Introdução a Ciências de Dados

Aula 6: Classificação

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

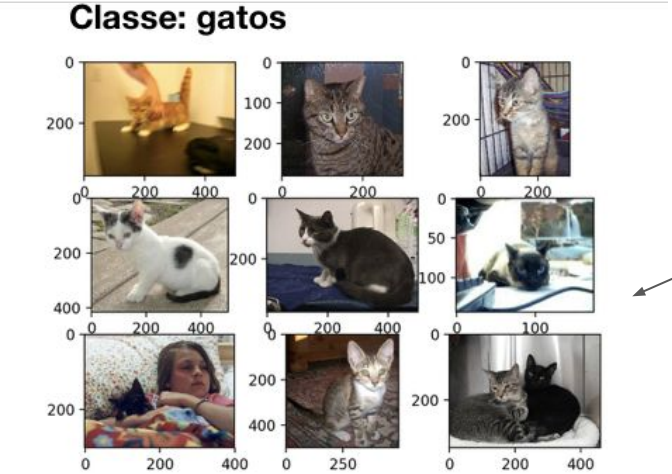
Aula 6: Classificação

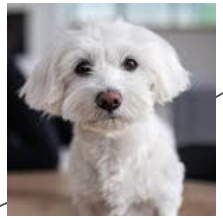
- k-vizinhos mais próximos
- Regressão Logística
- Naive Bayes

k-vizinhos mais próximos

K-vizinhos

- Uma maneira simples de definimos a classificação de objetos é através de distância entre eles:





?

K-vizinhos

- Como definir a distância entre objetos?

Vetor de atributos

$X_i = [x_{i1}, x_{i2}, x_{i3}, \ldots, x_{id}]$

Classe: dog

Distância (Xi,Xj)

$X_j = [x_{j1}, x_{j2}, x_{j3}, \ldots, x_{jd}]$

Classe: cat

K-vizinhos

- Precisamos definir uma medida de proximidade.
    - **Medida de similaridade**: d(Xi, Xi) é máxima.
        - Exemplo: Número de amigos compartilhados em uma rede social.
    - **Medida de dissimilaridade**: d(Xi,Xi) = 0.
        - Exemplo: distância entre cidades (distância Euclidiana).

K-vizinhos

Medida de dissimilaridade:

- d(p, q) ≥ 0 para todo p e q, e d(p, q) = 0 se, e somente se, p = q,
- d(p, q) = d(q,p) para todo p e q,
- d(p, r) ≤ d(p, q) + d(q, r) para todo p, q, e r, onde d(p, q) é a distância de dissimilaridade entre os pontos (objetos) p e q.

Medida de similaridade:

- s(p, q) = 1 (ou máximo de similaridade) se p = q,
- s(p, q) = s(q, p) para todo p e q, onde s(p, q) é a similaridade entre os objetos p e q.

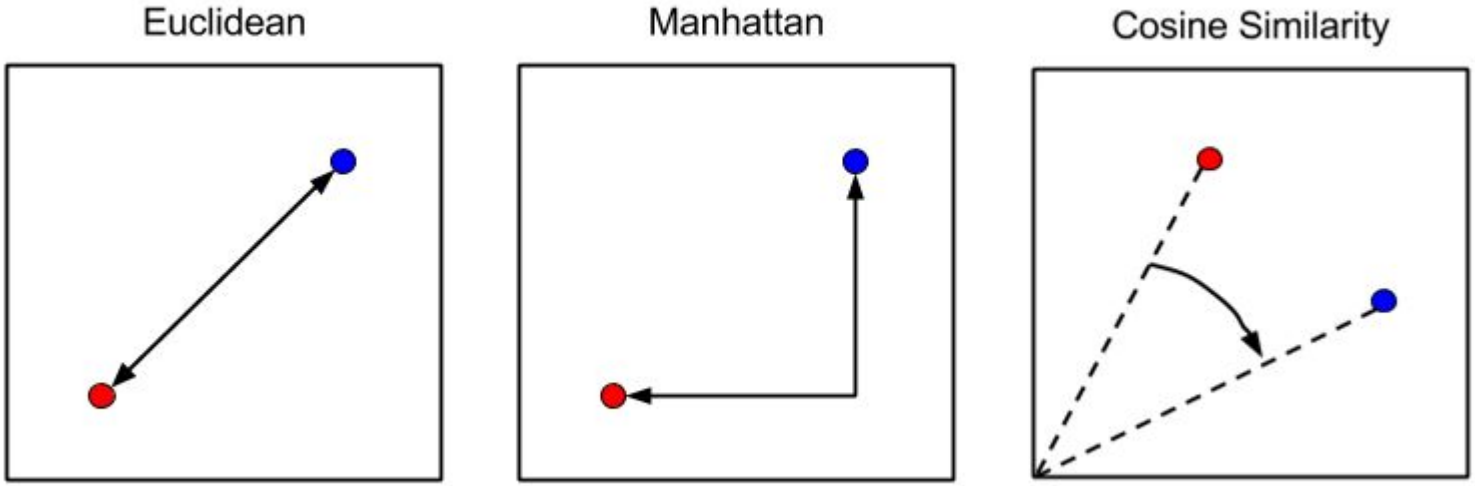
K-vizinhos

Métricas de distância:

- **Euclidiana** $D(X,Y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$ $[0,\infty)$
- **Minkowski** $D(X,Y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}}$ $[0,\infty)$
- **Cosseno** $D(X,Y) = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} (x_i)^2} \sqrt{\sum_{i=1}^{n} (y_i)^2}}$ $[0,1]$
- **Pearson** $D(X,Y) = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2} \cdot \sqrt{\sum_{i=1}^{n} (y_i - \bar{y})^2}}$ $[-1,1]$

K-vizinhos

Métricas de distância:



K-vizinhos

Métricas de distância:

- **Dados nominais**

Similaridade

$$s = \begin{cases} 1 & \text{if } p = q \\ 0 & \text{if } p \neq q \end{cases}$$

Dissimilaridade

$$d = \begin{cases} 0 & \text{if } p = q \\ 1 & \text{if } p \neq q \end{cases}$$

- **Dados ordinais**

Similaridade

$$s = 1 - \frac{\|p - q\|}{n - 1}$$

Dissimilaridade

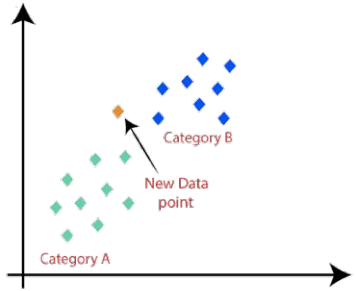
$$d = \frac{\|p - q\|}{n - 1}$$

K-vizinhos

Algoritmo:

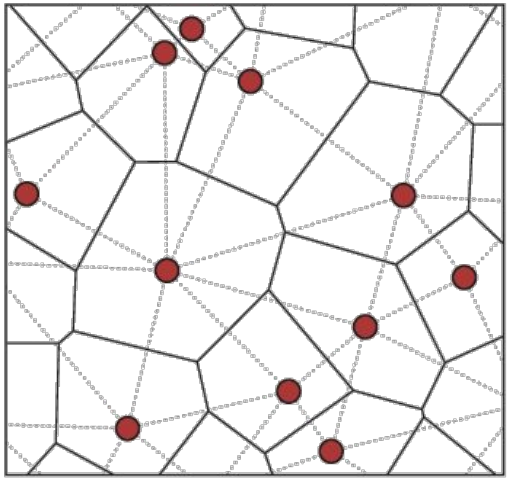
1. Identifique os k-vizinhos mais próximos do vetor de atributos **X** que se quer classificar.
2. Determine o número de vizinhos em cada classe.
3. Classifique **X** com pertencente à classe que resultou em um maior número de vizinhos (a moda entre o número de classes).

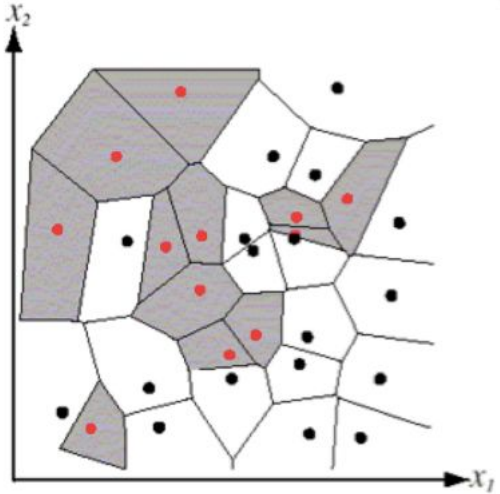
$$p(y = j|\mathbf{x}_\star) = \frac{1}{k} \sum_{i \in R_\star} \mathbb{I}\{y_i = j\}$$



K-vizinhos

Regiões de separação formam telhas de Voronoi.





K-vizinhos

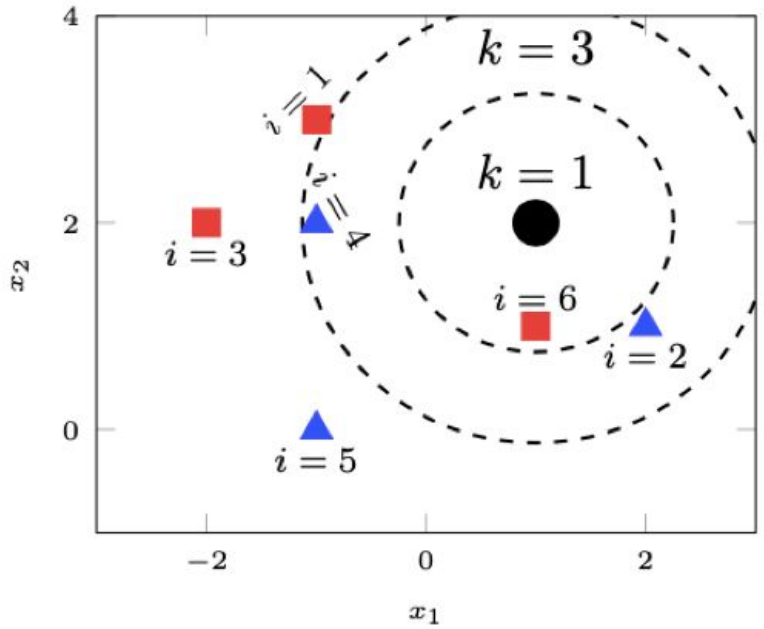
Exemplo:

Observação: $x_\star = [1\ 2]^T$

i	x_1	x_2	y
1	−1	3	Red
2	2	1	Blue
3	−2	2	Red
4	−1	2	Blue
5	−1	0	Blue
6	1	1	Red

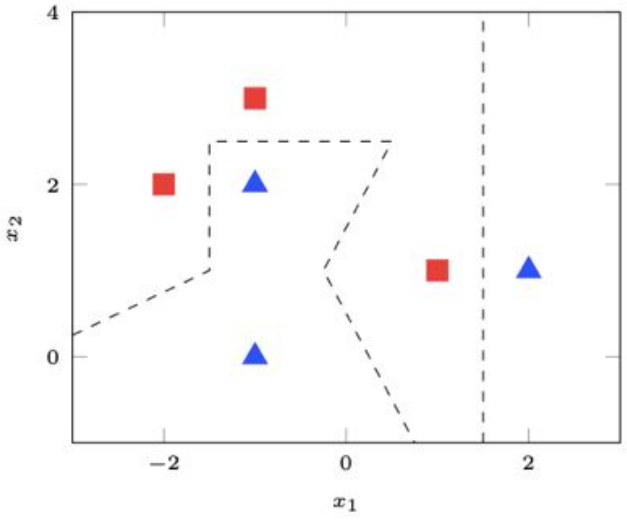
Distâncias

i	$\|\mathbf{x}_i - \mathbf{x}_\star\|$	y_i
6	$\sqrt{1}$	Red
2	$\sqrt{2}$	Blue
4	$\sqrt{4}$	Blue
1	$\sqrt{5}$	Red
5	$\sqrt{8}$	Blue
3	$\sqrt{9}$	Red

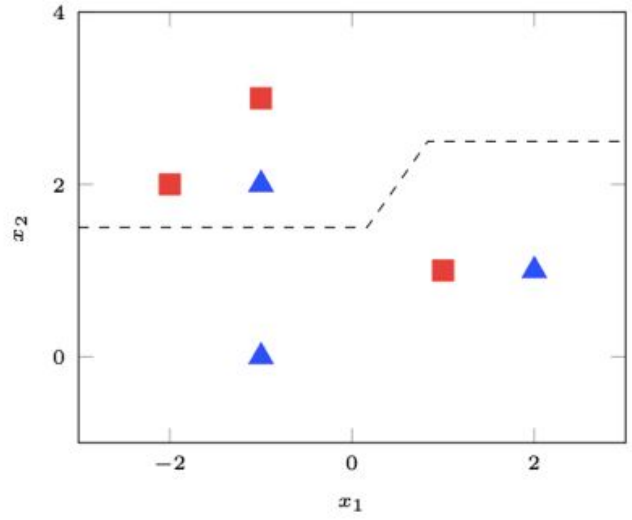


K-vizinhos

A região de decisão pode depender de k.



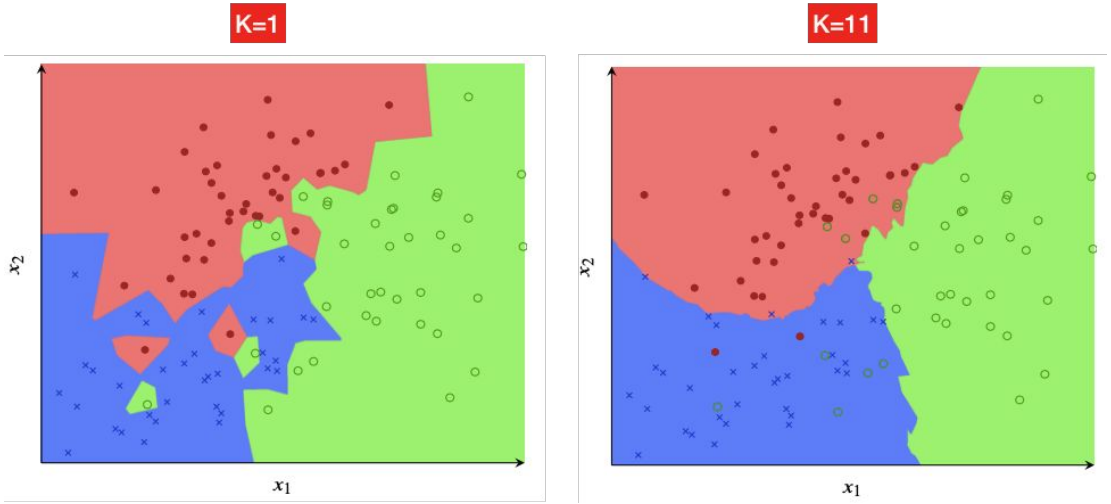
(a) $k = 1$



(b) $k = 3$

K-vizinhos

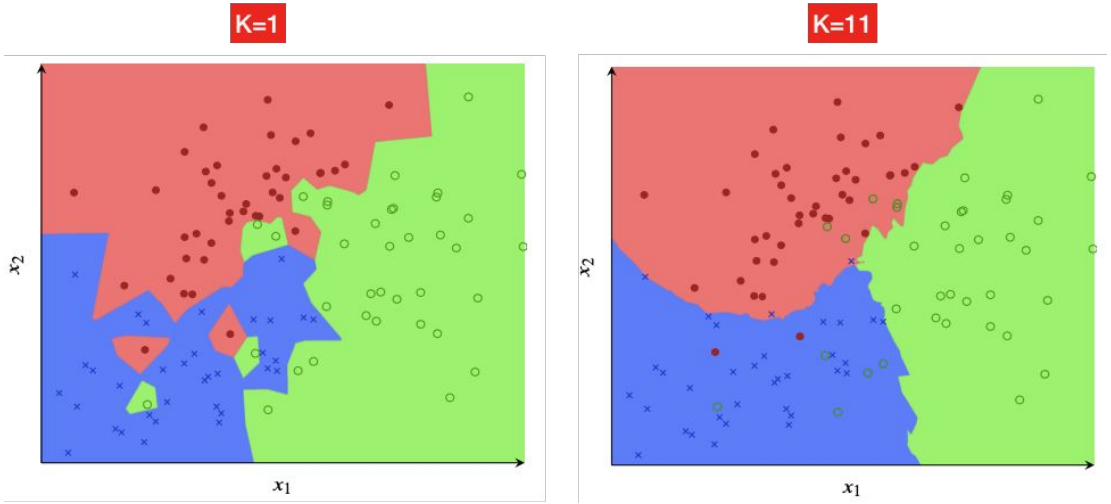
- Para k=1, vemos que a região de decisão se “**ajusta muito**” ao dados, ocorrendo overfitting.
- Para k = 11, vemos que região é bastante **suave**, o que sugere underfitting.



K-vizinhos

Qual o melhor valor de k?

A melhor maneira de encontrar o melhor valor de k é usar validação cruzada e uma medida para avaliar o resultado da classificação, como a acurácia.



K-vizinhos

Propriedades:

- O algoritmo não “aprende” um modelo, apenas memoriza objetos de treinamento
- Adia computação para a fase de classificação
- O algoritmo pode ser entendido como não paramétrico, dependendo apenas do número de vizinhos k.
- É um classificador não-linear, não sendo restrito a regiões de separação lineares.
- Como geralmente a distância Euclidiana é considerada, é necessário normalizar ou padronizar os dados.
- Dado que o conjunto de treinamento seja relativamente grande, pode-se provar que o erro cometido na classificação é no máximo duas vezes maior do que o classificador Bayesiano, que é ótimo.

Regressão Logística

Regressão Logística

- Vimos que modelos de regressão linear são dados por:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_d X_d = \beta^T \mathbf{x}$$

onde $\mathbf{x}^T = [1, x_1, x_2, \ldots, x_d]$

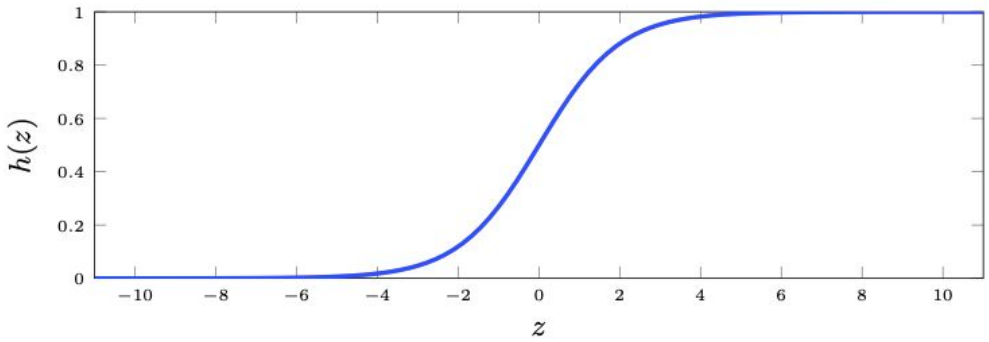
- Se considerarmos a saída ***Y*** como um valor inteiro, podemos usar o modelo de regressão para realizarmos a classificação de dados.
- Vamos definir as probabilidade para o caso de duas classes:

p(y=1|x) e **p(y=0|x)**

Regressão Logística

- Vamos considerar a função logística:

$$h(z) = \frac{e^z}{1 + e^z}$$



- Essa função retorna valores no intervalo [0,1].

Regressão Logística

- Usando a função logística, temos:

$$p(y = 1 \mid \mathbf{x}) = \frac{e^{\beta^T \mathbf{x}}}{1 + e^{\beta^T \mathbf{x}}} \qquad p(y = 0 \mid \mathbf{x}) = 1 - p(y = 1\mathbf{x}) = \frac{1}{1 + e^{\beta^T \mathbf{x}}}$$

- O aprendizado se resume a estimar o vetor de parâmetros **β** OBJ.
- Esse método é chamado regressão logística.

Regressão Logística

Para estimar os parâmetros do modelo, usamos o conjunto de treinamento $T = \{(x_i, y_i)\}_{i=1}^{N}$.

Usando estimação por máxima verossimilhança:

$$\hat{\beta} = \text{argmax}_{\beta} p(\mathbf{y} \,|\, \mathbf{X}; \beta\} \,.$$

Assim, a máxima verossimilhança:

$$\ell(\beta) = p(\mathbf{y} \,|\, \mathbf{X}; \beta\} = \prod_{i=1}^{N} p(y_i \,|\, x_i; \beta\} = \prod_{i:y_i=1} p(y = 1 \,|\, \mathbf{x_i}; \beta) \prod_{i:y_i=0} p(y = 0 \,|\, \mathbf{x_i}; \beta)$$

$$\ell(\beta) = \prod_{i:y_i=1} \frac{e^{\beta^T \mathbf{x}_i}}{1 + e^{\beta^T \mathbf{x}_i}} \prod_{i:y_i=0} \frac{1}{1 + e^{\beta^T \mathbf{x}_i}}$$

Regressão Logística

- Para estimar os parâmetros, precisamos otimizar essa função em relação à beta.

$$\ell(\beta) = \prod_{i:y_i=1} \frac{e^{\beta^T \mathbf{x}_i}}{1 + e^{\beta^T \mathbf{x}_i}} \prod_{i:y_i=0} \frac{1}{1 + e^{\beta^T \mathbf{x}_i}}$$

- Para facilitar os cálculos, podemos a função logaritmo, que é uma função monotônica e logo, o máximo é o mesmo que o da verossimilhança:

$$\log \ell(\beta) = \sum_{i:y_i=1} \left(\beta^T \mathbf{x}_i - \log(1 + e^{\beta^T \mathbf{x}_i}) \right) - \sum_{i:y_i=0} \log(1 + e^{\beta^T \mathbf{x}_i})$$

Regressão Logística

- Calculando a derivada e igualando a zero:

$$\nabla_\beta \log \ell(\beta) = \sum_{i=1}^{n} \mathbf{x}_i \left(y_i - \frac{e^{\beta^T \mathbf{x}_i}}{1 + e^{\beta^T \mathbf{x}_i}} \right) = 0$$

- Como essa equação é não-linear, devemos resolvê-la numericamente, por exemplo, usando o método de Newton–Raphson.
- Com isso, determinamos os parâmetros do modelo e realizamos a classificação em uma das duas classes.

Regressão Logística

- A superfície de decisão pode ser calculada usando:

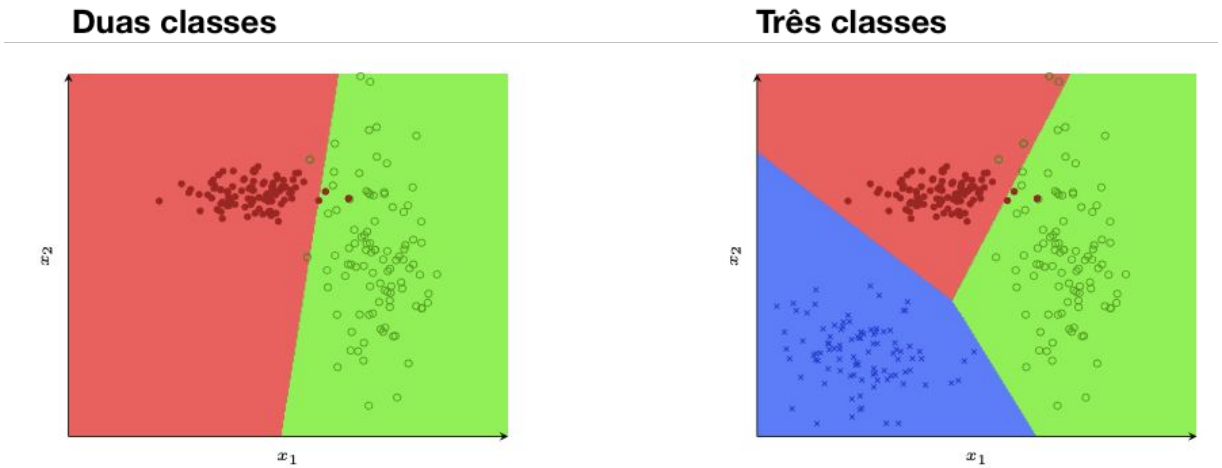
$$p(y = 1 \mid \mathbf{x}) = p(y = 0 \mid \mathbf{x})$$

$$\frac{e^{\beta^T \mathbf{x}}}{1 + e^{\beta^T \mathbf{x}}} = \frac{1}{1 + e^{\beta^T \mathbf{x}}}$$

- Ou seja, basta resolvermos: $\beta^T \mathbf{x} = 0$

- A solução são hiperplanos. Logo, a superfície de separação são hiperplanos (lineares).

Regressão Logística



- Para mais de duas classes, usamos one-hot-encoding e repetimos o mesmo procedimento.
  http://www.it.uu.se/edu/course/homepage/sml/literature/lecture_notes.pdf

Naive Bayes

Naive Bayes

Teoria da decisão Bayesiana:

- Dada **M** classes $\omega_1, \omega_2, ..., \omega_M$ e um padrão desconhecido **x**, determinar a probabilidade condicional $\mathbf{p(\omega_i|x)}$ do padrão pertencer a cada classe **i**.

OBJ

$$P(\omega_i|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\omega_i)P(\omega_i)}{p(\boldsymbol{x})}$$

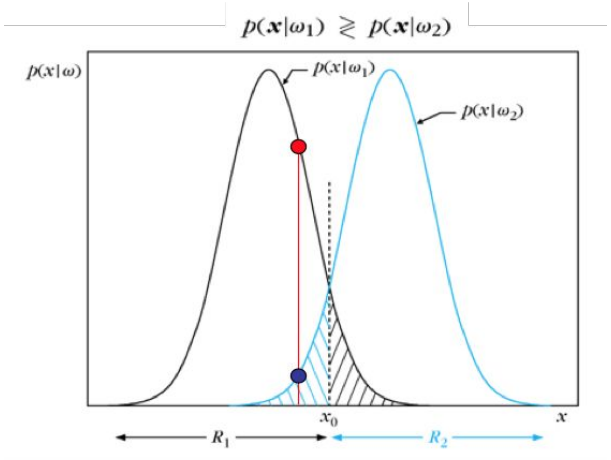
- Classificar de acordo com a classe mais provável.

$$\hat{y} = \underset{i \in \{1,\ldots,M\}}{\operatorname{argmax}} \; p(\omega_i|\mathbf{x})$$

Naive Bayes

Teoria da decisão Bayesiana:

- Se a probabilidade condicional for conhecida para cada classe, o erro obtido é o menor possível (classificador ótimo).

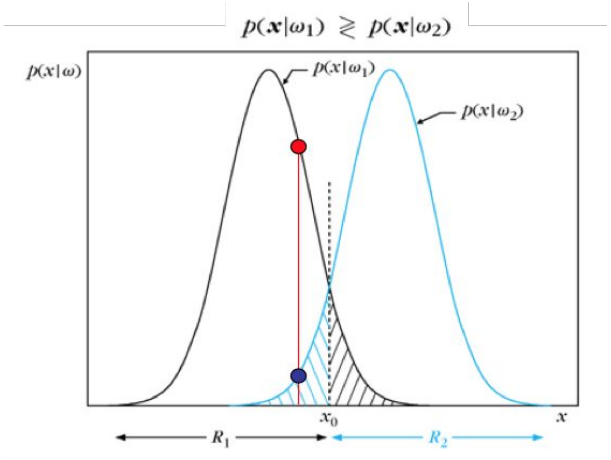


$R_1(\rightarrow \omega_1)$ e $R_2(\rightarrow \omega_2)$

Naive Bayes

Teoria da decisão Bayesiana:

- **Problema**: na maioria das vezes, não sabemos a distribuição de probabilidade conjunta e sua estimação é bastante complicada.



$R_1(\rightarrow \omega_1)$ e $R_2(\rightarrow \omega_2)$

Naive Bayes

Teoria da decisão Bayesiana:

- **Solução simples:** assumir que as variáveis que descreve os atributos são independentes.

$$P(\omega_i|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\omega_i)P(\omega_i)}{p(\boldsymbol{x})}$$

$$p(\mathbf{x}|\omega_i) = \prod_{j=1}^{d} p(x_j|\omega_i) \quad i = 1,2,\ldots,M$$

Naive Bayes

- Lembrem-se: Se duas variáveis X e Y aleatórias são independentes, então :

$$P(X|Y) = P(X) \longrightarrow P(X|Y) = \frac{P(X,Y)}{P(Y)} = P(X)$$

$$P(X,Y) = P(X)P(Y)$$

- Para variáveis aleatórias independentes, a distribuição de probabilidade conjunto é igual ao produto de suas marginais.

$$P(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{N} P(X_i) = P(X_1)P(X_2)\ldots P(X_n)$$

Naive Bayes

- Dada **M** classes $\omega_1, \omega_2, \ldots, \omega_M$ e um padrão desconhecido **x**, determinar a probabilidade condicional $\mathbf{p(\omega_i|x)}$ do padrão pertencer a cada classe **i**.

OBJ

$$P(\omega_i|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\omega_i)P(\omega_i)}{p(\boldsymbol{x})}$$

Classificar de acordo com a classe mais provável.

$$\hat{y} = \underset{i\in\{1,\ldots,M\}}{\operatorname{argmax}} \; p(\omega_i|\mathbf{x})$$

$$\hat{y} = \underset{i\in\{1,\ldots,M\}}{\operatorname{argmax}} \; p(\omega_i)\prod_{j=1}^{d} p(x_j \mid \omega_i)$$

Naive Bayes

Exemplo:

Classifique:

x = (Arrepios = Sim, Nariz escorrendo = Não, Dor de cabeça = média, febre = Não)

Arrepios	Nariz escorrendo	Dor de cabeça	febre	Gripe
Sim	Não	Média	Sim	Não
Sim	Sim	Não	Não	Sim
Sim	Não	Forte	Sim	Sim
Não	Sim	Média	Sim	Sim
Não	Não	Não	Não	Não
Não	Sim	Forte	Sim	Sim
Não	Sim	Forte	Não	Não
Sim	Sim	Média	Sim	Sim

Naive Bayes

Exemplo:

	Arrepios	
	Gripe = Sim	Gripe = Não
Arrepios = Sim	3/4	1/4
Arrepios = Não	2/4	2/4

	Febre	
	Gripe =Sim	Gripe = Não
Sim	4/5	1/5
Não	1/3	2/3

	Nariz	
	Gripe =Sim	Gripe = Não
Escorrendo = Sim	4/5	1/5
Escorrendo = Não	1/3	2/3

	Cabeça	
	Gripe = Sim	Gripe = Não
Dor = Forte	2/3	1/3
Dor = Média	2/3	1/3
Dor = Não	1/2	1/2

Podemos calcular todas as probabilidades conjuntas:

P(Arrepios=Sim, Gripe = Sim) = 3/4

P(Arrepios=Não, Gripe = Sim) = 2/4

P(Dor=Forte, Gripe = Sim) = 2/3

Naive Bayes

Exemplo:

- Vamos classificar os dados:

x = (Arrepios = Sim, Nariz escorrendo = Não,
Dor de cabeça = média, febre = Sim)

- Usando probabilidade condicional: **P(A|B)=P(B|A)P(A)**

P(Gripe=Sim| x) = P(x|Gripe=Sim)P(Gripe = Sim)

Naive Bayes

Exemplo:

- A probabilidade do paciente estar com gripe dados os sintomas:
  x = (Arrepios = Sim, Nariz escorrendo = Não, Dor de cabeça = média, febre = Não)
- P(Gripe=Sim) = 5/8
- P(Arrepios=Sim|Gripe=Sim) = 3/5
- P(Nariz escorrendo=Não|Gripe=Sim) = 1/5
- P(Dor de cabeça=Média|Gripe=Sim) = 2/5
- P(febre=Não|Gripe=Sim) = ⅕

- P(Gripe=Sim|x) = P(x|Gripe=sim)P(Gripe=Sim) =
- P(Arrepios=Sim|Gripe=Sim)*P(Escorrendo=Não|Gripe=Sim)*P(Dor=média|Gripe=Sim)P(Febre=Sim|Gripe=Sim)P(Gripe=Sim)= (3/5)*(1/5)*(2/5)*(1/5)*(5/8) = 0,006

P(Gripe=Sim|x) = 0,006

Naive Bayes

Exemplo:

- A probabilidade do paciente não estar com gripe dados os sintomas:

  x = (Arrepios = Sim, Nariz escorrendo = Não, Dor de cabeça = média, febre = Não)

- P(Gripe=Não) = 3/8
- P(Arrepios=Sim|Gripe=Não) = 1/4
- P(Nariz escorrendo=Não|Gripe=Não) = 2/3
- P(Dor de cabeça=Média|Gripe=Não) = 1/3
- P(febre=Não|Gripe=Não) = ⅔

- P(Gripe=Não|x) = P(x|Gripe=Não)P(Gripe=Não) = P(Arrepios=Não|Gripe=Sim)*P(Escorrendo=Não|Gripe=Não)*P(Dor=média|Gripe=Não)* P(Febre=Não|Gripe=Não)*P(Gripe=Não)=(1/4)*(2/3)*(1/3)*(2/3)*(3/8) = 0,013

P(Gripe=Não|x) = 0,013

Naive Bayes

Exemplo:

- Assim, temos:

P(Gripe=Sim|x) = 0,006
P(Gripe=Não|x) = 0,013

- Classificando de acordo com a classe mais provável.

$$\hat{y} = \underset{i \in \{1,\ldots,M\}}{\operatorname{argmax}} \; p(\omega_i \,|\, \mathbf{x})$$

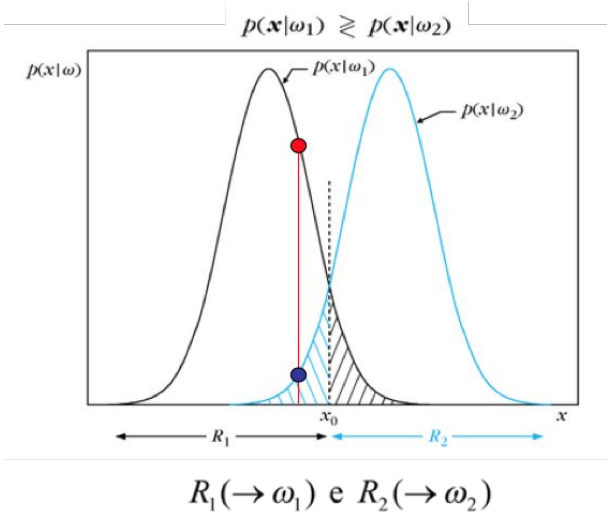
- **Concluímos que o paciente não está gripado.**

x = (Arrepios = Sim, Nariz escorrendo = Não, Dor de cabeça = média, febre = Sim)

Naive Bayes

Caso os atributos sejam contínuos, podemos assumir uma distribuição de probabilidade (geralmente Normal) e realizar a classificação maximizando a verossimilhança.

$$p(x_j \mid \omega_i) = \frac{1}{\sqrt{2\pi\sigma_{\omega_i}}} \exp\left(-\frac{(x_j - \mu_{\omega_i})^2}{2\sigma_{\omega_i}}\right)$$



$R_1(\rightarrow \omega_1)$ e $R_2(\rightarrow \omega_2)$

Naive Bayes

Algoritmo (para atributos continuos):

- Calcule a média e variância de cada atributo para cada classe.
- Calcule a verossimilhança para cada atributo dentro de cada classe.

OBJ

$$p(x_j \mid \omega_i) = \frac{1}{\sqrt{2\pi\sigma_{\omega_i}}} \exp\left(-\frac{(x_j - \mu_{\omega_i})^2}{2\sigma_{\omega_i}}\right)$$

- Assuma independência e calcule a distribuição conjunta.

OBJ

$$p(\mathbf{x} \mid \omega_i) = \prod_{j=1}^{d} p(x_j \mid \omega_i) \quad i = 1,2,\ldots,M$$

- Classifique de acordo com a classe mais provável.

OBJ

$$\omega_m = \arg\max_{\omega_i} \prod_{j=1}^{d} p(x_j \mid \omega_i), \quad i = 1,2,\ldots,M$$

Naive Bayes

Propriedades:

- Apesar da limitação em assumir independência dos atributos, o classificador Naive Bayes é robusto e apresenta boa performance para muitos dados reais.
- Todas as probabilidades exigidas podem ser calculadas dos dados de treinamento em uma passagem.
- Construção do modelo é bastante eficiente.
- Fácil de estender para incremental.
- Robusto a ruídos e atributos irrelevantes.

Sumário

- k-vizinhos mais próximos,
- Regressão Logística,
- Naive Bayes.

Leitura adicional

- Lindholm et al., Supervised Machine Learning, 2019. http://www.it.uu.se/edu/course/homepage/sml/literature/lecture_notes.pdf

- James et al., Introduction to statistical learning with applications in R, 2014. https://www.ime.unicamp.br/~dias/Intoduction%20to%20Statistical%20Learning.pdf