Aprendizado baseado em instâncias

Roteiro da Apresentação

- O que é?
- Métodos
  - *k* vizinhos mais próximos
  - Regressão localmente ponderada
  - Redes RBF
  - Raciocínio baseado em casos
- Paralelo entre aprendizado preguiçoso e guloso
- Bibliografia

O que é? (1/2)

- Aprender consiste em **armazenar** as instâncias de treinamento
- Calcula a **distância** entre as instâncias de treinamento e a instância desconhecida
- **Avalia** o valor da função de classificação a partir dos valores das instâncias mais próximas

O que é? (2/2)

- Diferentes métodos possuem diferentes formas de:
  - Representar as instâncias de treinamento
  - Calcular a distância entre instâncias
  - Avaliar o valor da função de classificação

k vizinhos mais próximos

- Método mais antigo (1967) e difundido
- Instâncias são representadas por pontos num espaço *n* dimensional $\Re^n$
  - Uma instância x é representada por

    $$<a_1(x), a_2(x), a_3(x), ..., a_n(x)>$$

    Onde $a_r(x)$ representa o valor do r-ésimo atributo
  - E sua função de classificação por

    $$f(x)$$

k vizinhos mais próximos

- Distância entre as instâncias pode ser calculada pela distância euclidiana
  - Distância euclidiana entre x_i e x_j

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

  - Existem outras, por exemplo, *Manhattan*

k vizinhos mais próximos

- A função de classificação $\hat{f}$
  - Caso seja discreta, seu resultado é aquele que aparecer mais vezes entre os *k* vizinhos mais próximos

$$f : \Re^n \rightarrow V$$

  - Caso seja contínua, seu resultado é a média dos resultados dos *k* vizinhos mais próximos

$$f : \Re^n \rightarrow \Re$$

k vizinhos mais próximos

- Algoritmo
  - Treinamento
    - Para cada instância de treinamento *<x,f(x)>* adicione a instância na lista *instancias_treinamento*

k vizinhos mais próximos

- (Algoritmo cont.)Classificação
  - Para cada instância x_q a ser classificada
    - Chame de $x_1, x_2, \ldots x_k$ as *k* instâncias mais *próximas* de x_q na lista *instancias_treinamento*
    - Retorna
      - Caso discreto

$$\hat{f}(x_q) \leftarrow \arg\max_{v \in V} \sum_{i=1}^{k} \delta(v, f(x_i))$$

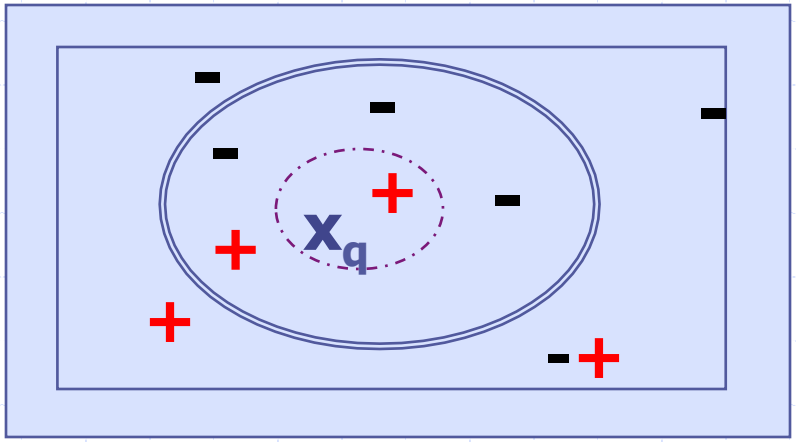
onde $\delta(a,b)$ é igual a 1 se $a = b$ e 0 se $a \neq b$

      - Caso contínuo

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

k vizinhos mais próximos

- Exemplo



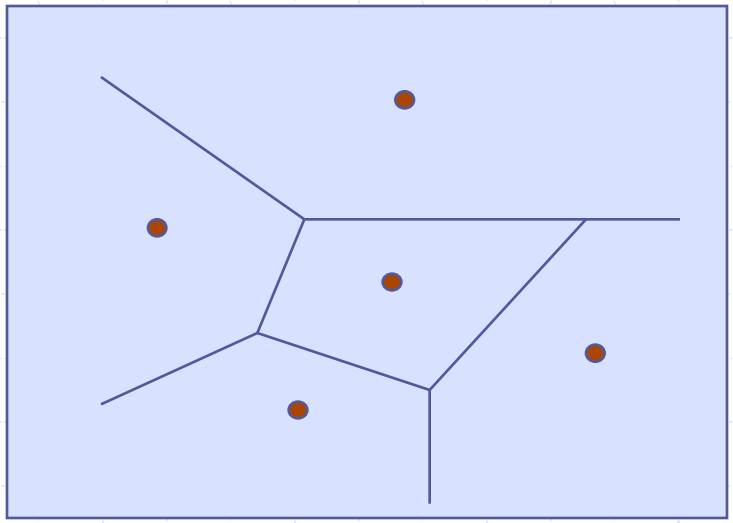
k = 1 classifica x_q como **+**

k = 5 classifica x_q como **-**

- Percebe-se que o *k* é determinante na classificação

k vizinhos mais próximos

- Diagrama de Voronoi
  - Define um poliedro convexo para cada instância de treinamento.
  - As instâncias dentro do poliedro são completamente classificados pela instância associada



Acima, $k = 1$

http://www.cs.cornell.edu/Info/People/chew/Delaunay.html

k vizinhos mais próximos

- Um refinamento óbvio é ponderar a contribuição de cada um dos k vizinhos de acordo com sua distância ao ponto de consulta x_q
  - Isso pode ser conseguido fazendo as seguintes alterações
    - Caso discreto

$$\hat{f}(x_q) \leftarrow \arg\max_{v \in V} \sum_{i=1}^{k} \omega_i \delta(v, f(x_i))$$

    - Caso contínuo

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

onde

$$\omega_i \equiv \frac{1}{d(x_i, x_q)}$$

k vizinhos mais próximos

- Problema da dimensionalidade
  - Para calcular a distância entre os pontos, o método utiliza ***todos*** os atributos da instância
  - Problema: quando alguns destes atributos não são importantes para a classificação
  - Soluções
    - Atribuir pesos ω_j aos atributos de maneira que minimize a taxa de erro de classificação
    - Usar a técnica de *validação cruzada* para automaticamente escolher os pesos
    - Eliminar atributos do espaço de instâncias

k vizinhos mais próximos

- Validação Cruzada
  - Serve para evitar overfitting e para averiguar robustez dos resultados
  - Algoritmo

1) Divide o conjunto de exemplos em dois sub-conjuntos: conjuntos de treinamento (TR) e de teste (TE)
2) Usa indução para gerar hipótese H sobre TR
3) Mede percentagem de erro de H aplicada à TE
4) Repete passos 1-3 com diferentes tamanhos de TE e TR, e tendo elemento escolhidos aleatoriamente

k vizinhos mais próximos

- Observações
  - Inferência indutiva
  - Efetiva para muitos problemas práticos
  - Robusto quanto a ruído nas instâncias de treinamento.
    - Utilizando a média ponderada isola ainda mais o efeito do ruído.

Regressão localmente ponderada

- Generalização de vizinhos mais próximos
- Constrói uma aproximação explicita de uma função $f(x_q)$ em uma região próxima de x_q

Regressão localmente ponderada

- Localmente
  - A aproximação é definida na vizinhança de x_q e servirá exclusivamente para sua classificação
- Ponderada
  - A contribuição de cada instância é ponderada pela distância entre estas e x_q
- Regressão
  - Designa o problema de encontrar uma função de aproximação

Regressão localmente ponderada

- Descrição
  - Construir uma aproximação $\hat{f}(x)$ que ajuste os valores das instâncias de treinamento na vizinhança de x_q.
  - A aproximação é então usada para calcular o valor ponto x_q.
  - A descrição de $\hat{f}$ é apagada, pois a função de aproximação será construída para cada instância a ser consultada

Regressão localmente ponderada

- Função de aproximação mais comum

$$\hat{f}(x) = \omega_0 + \omega_1 a_1(x) + \ldots + \omega_n a_n(x)$$

- Escolher ω_i que minimiza a soma dos quadrados dos erros em relação ao conjunto de treinamento D

$$\mathrm{E}(x_q) = \frac{1}{2} \sum_{x \in D} \left(f(x) - \hat{f}(x) \right)^2$$

Regressão localmente ponderada

- Existem diferentes propostas para minimizar o erro(1/3)
  - Erro quadrático sobre os k-vizinhos mais próximos

$$E(x_q) = \frac{1}{2} \sum_{x \in \text{ k vizinhos mais próximos de x}_q} \left(f(x) - \hat{f}(x) \right)^2$$

Regressão localmente ponderada

- Existem diferentes propostas para minimizar o erro(2/3)
  - Erro quadrático ponderado em D

$$E(x_q) = \frac{1}{2} \sum_{x \in D} \left(f(x) - \hat{f}(x) \right)^2 K(d(x_q, x))$$

  - Onde $K(d(x_q, x))$ é uma função que penaliza grandes distâncias entre os pontos

Regressão localmente ponderada

- Existem diferentes propostas para minimizar o erro(3/3)
  - Combinação das duas anteriores

$$E(x_q) = \frac{1}{2} \sum_{x \in \text{ k vizinhos mais próximos de } x_q} \left(f(x) - \hat{f}(x)\right)^2 K(d(x_q, x))$$

Regressão localmente ponderada

- Observações
  - Várias funções para cálculo da distância
    - Distância euclidiana bastante usada
  - Várias funções de aproximação
    - Constante, linear e quadrática
    - Funções mais complexas são evitadas
      - Custo de ajustamento
      - As funções mais simples fornecem aproximações boas sobre uma região suficientemente pequena do espaço de instâncias

Redes RBF (Radial Basis Function)

- Abordagem que envolve regressão ponderada por distância e redes neurais artificiais
- A aproximação de $\hat{f}(x)$ é representada por uma combinação de aproximações locais

Redes Neurais Artificiais (RNA)

- Paradigma Conexionista
  - Modelo inspirado no cérebro humano
    - Várias unidades de processamento (“neurônio”)
    - Grande número de conexões entre elas (“sinapses”)
  - Abordagem robusta para aproximar funções de valores contínuos, discretos e vetores

Redes Neurais Artificiais (RNA)

- Estrutura
  - Cada neurônio (ou nó) possui
    - Conjunto de entrada X_i que podem ser um exemplo de entrada ou saídas de outros nós
    - Um conjunto de pesos w_i associados a cada entrada
    - Uma função de ativação F_i, que indica quando o nó passará o sinal a diante
    - Uma função de saída y, que é o valor que a nó transmitirá

Redes Neurais Artificiais (RNA)

- Redes com uma camada escondida aproximam qualquer função linearmente dependente. Com duas, qualquer função
- *Backpropagation*
  - Algoritmo usado para treinamento de redes com duas ou mais camadas intermediárias que atualiza os pesos de acordo a diferença entre o resultado obtido e o esperado

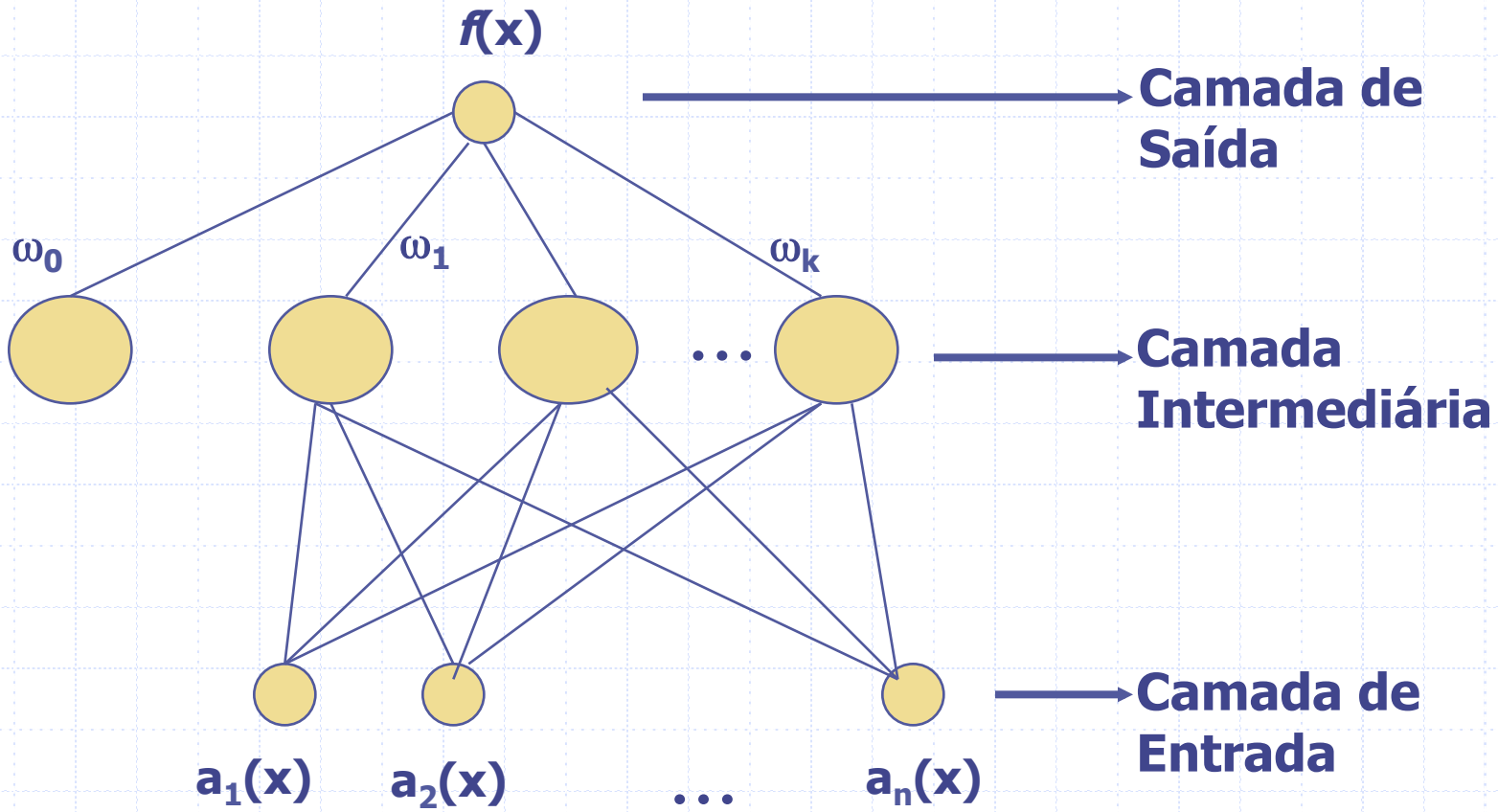
Redes RBF

- A função a aproximar é da forma

$$\hat{f}(x) = \omega_0 + \sum_{u=1}^{k} \omega_u K_u(d(x_u, x))$$

  - Onde k é o parâmetro que especifica a quantidade de *kernels* (nós da camada intermediária)
  - A equação acima pode ser vista como descrevendo duas camadas da rede.
    - A primeira computa os valores de k
    - A segunda computa uma combinação linear dos valores da primeira camada

Redes RBF



a_i i=1,..n:atributos descrevendo uma instância x

Redes RBF

- Função de kernel (ou de base radial) $K_u(d(x_u,x))$
  - Funções cujo valor muda conforme à distância entre um ponto x e um ponto central x_u
  - A mais comum é a função gaussiana centrada no ponto x_u com variância σ^2_u

$$K_u(d(x_u,x)) = e^{-\frac{1}{2\sigma_u^2}d^2(x_u,x)}$$

Redes RBF

- Treinamento
  - O valor de k é determinado pelo usuário
  - cada função *kernel* é definida pelos valores x_u e σ_u^2
  - os pesos ω_u são alterados para maximizar o ajuste da rede aos dados de treinamento, usando o critério de minimização do erro global

Redes RBF

- Escolha de *k* (1/2)
  - Abordagem 1
    - Alocar uma função gaussiana para cada exemplo de treinamento.
    - Cada kernel pode ter a mesma largura (variância σ^2).
    - As redes RBF aprendem uma aproximação global para *f* (x) onde o valor poder ser influenciado apenas pela vizinhança de x

Redes RBF

- Escolha de *k* (2/2)
  - Abordagem 2
    - Escolher um conjunto de funções de kernel menor que o conjunto de treinamento. Pode se mais eficiente que a abordagem 1.
    - Construir grupos do conjunto de treinamento, definir protótipos para os grupos e adicionar uma função de kernel para cada grupo.
    - O algoritmo EM fornece uma estimativa para as médias e as variâncias para os grupos.

Redes RBF

- Algoritmo EM (*Expactation-Maximization*)
  - Muito usado para aprender na presença de variáveis não observadas
  - A média é escolhida para maximizar a probabilidade de observar as instâncias x_i dadas as *k* médias estimadas
  - Inicia com uma hipótese arbitrária
  - Repetidamente calcula os valores esperados das variáveis escondidas (assumindo que a hipótese atual está correta)
  - Recalcula a hipótese de máxima probabilidade (assumindo que as variáveis escondidas tenham o valor esperado calculados no primeiro passo).
  - Converge para um máximo local.

Redes RBF

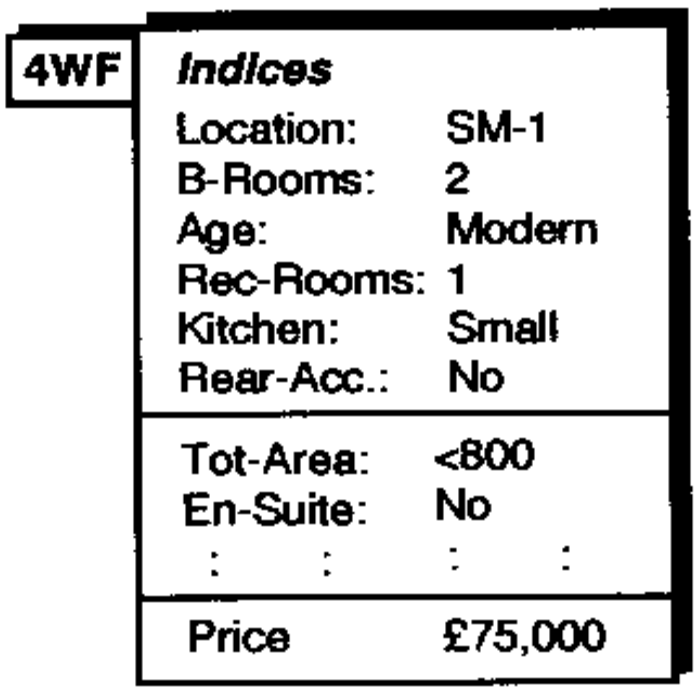
- Observações
  - Usada, por exemplo na classificação supervisionada de imagens
  - A aproximação global de *f(x)* é obtida antes que um novo exemplo seja apresentado (aprendizado guloso)
  - Treinamento mais eficiente que das redes utilizando *backpropagation*

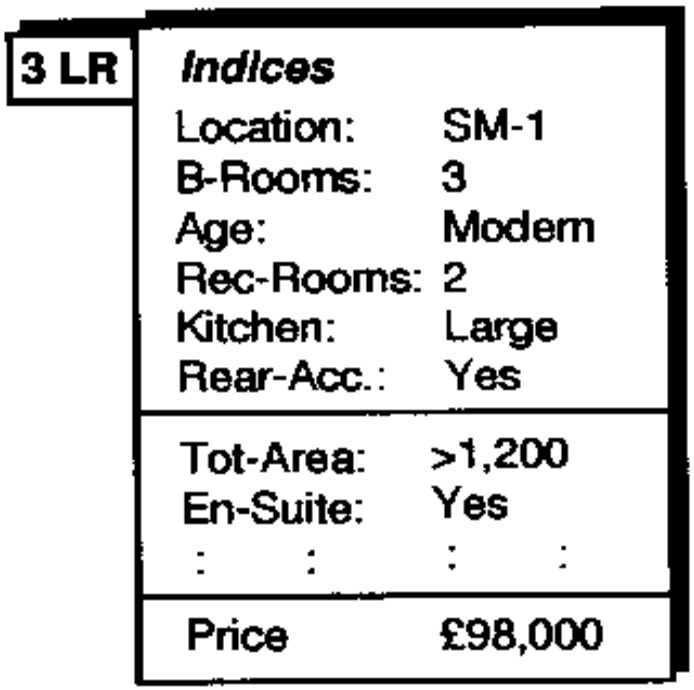
Raciocínio Baseado em Casos

- Os exemplos são representados por descrições simbólicas complexas
- Necessita de medidas de distância para dados mais complexos.
  - "Casamento" entre diferentes tipos de dados
  - Necessidade de indexação mais precisa
- Auxílio de outros métodos de aprendizado

Raciocínio Baseado em Casos

- Exemplo: Valor de Venda de Casas

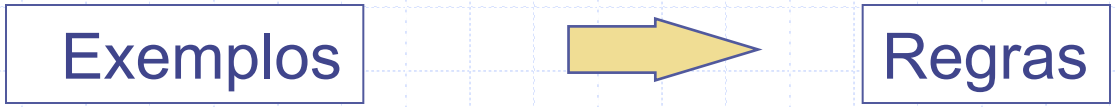




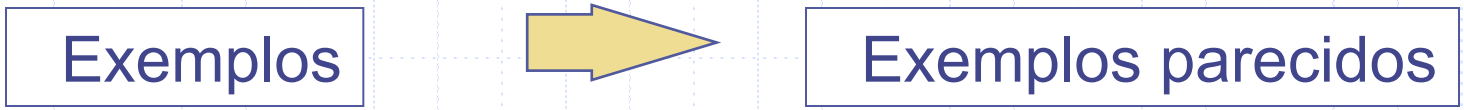
Raciocínio Baseado em Casos

- Aquisição manual do conhecimento:

- Aprendizagem indutivo:

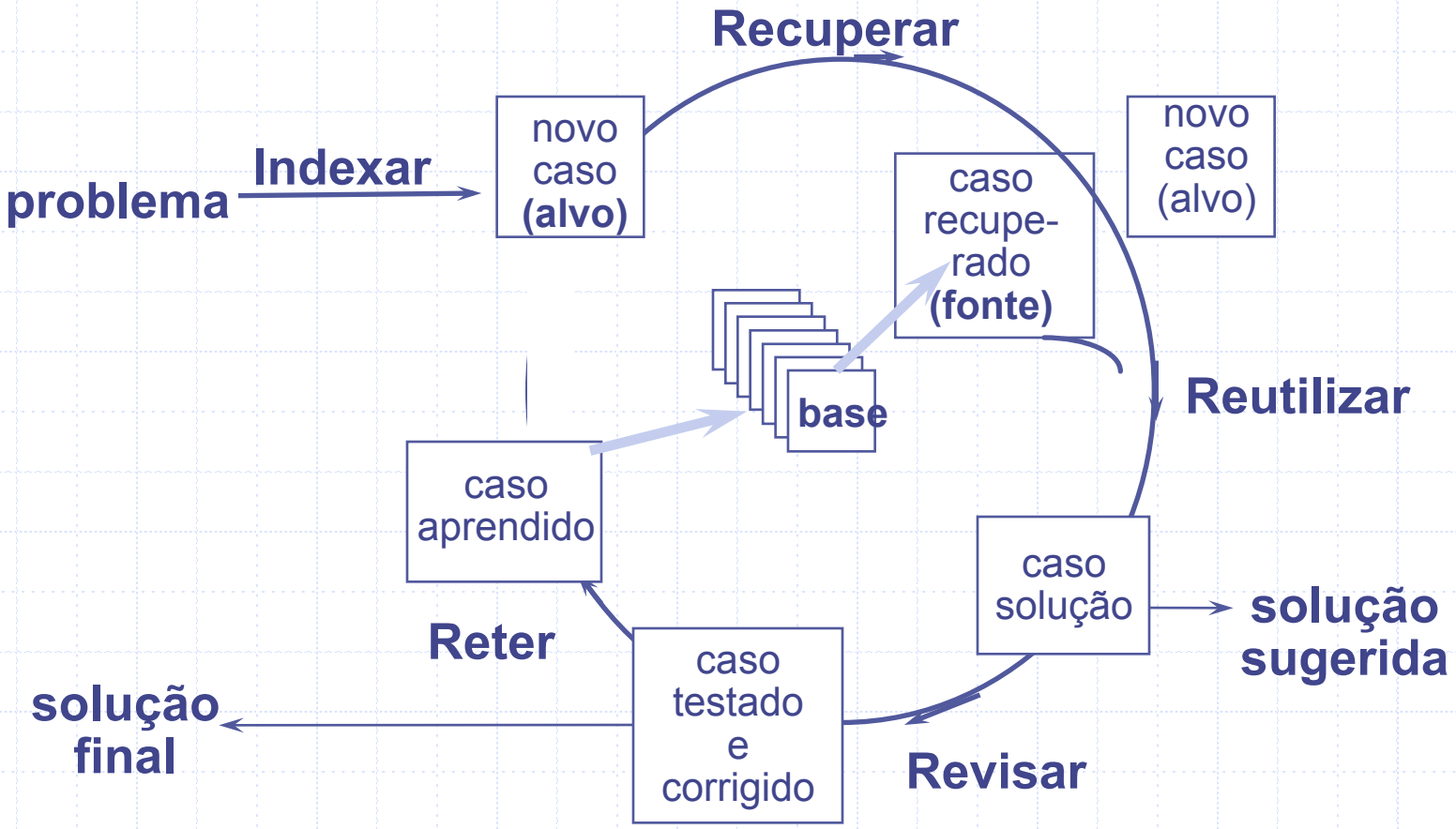


- Raciocínio Baseado em Casos (CBR)



Raciocínio Baseado em Casos

- Funcionamento do CBR



Raciocínio Baseado em Casos

- Indexação
  - Objetivo: dar ao sistema conhecimento sobre como estocar e comparar casos
  - levar em conta a utilização que se quer fazer (propósito)
    - ex. para um mecânico e para um cliente de locadora, a descrição de um automóvel é bem diferente

Raciocínio Baseado em Casos

- Recuperação
  - Dividida em 2 partes básicas
    - Encontrar os N casos mais similares ao caso alvo, a partir dos índices
    - Escolher o melhor caso em relação o alvo

Raciocínio Baseado em Casos

- Reutilização
  - Objetivo
    - Compensar as diferenças entre o problema-alvo e problema-fonte escolhido
  - Problema
    - Depende do domínio
    - Necessita de um conjunto de operadores de transformação

Raciocínio Baseado em Casos

- Revisão
  - Avaliar a solução (automaticamente ou não) após aplicada no ambiente real (simulação)
  - Reportar erro da solução proposta, caso este exista
  - Consertar o caso a partir da descrição do erro

Raciocínio Baseado em Casos

- Retenção
  - Extração da informação a reter
    - Pode ser um caso bem sucedido ou uma descrição de erro
  - Indexação
  - Inserção/integração do caso na base
    - exemplo: ajuste dos pesos de determinados atributos de um caso

Raciocínio Baseado em Casos

- Observações
  - Método novo (1993)
  - Indexação bastante importante
  - Geralmente usado com outro tipo de técnica como por exemplo, regras em lógica de primeira ordem

Paralelo entre aprendizado guloso e preguiçoso

- Guloso
  - Generaliza função de classificação
  - Treinamento lento
  - Classificação rápida
  - Aproximação global
  - Ex.: redes RBF, ID3
- Preguiçoso
  - Não generaliza a função de classificação
  - Treinamento rápido
  - Classificação lenta
  - Aproximação local ou global
  - Ex.: *k-NN*, *CBR*

Bibliografia

- Mitchell, T.M. (1997). *Machine Learning*, McGraw-Hill.
- Aamodt, A. & Plazas E. (1994). *Case-based reasoning: Foundational issues, methodological variations, and system approaches.AI Communications.*
- I.H. Witten & E. Frank (1999). *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, Morgan Kaufmann*