

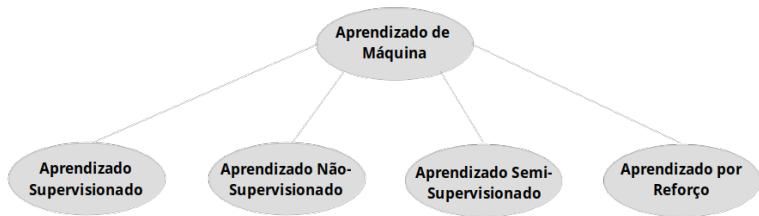
K-Vizinhos Mais Próximos

Prof. Jefferson T. Oliva

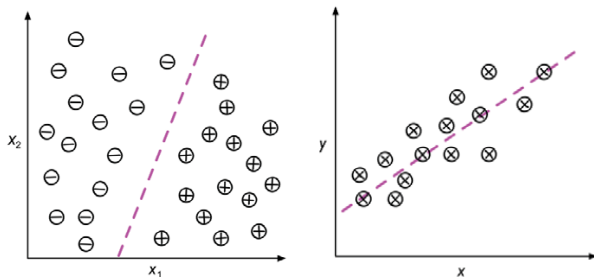
Aprendizado de Máquina e Reconhecimento de Padrões (AM28CP)
Engenharia de Computação
Departamento Acadêmico de Informática (Dainf)
Universidade Tecnológica Federal do Paraná (UTFPR)
Campus Pato Branco

- K-Vizinhos Mais Próximos

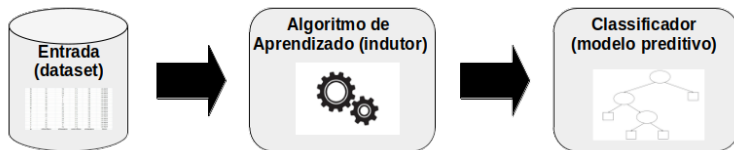
- Hierarquia do aprendizado de máquina



- Aprendizado supervisionado: classificação x regressão



- Classificador



- Exemplos de métodos de classificação
 - K-vizinhos mais próximos
 - Árvores de decisão
 - Naïve Bayes
 - Redes neurais artificiais
 - Máquinas de vetores de suporte
 - ...

K-Vizinhos Mais Próximos

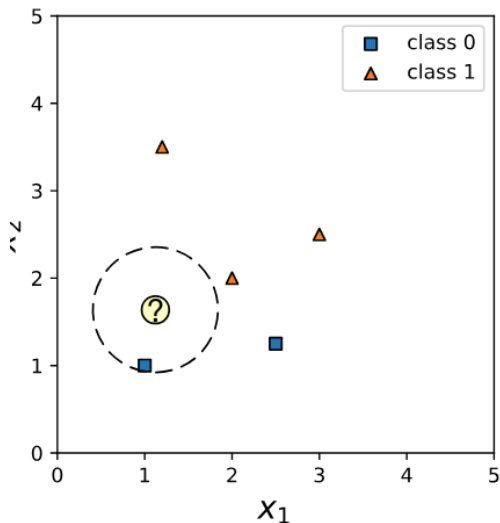
- *K-nearest neighbors* (KNN)
- É um dos algoritmos de classificação clássicos e um dos mais simples
- Classifica entrada de acordo com os exemplos de treinamento que estão mais próximos no espaço de características
 - O valor de K determina a quantidade de vizinhos mais próximos, conforme uma medida de distância, a serem analisados
 - A classe predominante na k vizinhança determina o rótulo do exemplo a ser classificado
 - Método não paramétrico
 - Aprendizado "preguiçoso"

K-Vizinhos Mais Próximos

- Para o uso do KNN é necessário
 - Uma base de dados de treinamento
 - Definir o valor de k para determinar a quantidade de vizinhos mais próximos que serão analisados no algoritmo
 - Definição de uma métrica para o cálculo de distância entre os exemplos de treinamento

- Passo-a-passo do algoritmo KNN para a classificação de um novo exemplo:
 - ① Calcular a distância entre a entrada e os exemplos de treinamento
 - ② Identificar os k vizinhos mais próximos, ou seja, aqueles em foram obtidos os menores valores de distância
 - ③ Classificação do exemplo de acordo com a classe majoritária (predominante) dos k vizinhos mais próximos
 - Caso $k = 1$ (*1-nearest-neighbor* ou 1-vizinho mais próximo), a classe será a mesma do exemplo em que foi obtido o menor valor de distância

- Exemplo 1NN ($k = 1$)



- Pseudo-código do algoritmo 1NN para a classificação de um novo exemplo (x^e)

```
ponto_mais_proximo := None  
menor_distancia =  $\infty$ 
```

```
Para i = 1 até n faça:  
    atual_distancia :=  $d(x^i, x^e)$ 
```

```
    Se atual_distancia < menor_distancia:  
        ponto_mais_proximo :=  $x^i$   
        menor_distancia := atual_distancia
```

```
retorne ponto_mais_proximo
```

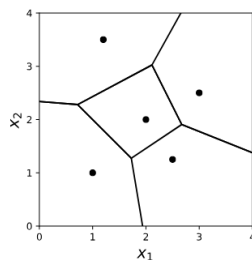
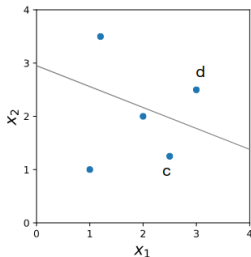
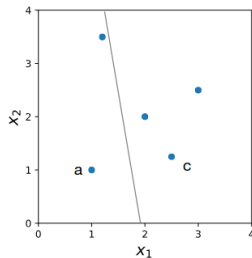
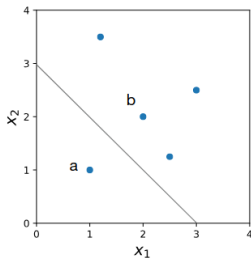
- Medida de distância comumente utilizada: Euclidiana

$$d(x^a, x^b) = \sqrt{\sum_{i=1}^m (x_i^a - x_i^b)^2}$$

onde m é a quantidade de atributos (características)

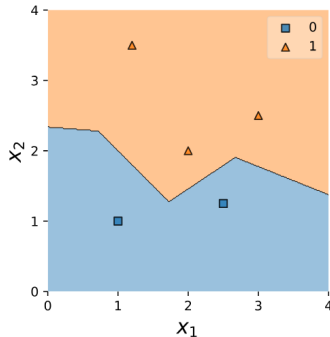
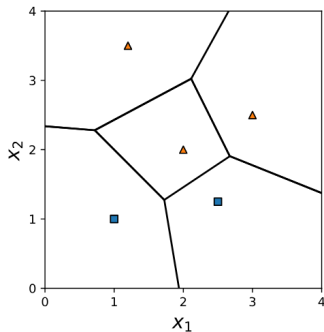
K-Vizinhos Mais Próximos

- Exemplo de fronteira de decisão do 1NN



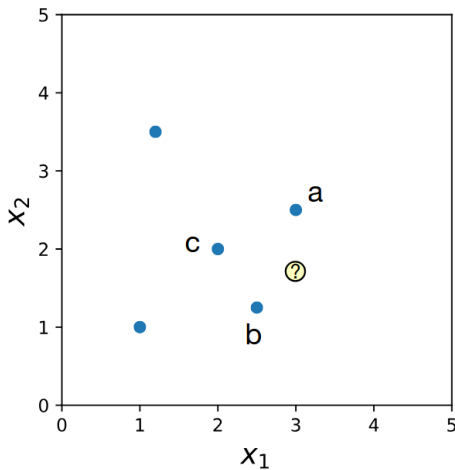
K-Vizinhos Mais Próximos

- Exemplo de fronteira de decisão do 1NN



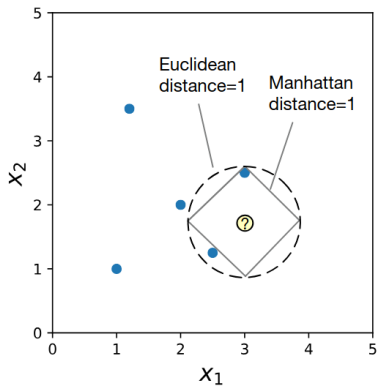
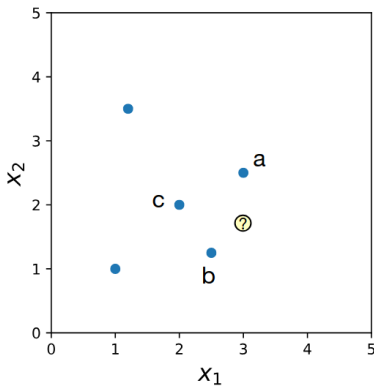
K-Vizinhos Mais Próximos

- Qual ponto é o mais próximo?



K-Vizinhos Mais Próximos

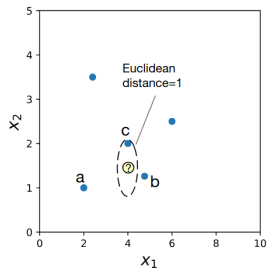
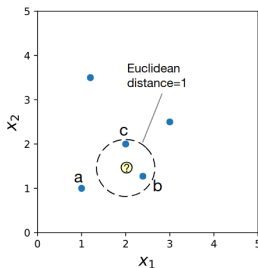
- Qual ponto é o mais próximo?
 - Res.: depende da medida de distância!



- Medidas de distância
 - Euclideana
 - Manhattan (*cityblock*)
 - Distância de Minkowski
 - Similaridade do cosseno
 - ...

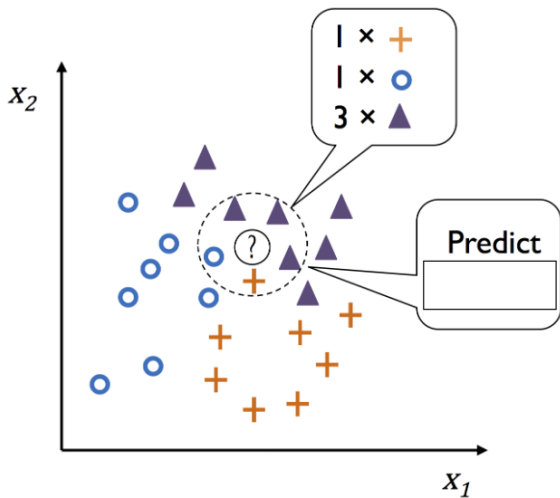
K-Vizinhos Mais Próximos

- Alguns atributos podem ter valores discrepantes em comparação com os outros, o que pode afetar negativamente o desempenho do classificador
 - Por exemplo, uma pessoa pode variar ter até 2,1 m de altura enquanto pode pesar mais que 100 kg
- Uma solução para esse tipo de problema seria fazer a re-escala de características por alguma técnica, como padronização, normalização Z, máximo absoluto, entre outras

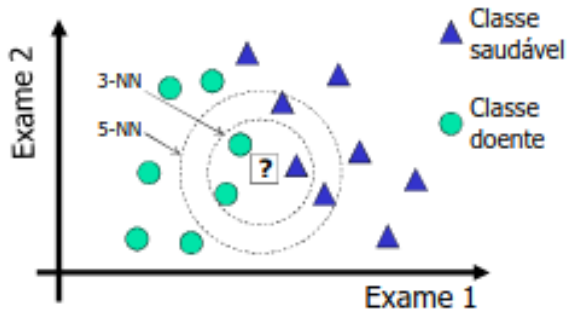


K-Vizinhos Mais Próximos

- KNN



- KNN



- KNN para classificação
 - Assumindo que os k-vizinhos ($D_k \subseteq D$) foram encontrados para um ponto x^q e a classe $y \in \{1, \dots, t\}$

$$D_k = \{\langle x^1, f(x^1) \rangle, \dots, \langle x^k, f(x^k) \rangle\}$$

- Temos a seguinte hipótese:

$$h^q = \arg \max_{y \in \{1, \dots, t\}} \sum_{i=1}^k \delta(y, f(x^i))$$

onde, $\delta(a, b) = 1$, se $a = b$; ou $\delta(a, b) = 0$, caso contrário

$$h(x^q) = \text{mode}(f(x^1), \dots, f(x^k))$$

- KNN para regressão
 - O conceito é o mesmo para a classificação:
 - ① Encontrar os k-vizinhos mais próximos
 - ② Fazer a predição com base no rótulo dos k-vizinhos mais próximos
 - A diferença está na função alvo, que consiste na média do rótulo dos k-vizinhos mais próximos

$$h(x^q) = \frac{1}{k} \sum_{i=1}^k f(x^i)$$

- Quantos vizinhos?
 - K muito grande
 - Custo computacional mais elevado
 - Predição tendenciosa para a classe majoritária
 - Vizinhos podem ser muito diferentes
 - K muito pequeno
 - Pode não utilizar quantidade suficiente de informação
 - A previsão é sensível a ruídos
- Uma abordagem que pode ajudar na escolha seria o teste de vários valores diferentes de k
 - Pode ser muito custoso dependendo da quantidade e dos valores de k

- Como o KNN utiliza medidas de distância (Euclidiana) no processo de classificação, caso houver atributos simbólicos, deve ser feito algum método de pré-processamento
 - Discretização
 - Normalização
 - Ponderação
 - ...

- No exemplo abaixo podemos "binarizar" os atributos (e.g. sim = 1, não = 0)

Febre	Enjôo	Manchas	Dores	Diagnóstico
sim	sim	pequenas	sim	doente
não	não	grandes	não	saudável
sim	sim	pequenas	não	saudável
sim	não	grandes	sim	doente
sim	não	pequenas	sim	saudável
não	não	grandes	sim	doente

- Como o algoritmo KNN não tem uma etapa explícita de treinamento e posterga toda computação até predição para cada exemplo, o mesmo é denominado como algoritmo "preguiçoso"
 - Podemos dizer que conjunto de treinamento é o próprio modelo
- KNN é categorizado como um método baseado em instância (ou memória)
- O k-vizinhos mais próximos é um método paramétrico e o seu respectivo modelo é categorizado como discriminativo
- O algoritmo é sucessível à maldição da dimensionalidade

- Complexidade do KNN (implementação ingênua): $O(n \times m)$, onde n é o tamanho do dataset (quantidade de exemplos) e m é o número de atributos
 - Se assumirmos que $n \gg m$, então podemos dizer que a complexidade do KNN é $O(n)$

$\mathcal{D}_k := \{\}$

while $|\mathcal{D}_k| < k$:

- `closest_distance` := ∞
- for $i = 1, \dots, n$, $\forall i \notin \mathcal{D}_k$:
 - `current_distance` := $d(\mathbf{x}^{[i]}, \mathbf{x}^{[q]})$
 - if `current_distance` < `closest_distance`:
 - * `closest_distance` := `current_distance`
 - * `closest_point` := $\mathbf{x}^{[i]}$
- add `closest_point` to \mathcal{D}_k

- Complexidade do KNN utilizando fila de prioridade: $O(k \log n)$
 - KD-tree
 - Ball-tree
- Hyperparâmetros do algoritmo KNN
 - Valor de k
 - Escala de características
 - Medida de distância
 - Ponderação da medida de distância



BISHOP, C. M.

Pattern Recognition and Machine Learning.

Springer, 2006.



DE CARVALHO, A. P. L. F.

Métodos Baseados em Distância. Aprendizado de Máquina.

Slides. Ciência de Computação e Matemática Computacional.

ICMC/USP, 2015.



DUDA, Richard O.; HART, Peter E.; STORK, David G.

Pattern classification.

2nd ed. New York, NY: J. Wiley & Sons, 2001.



RASCHKA, S.; MIRJALILI, V.

Python Machine Learning.

Packt, 2017.