Aprendizado Baseado em Instâncias

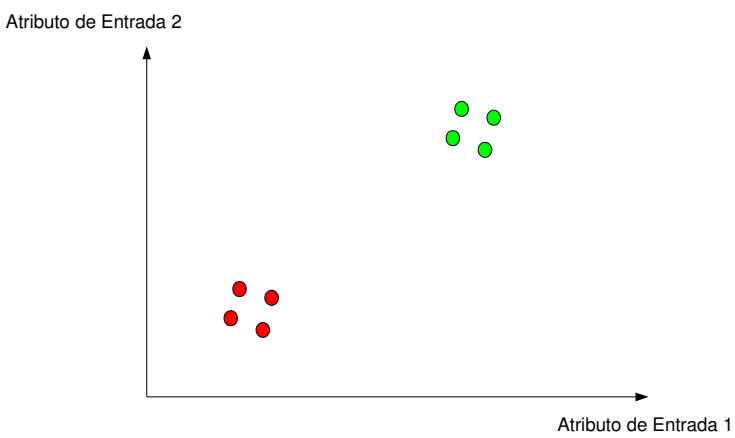
Universidade de São Paulo
Instituto de Ciências Matemáticas e de Computação
Departamento de Ciências de Computação
Rodrigo Fernandes de Mello
mello@icmc.usp.br

Aprendizado Baseado em Instâncias

- Classificador que armazena exemplos de treinamento:
  - Exemplos são também denominados instâncias
- Generaliza informações com base em exemplos de treinamento:
  - Para inferir a classe de novas instâncias (ou instâncias de consulta)
  - Assim cada vez que uma instância é recebida, computa-se uma função objetivo com base no conhecimento oferecido pela base de exemplos de treinamento
- Estimam a classe da nova instância com base em comportamentos locais
- É uma técnica incremental

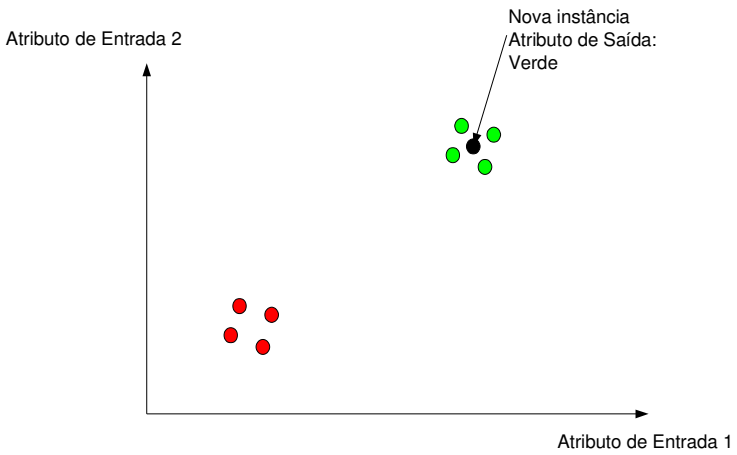
Aprendizado Baseado em Instâncias

- Exemplo (em que a classe, ou atributo de saída, é dada pelas cores):



Aprendizado Baseado em Instâncias

- Exemplo:

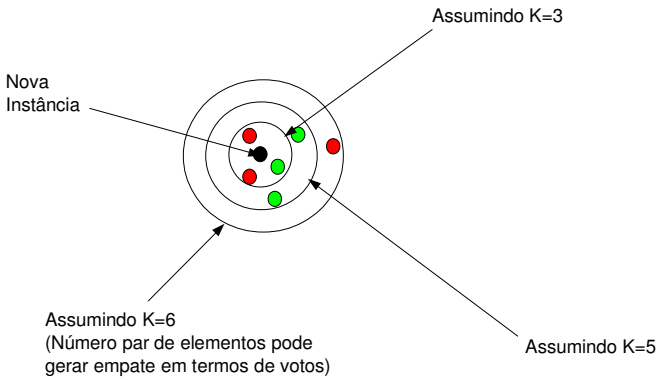


Aprendizado Baseado em Instâncias

- **K-Nearest Neighbors**
  - Técnica mais comum de IBL (Instance-Based Learning)
  - Calcula-se a distância Euclidiana da nova instância em função de cada instância na base de conhecimento
  - Seleciona-se as K instâncias de treinamento mais próximas
  - Define-se o atributo de saída na forma:
    - Discreta:
      - Por exemplo, por maior número de votos, decide-se se a cor é vermelha ou verde
    - Contínua:
      - Define-se o atributo de saída pela ponderação das saídas das K instâncias mais próximas

Aprendizado Baseado em Instâncias

- **K-Nearest Neighbors**
  - Forma Discreta:



Aprendizado Baseado em Instâncias

- **K-Nearest Neighbors**
  - Forma Contínua:

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

  - Considera uma média do atributo de saída de seus k vizinhos mais próximos

Aprendizado Baseado em Instâncias

- **Distance-Weighted Nearest Neighbors**
  - Um refinamento ou variação do KNN
  - Considera uma ponderação da influência de cada vizinho em função de sua distância à nova instância
- Para o caso **discreto**:

$$\hat{f}(x_q) \leftarrow \underset{v \in V}{\operatorname{argmax}} \sum_{i=1}^{k} w_i \delta(v, f(x_i))$$

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

  - Se nova instância é exatamente igual a uma instância de treinamento, então d(.) = 0, logo deve-se prever essa situação e o algoritmo deve retornar $f(x_i)$

Aprendizado Baseado em Instâncias

- **Distance-Weighted Nearest Neighbors**
- Para o caso **contínuo**:

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

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

  - Se nova instância é exatamente igual a uma instância de treinamento, então d(.) = 0, logo deve-se prever essa situação e o algoritmo deve retornar $f(x_i)$

Aprendizado Baseado em Instâncias

- **Distance-Weighted Nearest Neighbors**
  - Essas duas variantes de KNN consideram os k vizinhos mais próximos
    - No entanto, para o Distance-Weighted, podemos considerar todas as instâncias de treinamento pois as mais distantes terão pouca ou nenhuma influência na instância de consulta
    - A desvantagem dessa abordagem é que nosso classificador irá executar de maneira mais lenta
  - Se somente os k mais próximos forem considerados:
    - Algoritmo é denominado local
  - Se todas as instâncias de treinamento forem consideradas:
    - Algoritmo é denominado global

Aprendizado Baseado em Instâncias

- Questões importantes:
  - As duas versões do KNN sempre:
    - Computam a saída em função de todos atributos de entrada
    - Logo, se **atributos de entrada não forem bem definidos** o resultado pode ser ruim:
      - Por exemplo, instâncias são representadas por 10 atributos de entrada e 1 de saída
      - Mas somente 2 atributos de entrada são relevantes
      - Os 8 outros podem influenciar a saída, mas nem deveriam existir na base de aprendizado!
    - Uma abordagem para resolver esse problema é dar um peso para cada atributo:
      - Assim os mais relevantes serão mais considerados!
    - Outra é remover os atributos menos significativos

Aprendizado Baseado em Instâncias

- Função kernel
  - Função de distância utilizada para determinar o peso de cada exemplo de treinamento na saída da instância de consulta
  - Por exemplo, considere a função kernel K:

$$w_i = K(d(x_i, x_q))$$

  - K(.) pode ser linear, exponencial, etc.

Aprendizado Baseado em Instâncias

- Estime (Exemplos de uso de IBL):
  - O consumo de CPU de um processo com base em histórico
  - O consumo de memória de um processo com base em histórico
  - O volume de transações de um banco de dados com base em histórico
  - O volume de requisições de um Servidor Web com base em histórico
  - A carga de uma requisição web com base em histórico

Aprendizado Baseado em Instâncias

- Implementação
  - Vamos implementar com função kernel Radial

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

$$w_i = e^{-\frac{E(x_q, x_i)^2}{2 \cdot \sigma^2}}$$

  - Em que:
    - X_q é uma instância de consulta

Aprendizado Baseado em Instâncias

- **Case-Based Reasoning**:
  - O aprendizado é baseado em duas condições:
    - Decisões são emitidas assim que uma nova instância ou instância de consulta chega ao sistema
    - Decisões dependem de instâncias de treinamento similares
  - Nesses dois aspectos são similares às duas variações de KNN
    - No entanto não considera o caso contínuo

Aprendizado Baseado em Instâncias

- **Case-Based Reasoning**:
  - Abordando com exemplo:
    - Considere uma base de instâncias de treinamento que descrevem o funcionamento de uma peça mecânica por exemplo, o motor de um carro
    - Cada instância de treinamento é formada por uma descrição simbólica que conecta duas peças
      - Por exemplo:
        - Câmara de combustão X → Pistão Y
        - Mangueira de alimentação do radiador Z → Radiador T
      - Cada instância na base representa as conexões entre peças tal como em um grafo
        - Podem também apresentar um termo (aptidão) para tal conexão

Aprendizado Baseado em Instâncias

- **Case-Based Reasoning**:
  - Abordando com exemplo:
    - Assim, quando um projetista deseja criar um novo motor:
      - Entrada com partes das condições
      - Solicita maior aptidão
    - CBR
      - Faz um “match” das várias condições que satisfazem o conceito de entrada, oferecido pelo projetista, e pode prover diversas saídas e suas aptidões relativas
      - Tem alto custo computacional, pois considera a associação de regras armazenadas para montar um único sistema
        - A forma de combinação dessas regras difere do KNN que, em geral, faz uma soma ponderada

Aprendizado Baseado em Instâncias

- **Case-Based Reasoning**:
  - Abordando com exemplo:

$$A \xrightarrow{+} B$$

$$A \xrightarrow{+} x \xrightarrow{+} B$$

  - Cria-se uma série de grafos conectando as várias regras e mede-se o custo ou aptidão de cada grafo
    - Cada grafo funciona como uma solução candidata
  - Pode ser que não haja um grafo para representar o sistema todo
    - Então uma solução parcial é retornada

Exercícios

- Implemente IBL
  - Com Kernel Radial
- Execute para:
  - Classificação:
    - Iris
  - Predição:
    - Mapa Logístico

Referências

- Tom Mitchell, Machine Learning, 1997