

Inteligência Artificial

Sistemas de Conhecimento utilizando
Raciocínio Baseado em Casos

Prof. Paulo Martins Engel

***“Um sistema de CBR resolve
problemas por adaptar soluções que
foram utilizadas para resolver
problemas anteriores.”***

Riesbeck & Schank, 1989

Caso

- Um caso é a informação relativa a um problema e solução específicas, ou seja, é uma *instância* concreta do problema;
- Um caso não é genérico pois se aplica a uma situação em um determinado contexto;
- Um caso diferencia-se das outras experiências armazenadas por contribuir de alguma forma com uma melhor solução do problema.

O Caso é Composto de

- Uma descrição dos aspectos relevantes do problema naquela situação particular, que aqui será chamada de **representação** do caso;
- O contexto no qual o caso se insere, representado através dos **índices** do caso;
- A descrição da **solução** associada ao problema
- Uma **avaliação da solução** empregada ao problema particular.

Análise de Crédito

Descrição

- Nome: Paulo Rocha
- Nascimento: 20.05.64
- Endereço: Av. Carlos Gomes, POA
- Profissão: Analista de sistemas
- Salário mensal: \$3000
- Estado civil: solteiro
- Dependentes: 0
- Cartão crédito: Visa

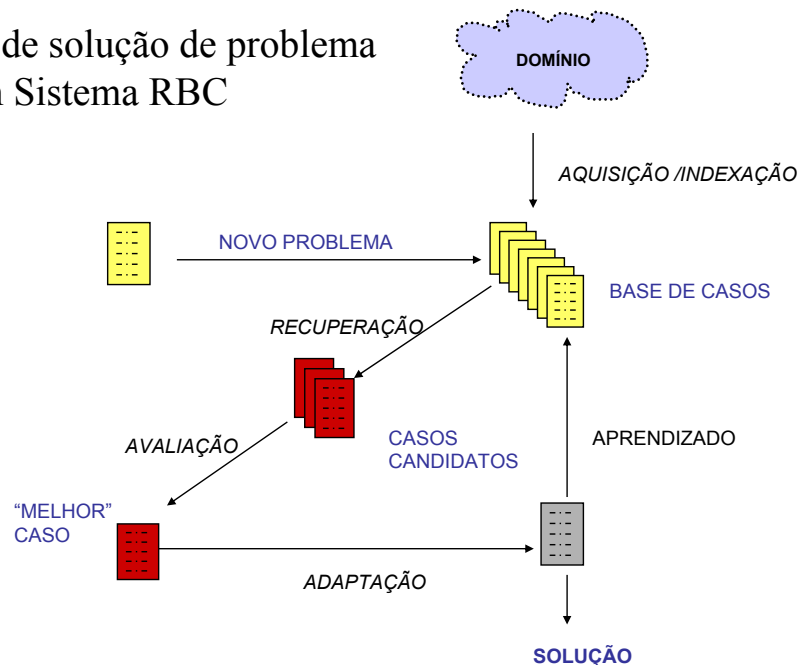
-
- Empréstimo solicitado: \$20000
 - Empréstimo concedido: sim

Solução!

Algumas Aplicações

- CFM International (França e USA) - Diagnóstico de falhas nos motores de Boeings 737
- Sepro Robotique (França) – "Help Desk" para suporte após venda de robôs
- Ansaldo Transporti (Itália) – Manutenção no metrô de Napoles
- Legrand (França) – Estimativa do custo de produção de peças em plástico

Ciclo de solução de problema de um Sistema RBC



Construir um Sistema de CBR é definir os mecanismos de

- Aquisição de casos
- Representação de casos
- Organização da base de casos
- Indexação
- Recuperação
- Adaptação

Vantagens da Aquisição de Conhecimento por Casos

- Construção de um protótipo antes de obter a completa estruturação do domínio
- Pode ser realizada mesmo em domínios pouco estruturados ou com base teórica mal definida.
- Casos são uma boa amostragem dos tipos de problemas que o sistema deve resolver
- Diminui a necessidade de entrevistas
- Encapsulamento do conhecimento

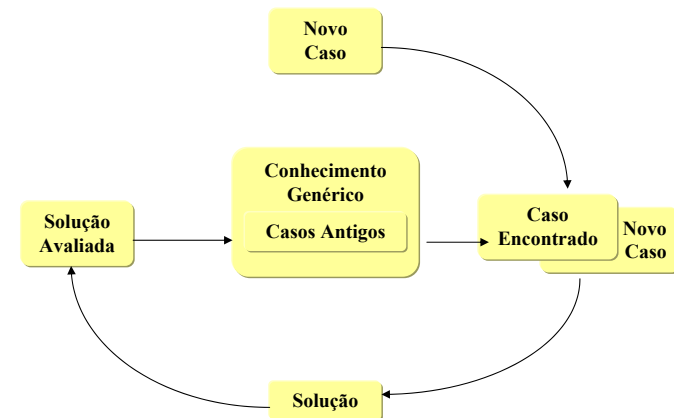
Tipos de Sistemas CBR Quanto à forma de utilização

- Sistemas completamente automatizados: resolução total de problemas
- Sistemas de recuperação de casos: trabalham de forma interativa com uma pessoa para resolver um problema.

Aspectos práticos: arquitetura e métodos

- Ciclo de sistemas CBR
- Representação de casos
- Cálculo de similaridade
- Organização da base

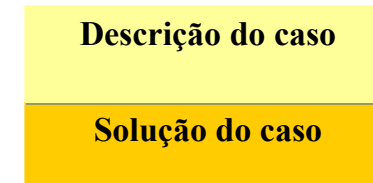
O ciclo dos sistemas CBR



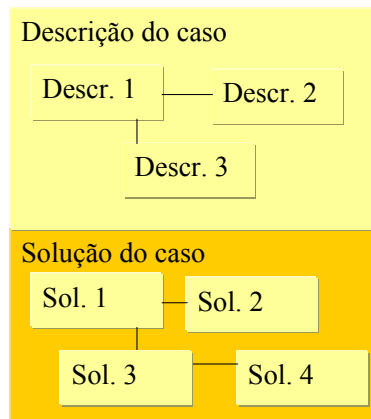
O ciclo dos sistemas CBR



Representação plana



Representação Estruturada



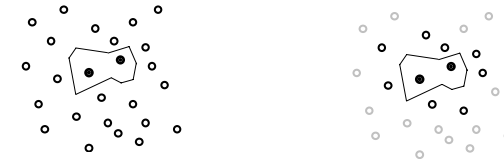
Representação baseada em casos

- Forma mais simples de aprendizado: memorizar
 - Dentre os exemplos de treinamento se busca aquele que é mais similar à nova amostra apresentada
 - Os próprios exemplos representam o conhecimento
 - Também chamado de *aprendizado baseado em casos*
- Função de similaridade define o que é “aprendido”
- Aprendizado baseado em casos é aprendizagem *preguiçosa*: adia os cálculos até o momento da consulta
- Métodos: *vizinho mais próximo*, *k vizinhos mais próximos*, ...

A função de distância

- Caso mais simples : um atributo numérico
 - Distância é a diferença entre os dois valores de atributo envolvidos (ou, alternativamente, uma outra função)
- Vários atributos numéricos: normalmente, distância euclidiana é usada e os atributos são normalizados
- Atributos nominais: distância é definida como 1 se os valores são diferentes, ou 0 se eles são iguais
- Todos os atributos são igualmente importantes?
 - Pode ser necessário ponderar os atributos

Aprendendo protótipos



- Só precisam ser armazenados os exemplos envolvidos numa decisão
- Exemplos ruidosos devem ser retirados
- Idéia: usar apenas exemplos *protótipos*

Algoritmo k -NN

- O algoritmo dos k vizinhos mais próximos (k -NN) é usado para estimar o valor de uma *função alvo* num determinado ponto do espaço de características (correspondente à posição da amostra consultada), a partir do valor desta função nos k pontos mais próximos da amostra consultada.
- A função alvo pode ser discreta (correspondente a um modelo de classificação, por exemplo), ou contínua (correspondendo a um modelo de regressão).
- O algoritmo k -NN não forma uma hipótese geral explícita para a função alvo, ele apenas calcula a classificação (ou valor da função contínua) para uma amostra específica consultada.
- O valor da função é estimado no momento da consulta.

Algoritmo dos k vizinhos mais próximos – k -NN

- Assume que as amostras correspondam a pontos no espaço de características \mathbb{R}^n
- Os vizinhos mais próximos (NN – *nearest neighbors*) de uma amostra são definidos em termos da distância euclidiana.
- Dada uma amostra x descrita por um vetor de características

$$\langle a_1(x), a_2(x), \dots, a_n(x) \rangle$$
 onde $a_r(x)$ representa o valor do r -ésimo atributo da amostra x .
- A distância de duas amostras x_i e x_j é definida como $d(x_i, x_j)$:

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

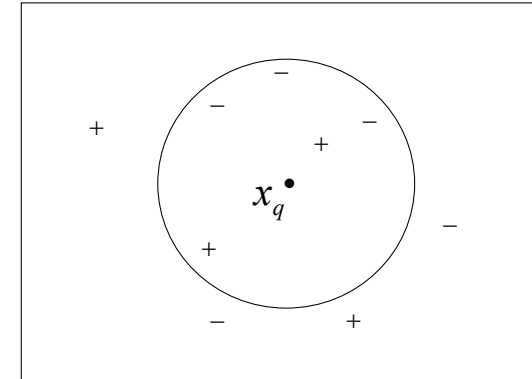
Algoritmo k -NN para valores discretos

- Considere a função discreta (por ex., *classificação*) $f: \mathbb{R}^n \rightarrow V$, com $V = \{v_1, \dots, v_s\}$ (*rótulos de classe*)
- **Treinamento:**
 - Para cada exemplo de treinamento $\langle x, f(x) \rangle$, acrescente o exemplo à lista *exemplos_treinamento*.
- **Classificação**
 - Dada uma amostra de consulta x_q para ser classificada, sendo $x_1 \dots x_k$ os k exemplos de treinamento mais próximos a x_q ,
 - Retornar

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

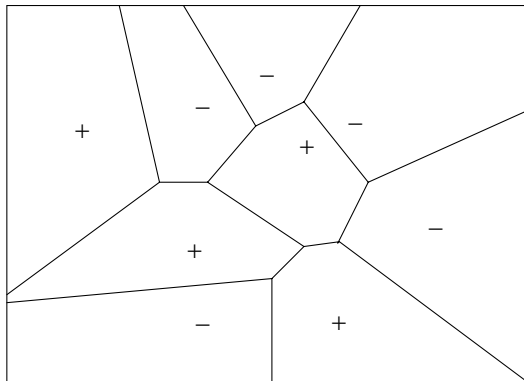
– onde $\delta(a, b) = 1$ se $a = b$, ou $\delta(a, b) = 0$, caso contrário

Exemplo do algoritmo k -NN para valores discretos



- O algoritmo 1-NN classifica x_q como positivo
- O algoritmo 5-NN classifica x_q como negativo

Superfície de decisão induzida pelo algoritmo 1-NN



- Diagrama de Voronoi: poliedros convexas em torno dos exemplos de treinamento

Algoritmo k -NN para função alvo de valor contínuo

- O aprendizado de uma função contínua é feito de forma análoga ao caso de função discreta.
- O algoritmo calcula a média dos valores da função para os k exemplos de treinamento mais próximos.
- Para aproximar uma função alvo de valor real $f: \mathbb{R}^n \rightarrow \mathbb{R}$, substituímos a última linha do algoritmo k -NN discreto por:

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

Algoritmo k -NN ponderado discreto

- Pode-se ponderar a contribuição de cada um dos k vizinhos de acordo com a sua distância à amostra consultada.
- Quanto mais próximo o vizinho, maior o peso atribuído a ele.
- Pode-se ponderar o voto de cada vizinho de acordo com o inverso do quadrado da sua distância a x_q ; substituindo a estimativa (para valor discreto) por:

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

onde

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

Ponderação de atributos para o algoritmo k -NN

- Diferentemente de outros métodos, como árvores de decisão e regras de produção, o algoritmo dos k vizinhos mais próximos leva em consideração *todos* os atributos dos exemplos para calcular as distâncias.
- Isto pode não ser interessante quando, por ex., de 20 atributos, apenas 2 são relevantes para determinar a classificação.
- Assim, duas amostras com os 2 valores relevantes coincidentes podem, apesar disso, ficar distantes entre si no espaço de 20 dimensões, levando a erros de classificação.
- A solução usual para isso é atribuir pesos diferentes para cada atributo, no momento do cálculo da distância.
- A determinação dos pesos pode ser obtida por métodos como *cross-validation* e árvores de decisão.

Algoritmo k -NN ponderado contínuo

- Podemos também ponderar pela distância os exemplos para funções alvo de valor real, substituindo a fórmula da estimativa do valor por:

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

onde

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

Cálculo da distância ponderada

- A ponderação dos atributos é equivalente a escalar os eixos de cada atributo correspondentemente à sua importância relativa para a estimativa da função desejada.
- A distância de duas amostras x_i e x_j , levando em consideração pesos diferentes para cada atributo, é definida como $d(x_i, x_j)$:

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

onde w_r é o peso do r -ésimo atributo de uma amostra.

Adaptação

⇒ Substituição

⇒ Reinstanciação

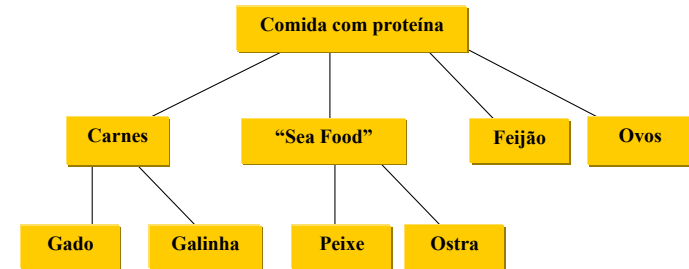
⇒ Ajuste de Parâmetros

⇒ Substituição baseada em casos

⇒ Transformação

Reinstanciação

⇒ Instanciação de soluções antigas com valores « substituíveis ».



Substituição baseada em casos

- ⇒ A técnica de *adaptação por reinstanciação* pode ser adequada se existe na memória uma rede semântica que descreve o problema e que nos permite substituir um elemento da rede por um outro elemento descrito sob a mesma classe.
- ⇒ A *substituição baseada em casos* faz a procura de um substituto para o elemento que necessita ser modificado numa base de casos.
- ⇒ Exemplo: Um problema de preparação de um menu, temos um convidado que é vegetariano e temos que substituir uma lasanha a bolognesa por outro prato.
- ⇒ O conjunto de casos armazenados na base que incluem “massas” é percorrido, à busca de uma alternativa para a lasanha.

Métodos de transformação

- ⇒ Exemplo: Problema da preparação do menu para um grupo de convidados. Como na situação anterior, já tínhamos selecionado o caso com o menu de lasanha bolognesa. Mas lasanha vai contra a nossa restrição “vegetariano”.
- ⇒ Mas aqui, ao invés de substituir a lasanha por outro prato, nós vamos tentar transformar a lasanha em “lasanha vegetariana”.
- ⇒ Como carne é um ingrediente secundário numa lasanha, podemos simplesmente remover carne do prato através de uma heurística chamada: *remover ingrediente secundário*.
- ⇒ Outra solução seria substituir leite por outro produto que cumpriria a mesma função no prato, usando a heurística de substituição chamada “substituir item”, por carne de soja por exemplo.

Métodos de indexação e busca

- ⇒ Indexar casos quer dizer definir caminhos (atalhos) que nos levam de fatos a casos.
- ⇒ Isto nos permite que quando estamos procurando pelo caso mais similar na base, não tenhamos que percorrer toda a base, mas possamos utilizar este "atalho"

Métodos de indexação

Manual

- ⇒ a pessoa tem que analisar casos e dizer “este caso é importante por essa ou aquela razão”.
- ⇒ Definição de uma *checklist* - (trabalho de aquisição de conhecimento);

Automático

- ⇒ Indução e árvores de decisão