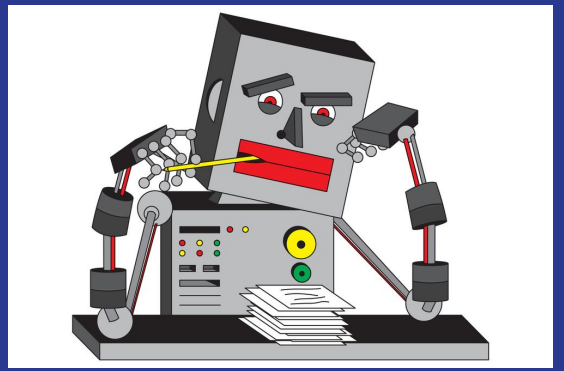
CIC1205
APRENDIZADO DE MÁQUINA

Prof. Eduardo Bezerra
ebezerra@cefet-rj.br
CEFET/RJ - PPCIC



K NEAREST NEIGHBORS

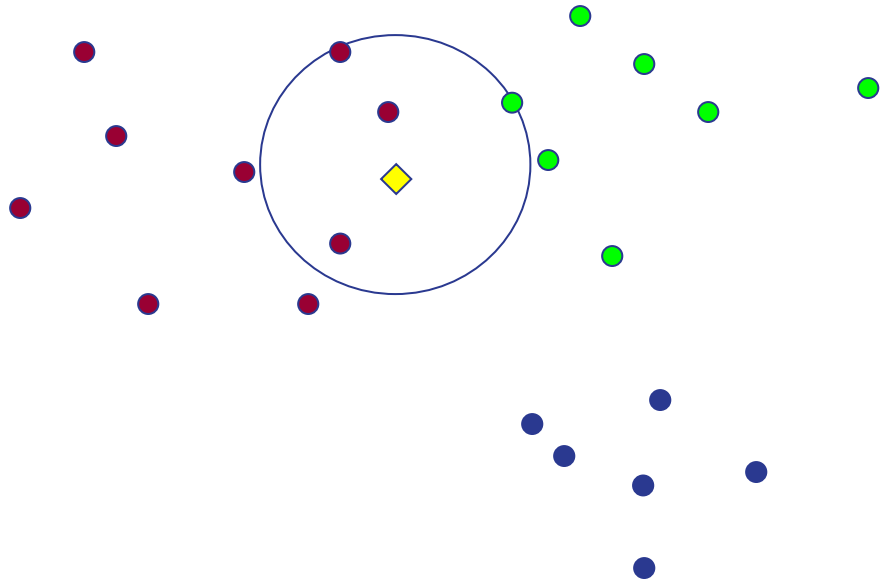
kNN

- Um dos métodos de AM mais simples.
- Pode ser usado para classificação ou regressão.
- Terminologia relacionada:
  - *case-based learning*,
  - *memory-based learning*,
  - *instance-based learning*.

kNN (classificação)

- Para categorizar um exemplo **x**, o kNN produz uma ordem total sobre os “vizinhos” de **x**, retirados do conjunto de treinamento X.
  - Vizinho no contexto de similaridade.
- As classes dos *k* vizinhos mais similares são usadas para predizer a classe de **x**.
- As categorias desses vizinhos são ponderadas pelas similaridades deles em relação a **x**.
  - Quanto mais similar o vizinho, mais influência tem sua categoria na determinação da classe de **x**.

kNN (classificação): exemplo



Pr(ciência|◆)?

Pr(governo|◆)?

Pr(artes|◆)?

Governo

Ciência

Artes

kNN (classificação)

- Definir a k-vizinhança (k-neighborhood) de um exemplo **x** como os k exemplos vizinhos “mais próximos” de **x**.
- Contar a quantidade $q(c_i)$ de exemplos na k-vizinhança que pertencem à classe c_i.
- Produzir uma estimativa de $\Pr(c_i \mid \mathbf{x})$:
- Classificar **x** como pertencente à classe mais prov

$$\Pr(c_i \mid \mathbf{x}) \approx \frac{q(c_i)}{k}$$

:

$$\mathrm{c}(\mathbf{x}) := \arg\max_{c_i \in C} \Pr(c_i \mid \mathbf{x})$$

Cálculo da proximidade

- Há diversas métricas possíveis para cálculo da proximidade.
- Variáveis contínuas:
  - Distância de Manhattan
  - Distância euclidiana
  - Similaridade por cosseno
- Variáveis discretas:
  - Hamming
  - Value distance measure

Distâncias *Manhattan* e *Euclidiana*

$$L_1(\mathbf{u},\mathbf{v}) = \|\mathbf{u}-\mathbf{v}\|_1 = \sum_{i=1}^{n} |u_i - v_i|$$

$$L_2(\mathbf{u},\mathbf{v}) = = \|\mathbf{u}-\mathbf{v}\|_2 =$$

$$= \sqrt{(u_1-v_1)^2 + (u_2-v_2)^2 + \cdots + (u_n-v_n)^2} =$$

$$= \sqrt{\sum_{i=1}^{n} (u_i - v_i)^2}$$

Similaridade por cosseno

$$\cos(\theta) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\|\|\mathbf{v}\|} = \frac{\sum_{i=1}^{n} u_i v_i}{\sqrt{\sum_{i=1}^{n} u_i^2}\sqrt{\sum_{i=1}^{n} v_i^2}}$$

Valor de k (hiperparâmetro)

- Valores de k muito pequenos podem levar ao *overfitting*…
  - e.g., definir $k = 1$ é uma estratégia sujeita a erros, devido a eventual ruído na categoria do vizinho.
- O valor de k pode ser determinado por *model selection*.

Avoiding overfitting in kNN

- Alternatives to avoid overfitting:
  - Model selection (measure predictive performance on validation data for increasing values of k);
  - Replace a bunch of points by their prototypes;
  - Remove *outliers*, i.e., remove data points for which all (or at least the majority) of the neighbor voronoi cells are from a different class.

kNN (classificação) - treinamento

- Não há!
- O método é preguiçoso (*lazy learning*): a fase de treinamento consiste apenas em armazenar as representações dos exemplos.
- O método não produz explicitamente um modelo ou hipótese.

kNN (classificação): pseudocódigo

- **Entrada**:
  - Coleção de exemplos rotulados X; ***x*** a ser categorizado.
- **Saída**:
  - Classe majoritária para ***x***.
- **Algoritmo**:
  - Para cada $\boldsymbol{x}^{(i)} \in$ X
    - $dist_i$ ← Similaridade($\boldsymbol{x}$, $\boldsymbol{x}^{(i)}$)
  - Ordene exemplos em X por valores decrescentes de $dist_i$
  - Seja X_k o conjunto dos primeiros k exemplos em Sort(X)
  - Retorne a classe majoritária em X_k.

kNN (regressão): pseudocódigo

- **Entrada**:
  - Coleção de exemplos rotulados X; ***x*** a ser categorizado.
- **Saída**:
  - Classe majoritária para ***x***.
- **Algoritmo**:
  - Para cada $\boldsymbol{x}^{(i)} \in$ X
    - $dist_i$ ← Similaridade($\boldsymbol{x}$, $\boldsymbol{x}^{(i)}$)
  - Ordene exemplos em X por valores decrescentes de $dist_i$
  - Seja X_k o conjunto dos primeiros *k* exemplos em Sort(X)
  - Retorne a média da variável dependente em X_k.

Distance-weighted kNN

- Variante do kNN que considera que há um peso associado a cada vizinho.

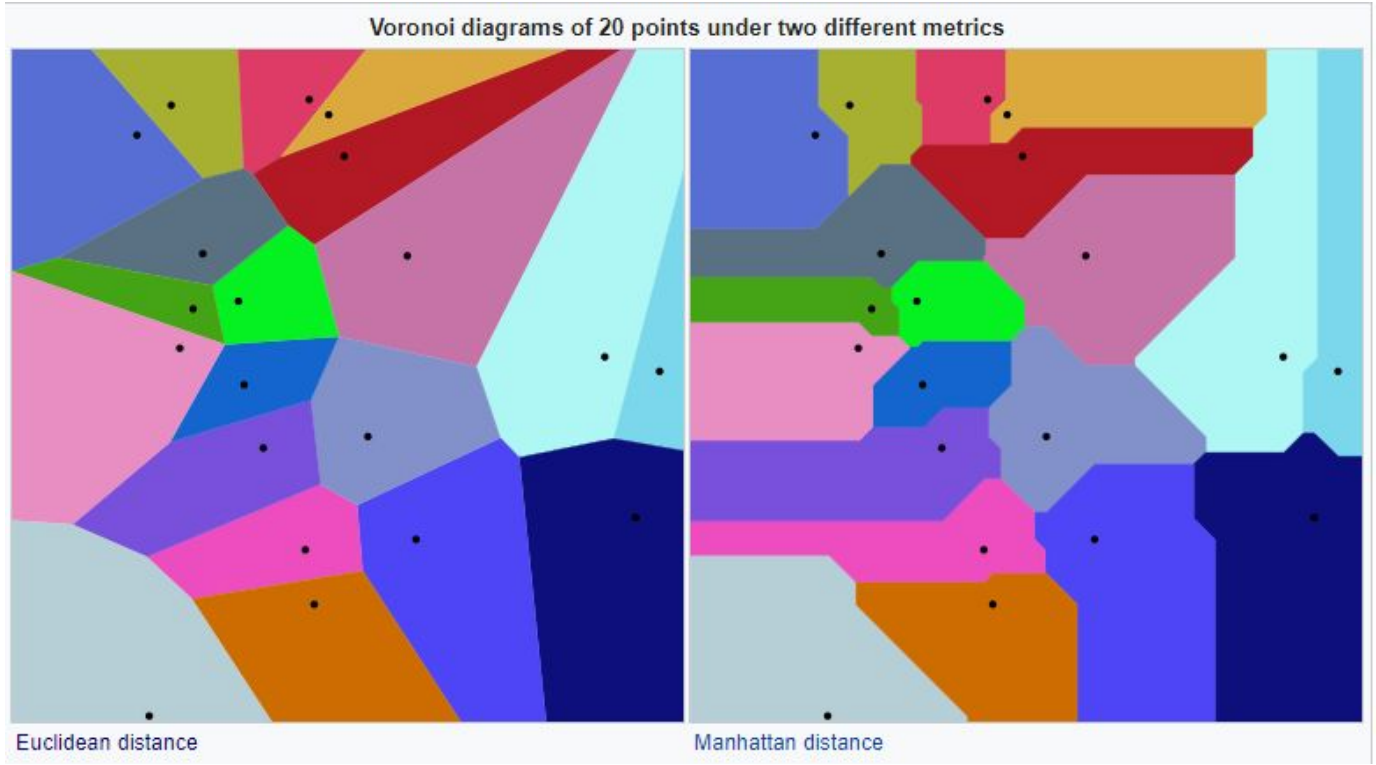
$$\hat{f}(x_q) \leftarrow \frac{\sum_{i=1}^{k} w_i f(x_i)}{\sum_{i=1}^{k} w_i}$$

$$w_i \equiv \frac{1}{d(x_q, x_i)^2}$$

- Nessa variante, faz sentido usar todos os exemplos de treinamento como vizinhos.

Maldição da Dimensionalidade e Normalização de Dados

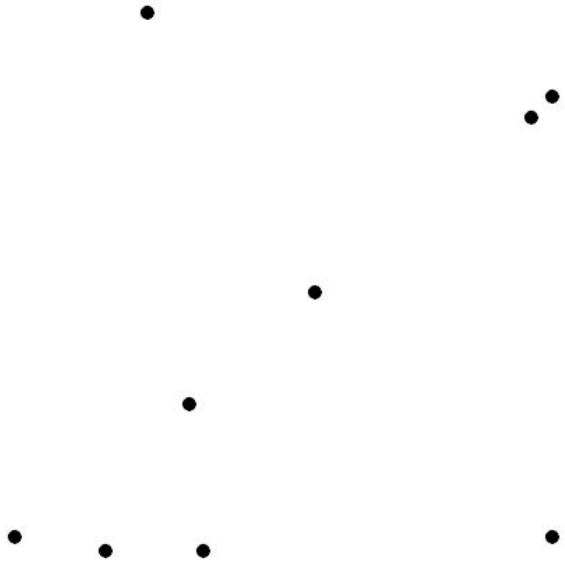
Decision boundary for kNN



Source: https://en.wikipedia.org/wiki/Voronoi_diagram

Decision boundary for kNN

- A Voronoi tessellation emerging by radial growth from examples outward.



Source: https://en.wikipedia.org/wiki/Voronoi_diagram