Aprendizagem de Máquina

Alessandro L. Koerich / Alceu S. Britto

Programa de Pós-Graduação em Informática
Pontifícia Universidade Católica do Paraná (PUCPR)

Aprendizagem Baseada em Instâncias

Plano de Aula

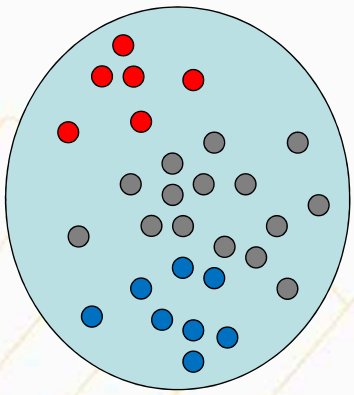
- Introdução
- Espaço Euclidiano
- Aprendizagem Baseada em Instâncias
- Regra *k* vizinhos mais próximos (k-NN)
- Exemplos

Introdução

- O problema central de aprendizagem é induzir funções gerais a partir de exemplos de treinamento específicos.
- Muitos métodos de aprendizagem constroem uma descrição geral e explícita da função alvo a partir de exemplos de treinamento (AD, NB, etc...).
- Os métodos de aprendizagem baseados em instâncias simplesmente armazenam os exemplos de treinamento.
- A generalização é feita somente quando uma nova instância tiver que ser classificada.

Paradigmas de Aprendizagem

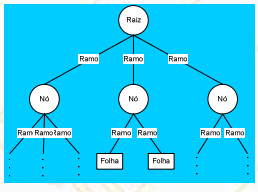
D (exemplos de treinamento)



treinamento

C (classes)

c_1
c_2
c_3



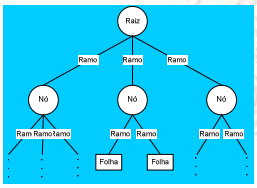
Árvore

$P(c_1), P(D|\ c_1)$
$P(c_2), P(D|\ c_2)$
$P(c_3), P(D|\ c_3)$

Bayes

Paradigmas de Aprendizagem

- Na classificação...



Árvore de Decisão

x_t (exemplo de teste)

Valor do Conceito Alvo (ou classe)

$P(c_1), P(D| c_1)$
$P(c_2), P(D| c_2)$
$P(c_3), P(D| c_3)$

Bayes

Introdução

D (exemplos de treinamento)

Comparação (distância)

x_t (exemplo de teste)

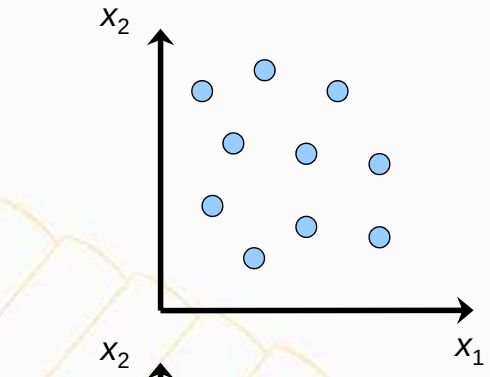
- Paradigma baseado em instâncias...

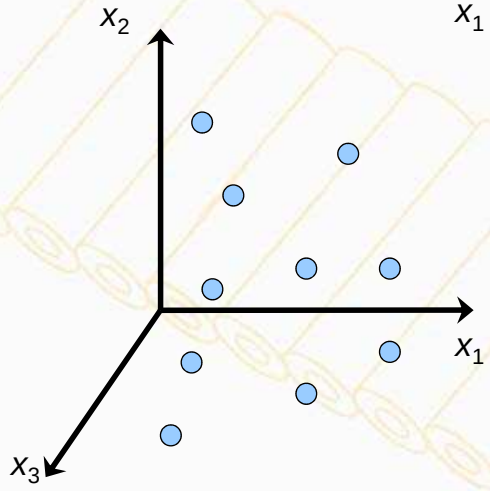
Valor do Conceito Alvo (exemplo de treinamento mais próximo)

Introdução

- Métodos de aprendizagem baseados em instâncias assumem que as instâncias podem ser representadas como pontos em um espaço Euclidiano.

Espaço Euclidiano





Vetores de Características ou Atributos

x_1	x_2

2-dimensional

x_1	x_2	x_3

3-dimensional

Espaço Euclidiano

Vetores de Características ou Atributos

?

x_1 x_2 x_3 x_4 4-dimensional

?

x_1 x_2 x_3 x_4 . . . x_n *n*-dimensional

Espaço Euclidiano?

Vetores de Características ou Atributos

sunny *high* 2-dimensional

sunny *high* *strong* 3-dimensional

sunny *high* *strong* *cool* 4-dimensional

Introdução

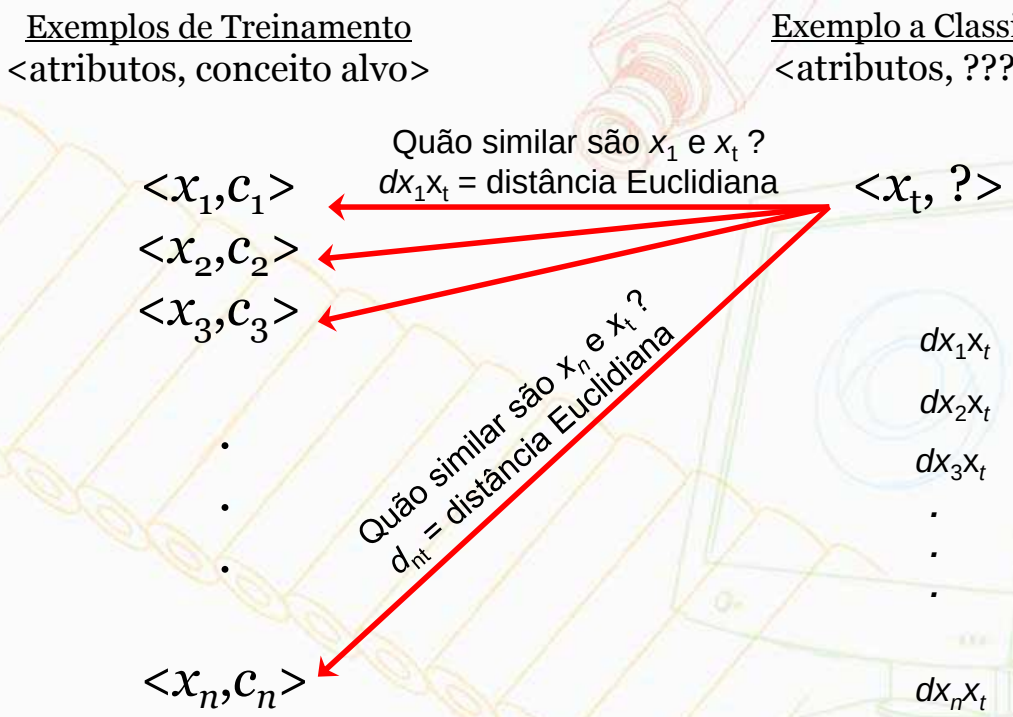
- Os métodos de aprendizagem baseados em instâncias são métodos não paramétricos.
- **Métodos não paramétricos**: podem ser usados com distribuições arbitrárias e sem a suposição de que a forma das densidades são conhecidas.

Aprendizagem Baseada em Instâncias

- A aprendizagem consiste somente em armazenar os exemplos de treinamento $<x_1,c_1>$, $<x_2,c_2>$ . . . $<x_n,c_n>$.

- Após a aprendizagem, para encontrar o valor do conceito alvo (classe) associado a uma instância de testes $<x_t, ? >$, um conjunto de instâncias similares são buscadas na memória e utilizadas para classificar a nova instância.

Aprendizagem Baseada em Instâncias



Aprendizagem Baseada em Instâncias

- No final teremos um conjunto de distâncias (medida de similaridade) entre a instância de teste x_t e todos as instâncias de treinamento x_1, x_2, . . ., x_n
- Qual valor de conceito alvo (classe) atribuímos a instância x_t?

O conceito alvo associado ao exemplo de treinamento mais similar !!

Aprendizagem Baseada em Instâncias

- Isto é, pegamos a instância de treinamento cuja distância seja a menor e verificamos a classe associada a esta instância.
- Suponha que a distância dx_9x_t seja a menor entre as n distâncias avaliadas, logo a instância mais próxima da instância de teste x_t é $<x_9,c_9>$.
- Assim, atribuímos à x_t a classe associada à x_9, ou seja, c_9!!!

Aprendizagem Baseada em Instâncias

- Observações importantes:
  - Constroem uma aproximações para a função alvo para cada instância de teste diferente.
  - Constrói uma aproximação local da função alvo.
  - Podem utilizar representações mais complexas e simbólicas para as instâncias
  - Uma desvantagem é o alto custo para classificação.
    - Toda computação ocorre no momento da classificação !!!
    - Aumenta com a quantidade de exemplo de treinamento.

Aprendizagem *k*–NN

- *k*-NN = *k* Nearest Neighbor = *k* Vizinhos mais Próximos
- O algoritmo *k*–NN é o método de aprendizagem baseado em instâncias mais elementar.
- O algoritmo *k*–NN assume que todas as instâncias correspondem a pontos em um espaço *n*–dimensional.

$\Re^n$

- Os "*vizinhos mais próximos*" de uma instância são definidos em termos da distância Euclidiana.

Regra *k*–NN

- A regra dos vizinhos mais próximos:

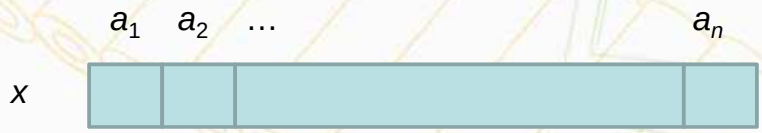
Meta: Classificar x_t atribuindo a ele o rótulo representado mais freqüentemente dentre as *k* amostras mais próximas e utilizando um esquema de votação.

Aprendizagem *k*–NN

- Vamos considerar uma instância arbitrária *x* que é descrita pelo vetor de características:

$$x = \langle a_1(x), a_2(x), ..., a_n(x) \rangle$$

onde $a_r(x)$ representa o valor do *r*–ésimo atributo da instância *x*.



Aprendizagem *k*–NN

- Então a distância Euclidiana entre duas instâncias x_i e x_j é definida como $d(x_i, x_j)$, onde:

$$d(x_i, x_j) \equiv \sqrt{\sum_{r=1}^{n} \left(a_r(x_i) - a_r(x_j)\right)^2}$$

Algoritmo *k*–NN

Algoritmo de treinamento

- Para cada exemplo de treinamento <*x*,*c*>, adicione o exemplo à lista *training_examples*

Algoritmo de classificação

- Dada uma instância x_t a ser classificada,
  - Faça x_1, ...,x_k representar as *k* instâncias de *training_examples* que estão mais próximas de x_t
  - Retorne

$$f(x_t) \leftarrow \arg\max_{c \in C} \sum_{i=1}^{k} \delta(c, f(x_i))$$

onde $\delta(a,b)=1$ se $a=b$ e $\delta(a,b)=0$ caso contrário.

Aprendizagem *k*–NN

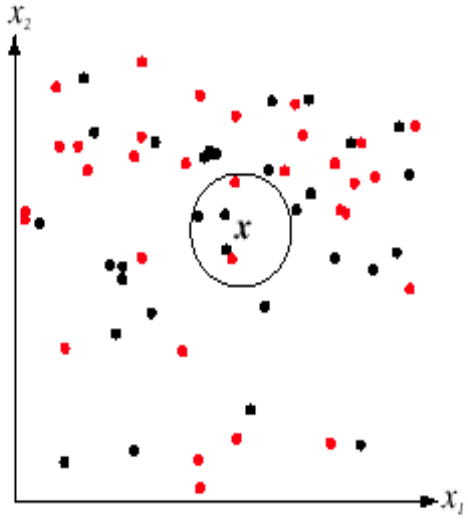


FIGURE 4.15. The *k*-nearest-neighbor query starts at the test point **x** and grows a spherical region until it encloses *k* training samples, and it labels the test point by a majority vote of these samples. In this $k = 5$ case, the test point **x** would be labeled the category of the black points. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Regra *k*–NN

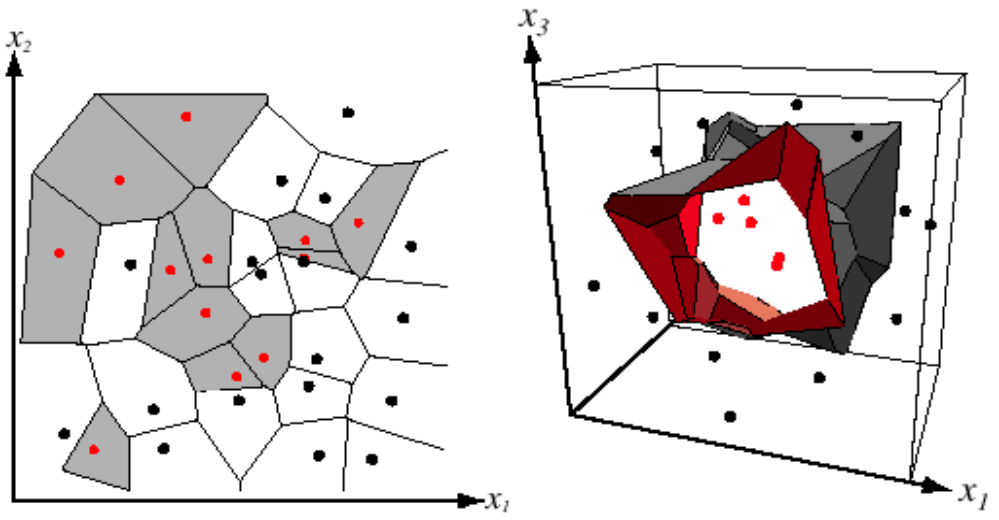
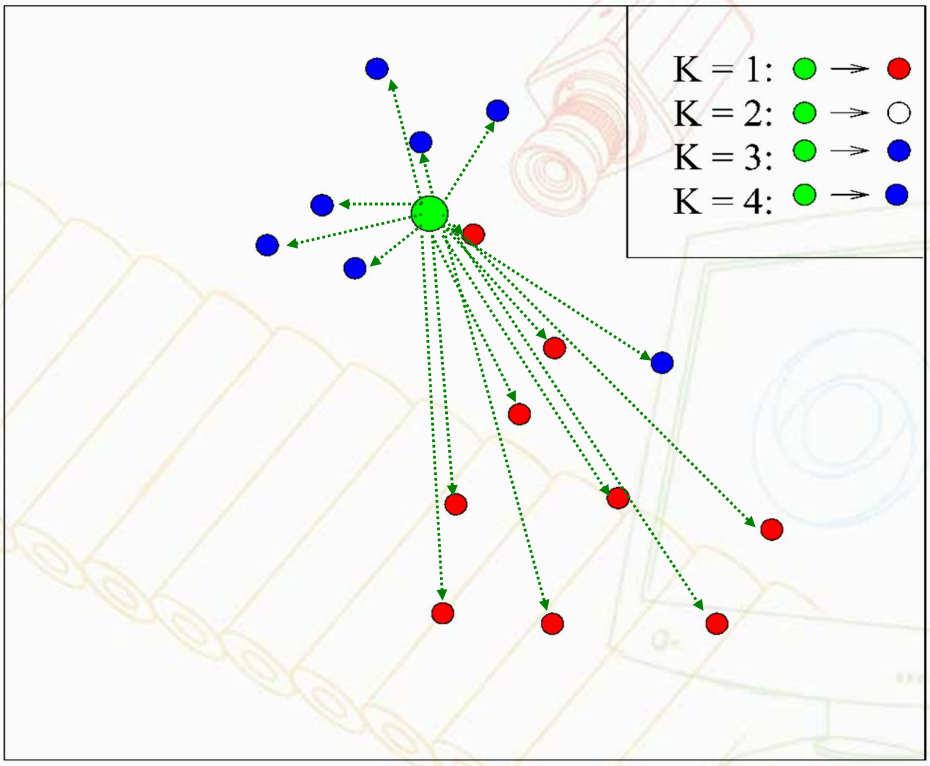
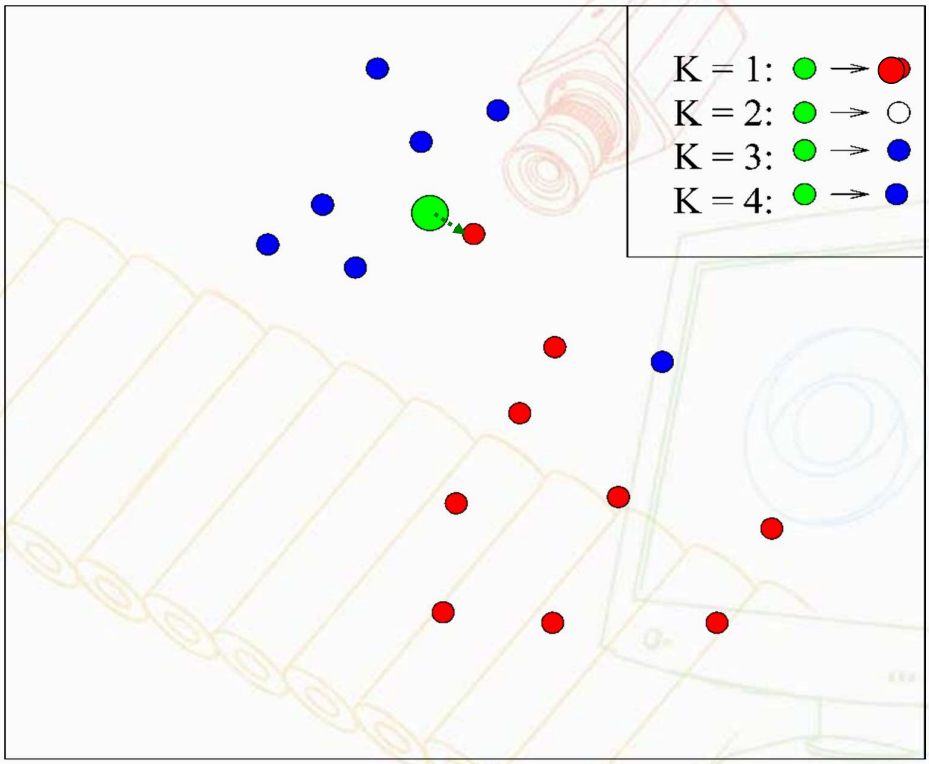


FIGURE 4.13. In two dimensions, the nearest-neighbor algorithm leads to a partitioning of the input space into Voronoi cells, each labeled by the category of the training point it contains. In three dimensions, the cells are three-dimensional, and the decision boundary resembles the surface of a crystal. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

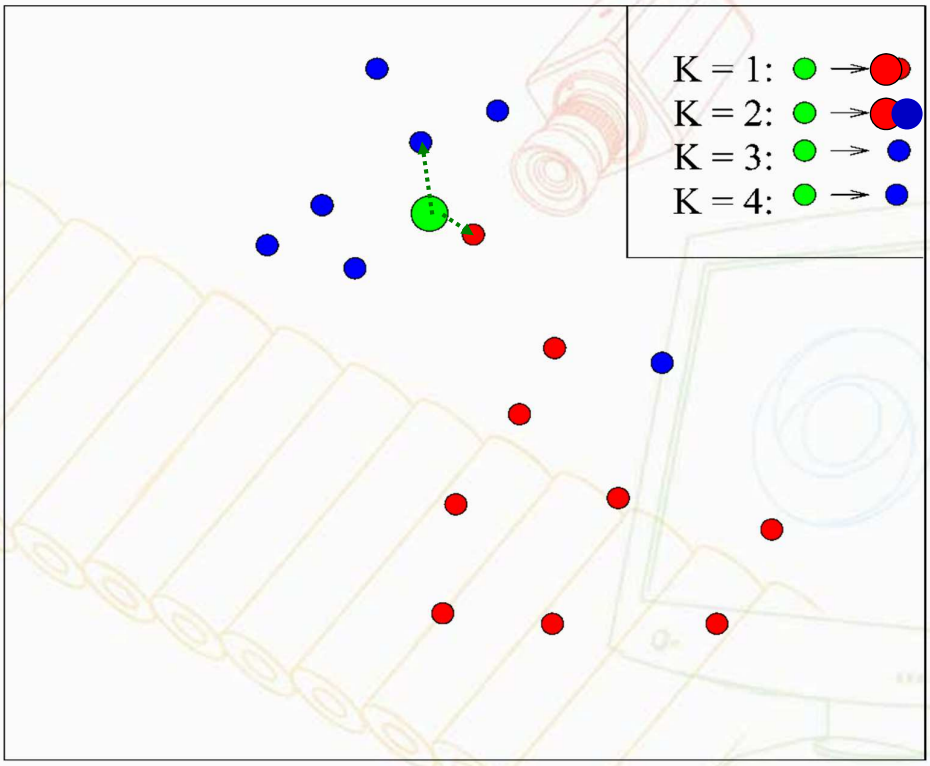
Algoritmo *k*–NN



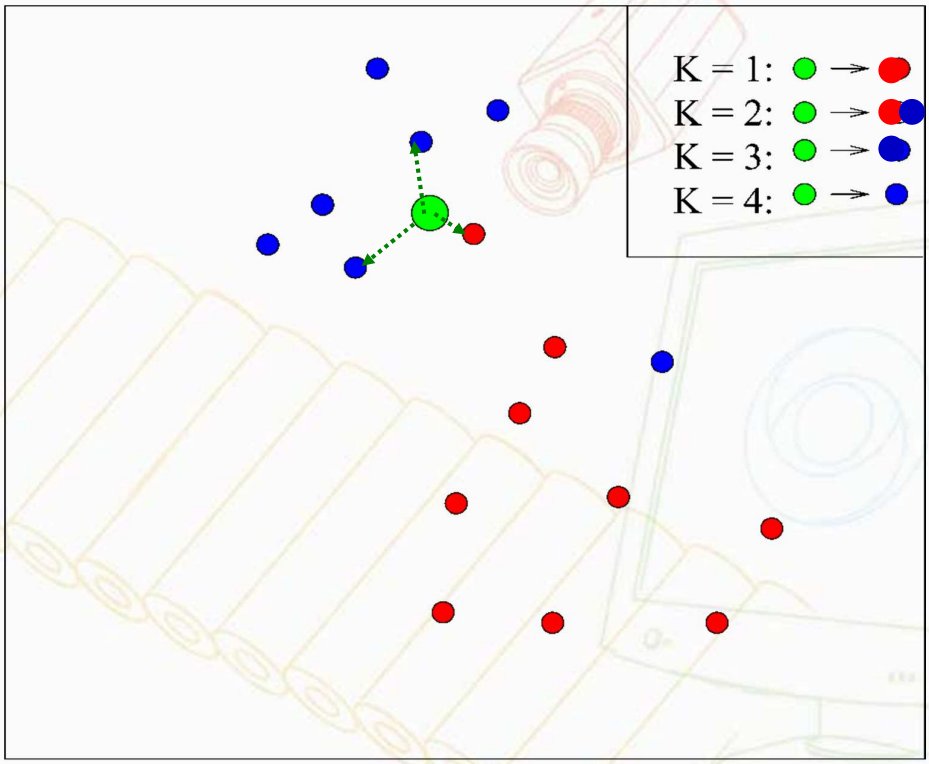
Algoritmo *k*–NN



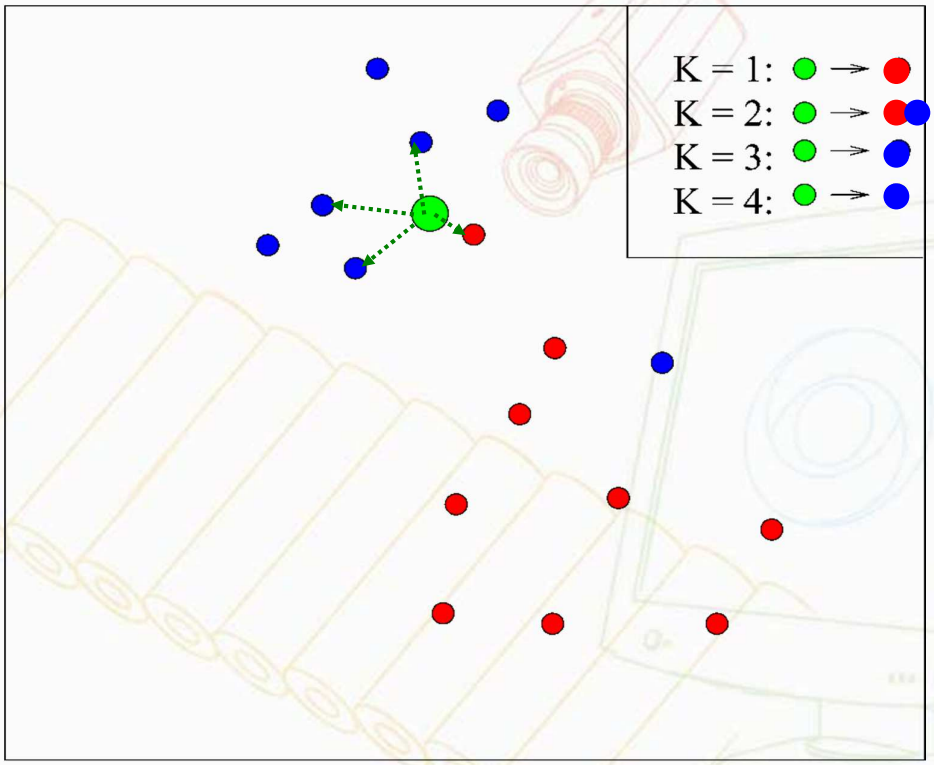
Algoritmo *k*–NN



Algoritmo *k*–NN



Algoritmo *k*–NN



Aprendizagem *k*–NN

- Supondo:
  - *n* exemplos de treinamento rotulados;
  - Vetores com *d* dimensões;
  - buscamos o exemplo mais próximo de uma instância de teste x_t (k=1).
- A eficiência do algoritmo *k–NN* = $O(dn)$:
  - inspeciona cada exemplo armazenado, um após o outro;
  - calcula a distância Euclidiana até x_t [$O(d)$];
  - retém a identidade somente do mais próximo;

Aprendizagem *k*–NN

- Existem técnicas para reduzir a computação do algoritmo *k*–NN:
  - distância parcial;
  - pré–estruturação;
  - edição dos protótipos armazenados;

Consultar pgs 185 e 186, R. O. Duda, P. E. Hart e D. G. Stork, *Pattern Classification*, Wiley Interscience, 2001.

k–NN com Distância Ponderada

- Refinamento do k–NN:
  - Ponderar a contribuição de cada um dos *k* vizinhos de acordo com suas distâncias até o ponto x_t que queremos classificar, dando maior peso aos vizinhos mais próximos.

- Podemos ponderar o voto de cada vizinho, de acordo com o quadrado do inverso de sua distância de x_t.

$$\hat{f}(x_t) \leftarrow \arg\max_{c \in C} \sum_{i=1}^{k} w_i \delta(c, f(x_i)) \qquad w_i \equiv \frac{1}{d(x_t, x_i)^2}$$

- Porém, se $x_t = x_i$, o denominador $d(x_t,x_i)^2$ torna–se zero. Neste caso fazemos $f(x_t) = f(x_i)$.

k–NN com Distância Ponderada

- Todas as variações do algoritmo k–NN consideram somente os k vizinhos mais próximos para classificar o ponto desconhecido.

- Uma vez incluída a ponderação pela distância, não há problemas em considerar todos os exemplos de treinamento:
  - Exemplos muito distantes terão pouco efeito em $f(x_t)$.
  - Desvantagem: mais lento

Resumo

- Métodos de aprendizagem baseados em instâncias não necessitam formar uma hipótese explícita da função alvo sobre o espaço das instâncias.
- Eles formam uma aproximação local da função alvo para cada nova instância a “classificar”.

Resumo

- O *k*–NN é um algoritmo baseado em instâncias para aproximar funções alvo de valor real ou de valor discreto, assumindo que as instâncias correspondem a pontos em um espaço *d*–dimensional.
- O valor da função alvo para um novo ponto é estimada a partir dos valores conhecidos dos *k* exemplos de treinamento mais próximos.

Resumo

- Vantagens:
  - habilidade para modelar funções alvo complexas por uma coleção de aproximações locais menos complexas.
  - A informação presente nos exemplos de treinamento nunca é perdida.
- Dificuldades:
  - Tempo ?
  - Determinação de uma métrica.