	Falso Negativo (FN)
Falso Positivo (FP)	Verdadeiro Negativo (VN)

Avaliação de modelos

Problema de duas classes:

- Medidas de desempenho:
  - **Sensibilidade ou revocação:**

$$S(f) = \frac{VP}{VP + FN}$$

  - **Especificidade:**

$$\mathrm{Esp}(f) = \frac{VN}{VN + FP}$$

  - **Medida F1:**

$$\mathrm{F1}(f) = \frac{2S(f)\mathrm{Prec(f)}}{S(f) + \mathrm{Prec}(F)}$$

Verdadeiro Positivo (VP)	Falso Negativo (FN)
Falso Positivo (FP)	Verdadeiro Negativo (VN)

Avaliação de modelos

Problema de duas classes:

- **Exemplo:**
  - **Acurácia:**

$$\mathrm{Ac}(f) = \frac{VP + VN}{n} = \frac{70 + 60}{200} = 0.65$$

  - **Precisão:**

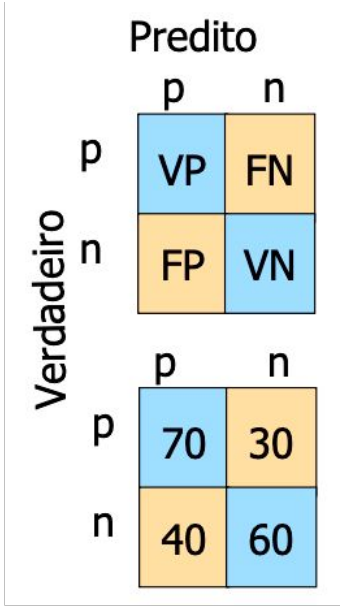
$$\mathrm{Prec}(f) = \frac{VP}{VP + FP} = \frac{70}{70 + 40} = 0.64$$

  - **Sensitividade:**

$$S(f) = \frac{VP}{VP + FN} = \frac{70}{70 + 30} = 0.7$$

  - **Especificidade:**

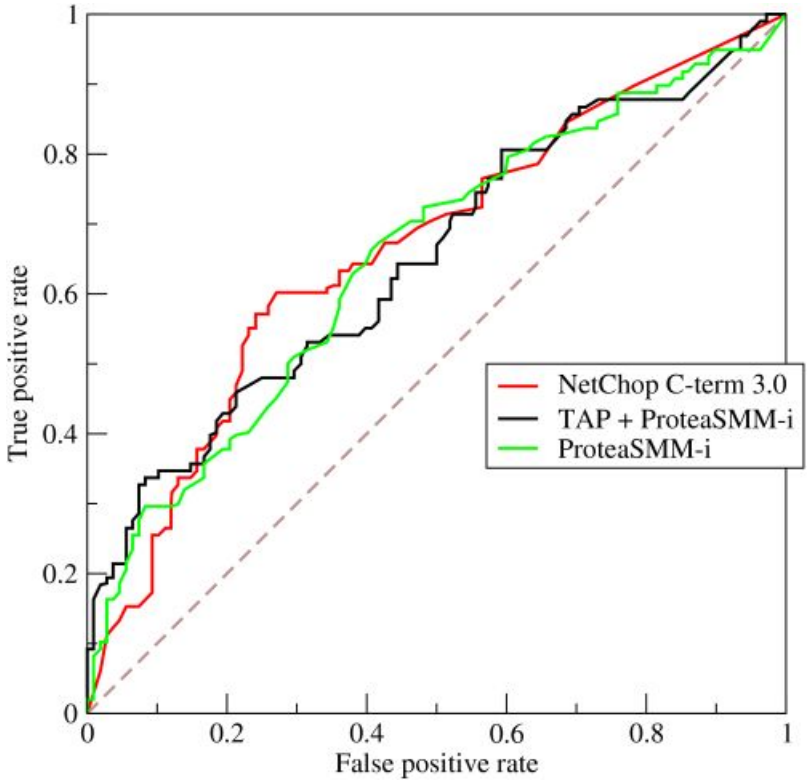
$$\mathrm{Esp}(f) = \frac{VN}{VN + FP} = \frac{60}{60 + 40} = 0.60$$



Avaliação de modelos

Problema de duas classes:

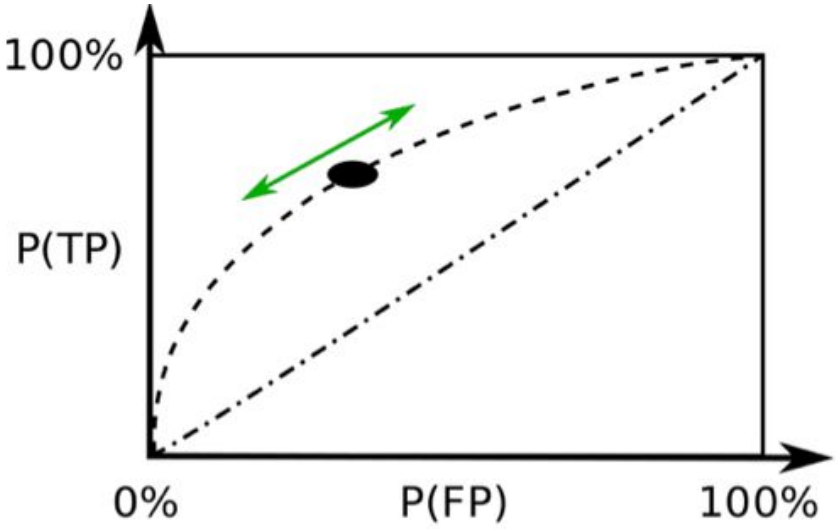
- Curva ROC (Receiving Operating Characteristics):

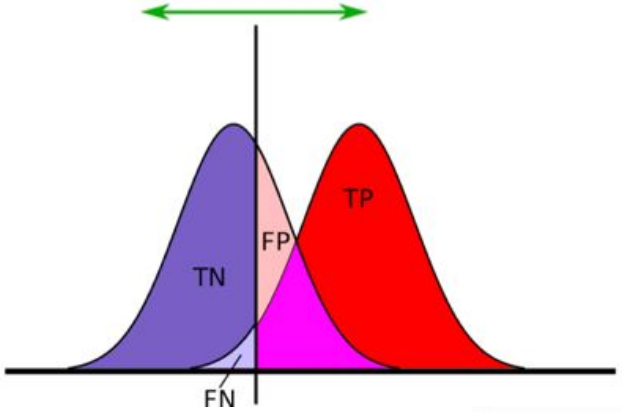


Avaliação de modelos

Problema de duas classes:

Curva ROC (Receiving Operating Characteristics):



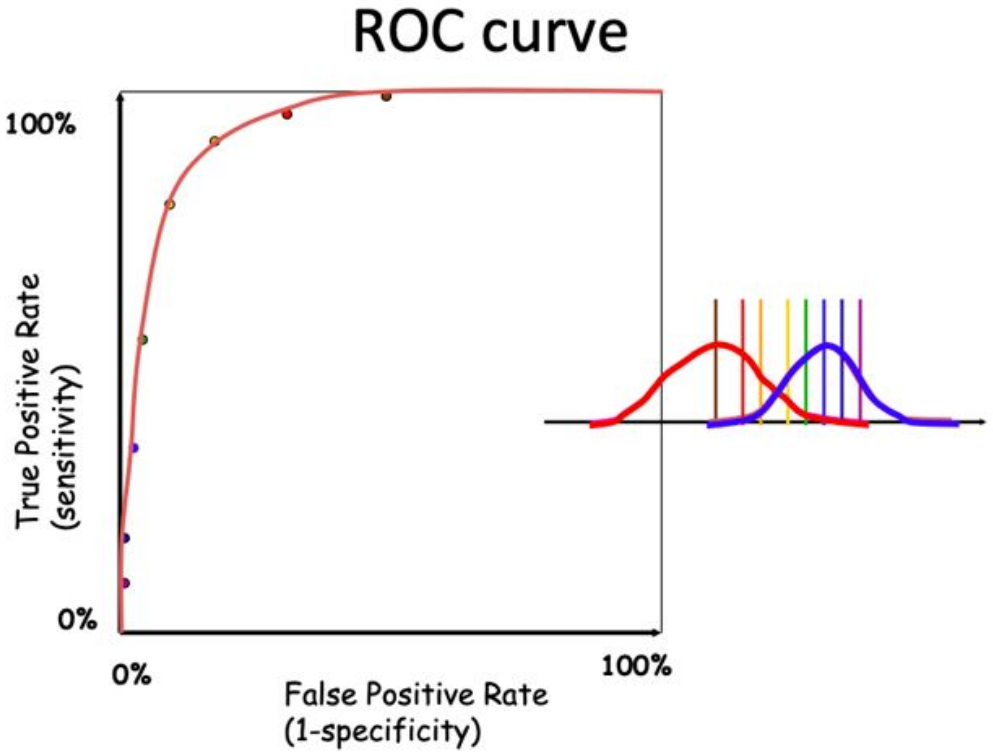


TP	FP
FN	TN

Avaliação de modelos

Problema de duas classes:

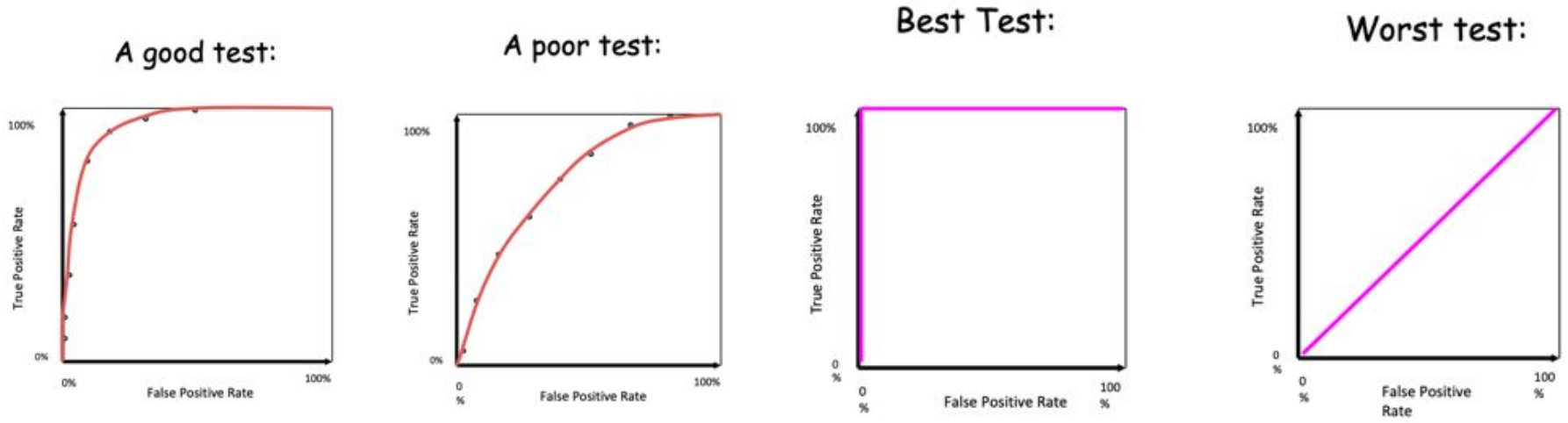
- Curva ROC (Receiving Operating Characteristics):



Avaliação de modelos

Problema de duas classes:

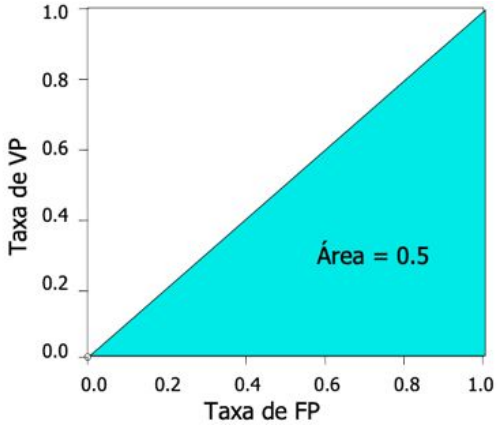
- Curva ROC (Receiving Operating Characteristics):

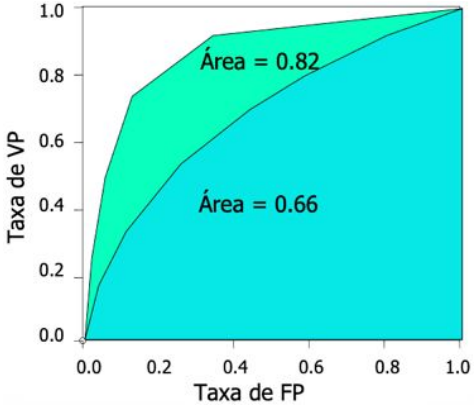


Avaliação de modelos

Problema de duas classes:

- Curva ROC (Receiving Operating Characteristics):
  - Área sob a curva ROC:
  - Produz valores no intervalo [0,1]
  - Valores mais próximos de 1 são considerados melhores.





Avaliação de modelos

Propriedades da Curva ROC:

- Permite realizar medidas de desempenho independentes do limiar de classificação e de custos associados às classificações incorretas e distribuição das classes.
- Uso de diferentes limiares representa maior ou menor ênfase à classe positiva.
- Taxa de erro/acerto é bastante sensível a desbalanceamentos (ex. Conjunto com 90 + e 10 -, taxa de acerto de 0,90 não necessariamente indica bom desempenho preditivo
- Desvantagem: análise originalmente limitada a classificação binária.

Sumário

- Support Vector Machines
- Avaliação de modelos

Leitura adicional

- Introduction to Data Mining, Tan, Steinbach, Karpatne, Kumar, Pearson, 2013.