



Objectivos

- Adquirir a noção de aprendizagem baseada em instâncias
- Descrever os elementos fundamentais do algoritmo *k Vizinhos Mais Próximos* (kNN)



•Sumário:

- Aprendizagem baseada em instâncias
 - Introdução
 - Algoritmo kNN



Aprendizagem baseada em instâncias (1/5)

- As técnicas de aprendizagem baseada em instâncias permitem realizar tarefas de classificação sem a necessidade de criar estruturas abstractas de generalização
- Neste caso a aprendizagem consiste apenas em armazenar o conjunto de dados de treino apresentado



Aprendizagem baseada em instâncias (2/5)

- Ao classificar um novo exemplo, se recupera da memória um conjunto de instâncias similares e estas são utilizadas como base da classificação
- A generalização para além dos exemplos de treino se realiza somente no momento da classificação de uma nova instancia



Aprendizagem baseada em instâncias (3/5)

- Por não criar um modelo de aprendizagem explícito e relegar a aprendizagem ao momento da classificação são denominadas técnicas de aprendizagem ociosas (*lazy learning*)



Aprendizagem baseada em instâncias (4/5)

- A recuperação das instâncias da memória é feita com base numa função da distância
- Na maioria dos casos se utiliza a distância euclidiana

$$d(\mathbf{x}_1, \mathbf{x}_2) = \sqrt{\sum_{i=1}^n (a_{1i} - a_{2i})^2}$$

- Outras alternativas
 - Distância Manhattan ou city-block
 - Distâncias calculadas com potências superiores a 2...



Aprendizagem baseada em instâncias (5/5)

- O método mais básico incluído nesta classe é o denominado *k Vizinhos Mais Próximos* (*k Nearest Neighbours, kNN*)



Algoritmo kNN (1/3)

- O algoritmo assume que todas as instâncias correspondem a pontos num espaço n-dimensional R^n
- Considerando uma função objectivo discreta (problema de classificação), o algoritmo retorna a classe mais comum entre os k exemplos de treino mais próximos da instância de teste
- Para um problema com duas classes o valor de k deve ser ímpar e, em geral, não deve ser múltiplo do número de classes

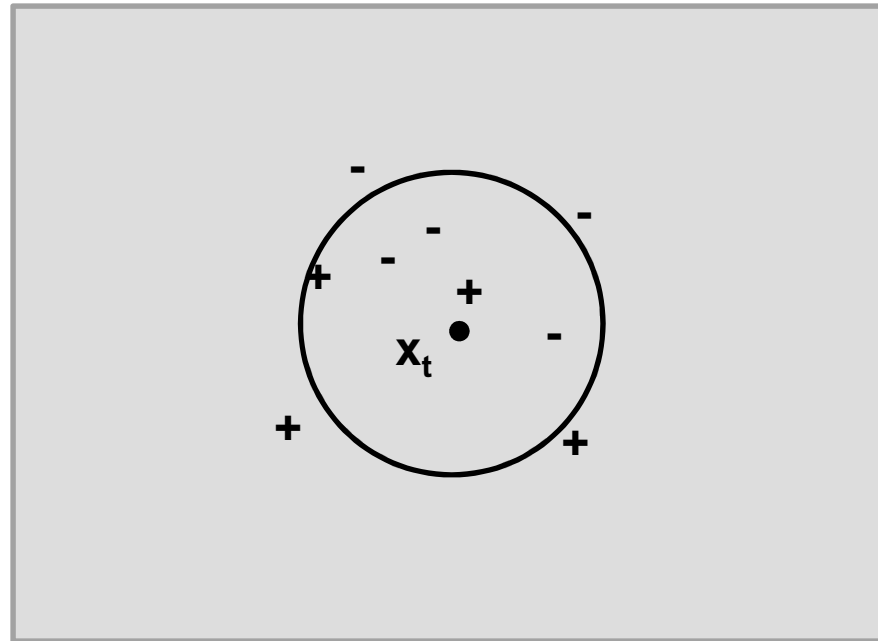


Algoritmo kNN (2/3)

- Algoritmo de treino
 - Para cada exemplo de treino $\langle \mathbf{x}, c \rangle$
 - Adicionar o exemplo à lista *exemplos_de_treino*
- Algoritmo de teste
 - Dada uma instância \mathbf{x}_t a ser classificada
 - Sejam $\mathbf{x}_1, \dots, \mathbf{x}_k$ as instâncias da lista *exemplos_de_treino* mais próximas de \mathbf{x}_t
 - Retornar a classe mais comum entre as k instâncias

$$c(x_t) = \arg \max_{c \in C} \sum_{i=1}^k \delta(c, c(x_i)) \quad \delta(a, b) = \begin{cases} 1 & \text{se } a = b \\ 0 & \text{se } a \neq b \end{cases}$$

Algoritmo kNN (3/3)





Atributos nominais

- A formulação apresentada assume implicitamente que os atributos são numéricos
- Para atributos ordinais
 - A diferença entre dois valores distintos se considera igual a 1
 - Se for o mesmo valor a diferença é igual a 0



Valores omissos

- Caso existam valores omissos, a diferença entre os valores dos atributos é tomada de forma a que seja a maior possível
- Para atributos nominais
 - Se um ou ambos valores são omissos a diferença entre os mesmos se considera igual a 1
- Para atributos ordinais
 - Se ambos valores forem omissos, a diferença é igual a 1
 - Se apenas um dos valores for omissos, a diferença se toma como o maior valor entre o valor de atributo presente e um menos o referido valor $a_{1i} - a_{2i} = \max(a_{1i}; 1 - a_{1i})$



Algoritmo kNN com distâncias ponderadas (1/2)

- Uma variante do método consiste em sopesar a contribuição dos k vizinhos mais próximos na função de decisão (votação) de acordo à sua proximidade à instância a ser classificada
- Atribui-se maior peso aos vizinhos mais próximos da instância a ser classificada



Algoritmo kNN com distâncias ponderadas (2/2)

- Algoritmo semelhante. Ao determinar a classe

$$c(x_t) = \arg \max_{c \in C} \sum_{i=1}^k w_i \delta(c, c(x_i)) \quad w_i = \frac{1}{1 + d(x_t, x_i)}$$



Questões práticas a considerar...

- Normalização dos dados
- Atributos irrelevantes
- Custo computacional



Normalização

- Diferentes atributos geralmente são medidos em distintas escalas
- Ao calcular as distâncias o efeito de alguns atributos pode predominar sobre outros
- Consequentemente, é usual a normalização dos valores dos atributos de forma a que estejam compreendidos entre 0 e 1



Atributos irrelevantes

- A distância entre instâncias é calculada com base em todos os atributos das instâncias
- Em consequência, caso existam atributos irrelevantes, a distância entre vizinhos pode ser dominada por estes e conduzir a resultados enganadores (problema conhecido como *maldição da dimensionalidade* – *curse of dimensionality*)



Custo computacional

- O custo computacional associado a classificação de novas instâncias pode ser alto
- Quase todo o processo de cálculo é levado a cabo em tempo de classificação em vez de ser levado a cabo ao observar os exemplos de treino por primeira vez
- Técnicas de indexação dos exemplos de treino podem ser significativas na redução do cálculo necessário para a classificação de novos exemplos



Exercício

- Considere o conjunto de 6 exemplos de pontos no plano real e as suas correspondentes classificações. Dada uma nova instância a classificar, $(4, 4)$, qual seria a classificação atribuída:
 - Utilizando o algoritmo kNN, com $k = 1$ e $k = 3$?
 - Que resultado seria obtido utilizando a ponderação das distâncias?



Exercício

Instância	X	Y	Classe	$d(I_t, I_i)$
1	6	8	-	4,47
2	2	6	+	2,83
3	5	6	-	2,24
4	4	5	+	1,00
5	4	2	-	2,00
6	2	1	-	3,61



Tarefa

- Leitura da semana *lazy vs eager learning*
epígrafe 8.6, pg. 244 – 245 Mitchell



Bibliografia

- Mitchell, pg. 230 – 236
- Witten, pg. Pg. 131 – 132, 137 – 138
- Borrajo Millán