CLARANS

Diego Raphael Amancio

Resumo

- Introdução
- PAM
- CLARA
- Modelagem como grafo
- CLARANS
- Comparação de eficiência
- Conclusão

Artigo de Referência [1]

Efficient and Effective Clustering Methods for Spatial Data Mining

Raymond T. Ng
Department of Computer Science
University of British Columbia
Vancouver, B.C., V6T 1Z4, Canada
rng@cs.ubc.ca

Jiawei Han
School of Computing Sciences
Simon Fraser University
Burnaby, B.C., V5A 1S6, Canada
han@cs.sfu.ca

Imagem do artigo original entitulado "Efficient and Effective Clustering Methods for Spatial Data Mining (1994)

Introdução

- Algoritmo projetado para trabalhar com dados espaciais
  - Banco de dados espaciais geralmente são muito grandes
- Algoritmos tradicionais não são projetados para trabalhar com grandes banco de dados
- Pode ser aplicado para base de dados com outros tipos de atributos, mas só usa os atributos espaciais

Características Gerais

- Baseado em busca aleatória
- Motivado por dois outros algoritmos:
  - PAM
  - CLARA
- Baseado em k-medóides
  - Pode ser empregado em bases relacionais
  - Robusto à outliers
- É mais eficiente que os anteriores

PAM

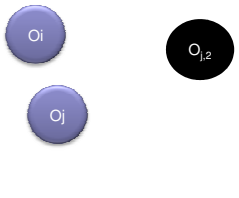
- Para encontrar *k* clusters, determina um objeto representativo para cada cluster (medóide)
  - Medóide deve ser o objeto mais ao centro do cluster
- Após a escolha dos medóides, outros objetos são agrupados em torno do medóide mais próximo
- A qualidade das partições obtidas é dada como a média de todas as dissimilaridades de objetos aos seus respectivos medóides

PAM – Idéia Geral

- Os primeiros *k* medóides são escolhidos aleatoriamente
- A cada passo é feita uma troca entre medóide e objeto não selecionado, supondo que a troca melhore a qualidade do agrupamento
- Seja:
  - O_i → medóide
  - O_h → objeto não selecionado como medóide
  - C_{jih} → custo para objeto O_j de trocar medóide O_i por objeto O_h
- Quatros tipos de definições para C_{jih}

PAM – Idéia Geral

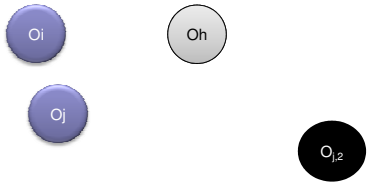
- Primeiro Caso
  - Condição 1: O_j no cluster de O_i
  - Condição 2: O_j é **mais** similar a $O_{j,2}$ (segundo medóide mais similar a O_j) do que a O_h
  - Conclusão: se a substituição de O_i por O_h, então O_j deve pertencer a $O_{j,2}$



$$C_{jih} = d(O_j, O_{j,2}) - d(O_j, O_i).$$

PAM – Idéia Geral

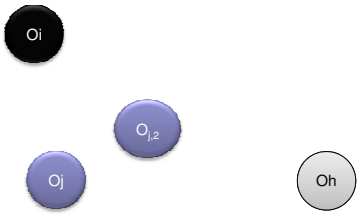
- Segundo Caso
  - Condição 1: O_j no cluster de O_i
  - Condição 2: O_j é **menos** similar a $O_{j,2}$ (segundo medóide mais similar a O_j) do que a O_h
  - Conclusão: se a substituição de O_i por O_h, então O_j deve pertencer a O_h



$$C_{jih} = d(O_j, O_h) - d(O_j, O_i).$$

PAM – Idéia Geral

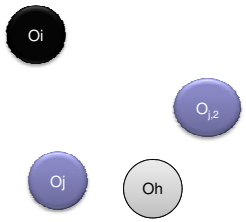
- Terceiro Caso
  - Condição 1: O_j está em outro cluster: não em O_i
  - Condição 2: O_j é **mais** similar a $O_{j,2}$ (medóide de O_j) do que a O_h
  - Conclusão: se a substituição de O_i por O_h, então O_j deve continuar a pertencer $O_{j,2}$



$$C_{jih} = 0.$$

PAM – Idéia Geral

- Quarto Caso
  - Condição 1: O_j está em outro cluster: não em O_i
  - Condição 2: O_j é **menos** similar a $O_{j,2}$ (medóide de O_j) do que a O_h
  - Conclusão: se a substituição de O_i por O_h, então O_j deve pertencer a O_h



$$C_{jih} = d(O_j, O_h) - d(O_j, O_{j,2}),$$

PAM – Idéia Geral

- Custo final da troca:

$$TC_{ih} = \sum_j C_{jih}$$

PAM – O algoritmo

1. Selecionar *k* medóides aleatórios
2. Calcular TC_{ih} para todos os pares de objetos, onde O_i é medóide e O_h é objeto
3. Selecionar o par com valor mínimo de custo TC_{ih}. Se o custo é negativo, fazer a substituição e voltar ao passo 2
4. Caso contrário, para cada objeto não selecionado, encontrar o objeto mais próximo

CLARA

- Clustering Large Applications
- Em vez de encontrar os medóides considerando todos os objetos, faz uma amostragem e aplica o PAM
- Resultados experimentais mostram que 5 amostras de tamanho 40 + 2k são suficientes

CLARA – O algoritmo

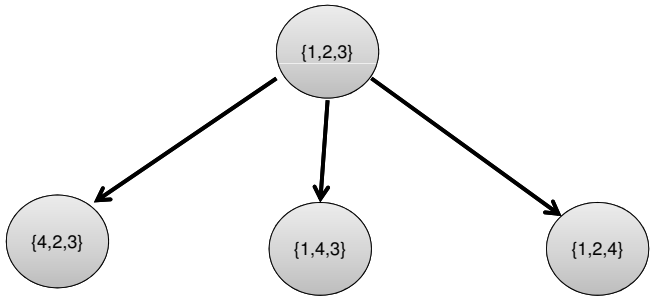
- Repetir para i = 1 até 5
  - Amostrar com tamanho 40 + 2k e encontrar k medoides com PAM
  - Para cada objeto do banco inteiro, determinar o medóide mais próximo
  - Encontrar a qualidade do cluster obtido no passo anterior usando o banco inteiro.
  - Se a qualidade é a melhor encontrada até o momento, armazenar

Modelagem como grafo

- Modelagem de PAM e CLARA como grafo
  - Cada vértice é representado pelo conjunto de *k* medóides atuais
    - Ex.: $v_1 = \{O_{m1}, O_{m2}, \ldots, O_{mk}\}$
  - Dois vértices são vizinhos se o conjunto de medóides diferem por apenas um elemento
    - Ex.: $v_2 = \{O_{mx}, O_{m2}, \ldots, O_{mk}\}$ é vizinho de v_1

Modelagem como grafo

- Exemplo com n=4 e k=3



Modelagem como grafo

- Visão de Grafo
  - Cada vértice possui k(n-k) vizinhos
  - Cada vértice representa um resultado de clustering, já que define quem são os medóides

Modelagem como grafo

- PAM
  - A cada passo, todos os vizinhos do vértice atual são analisados
  - O próximo vizinho é escolhido como aquele que mais diminui o custo, até que o mínimo seja obtido
  - Se n e k é alto, é custoso analisar k(n-k) vizinhos a cada passo

Modelagem como grafo

- CLARA
  - Restringe a busca em subgrafos menores
  - Diminui *n* e depois constrói o grafo
  - Problema: o subgrafo é restringido apenas aos objetos pertencentes à amostra
  - Se um dado vértice M é mínimo é a amostragem não considerou tal vértice, nunca o melhor caso será encontrado

CLARANS

- Características
  - Não restringe a busca a um subgrafo
  - Exatamente igual ao PAM, mas analisa apenas uma amostra dos vizinhos
  - Enquanto CLARA faz amostragem no começo do método, CLARANS faz amostragens a cada passo da busca

O algoritmo

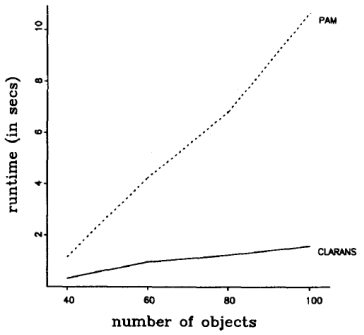
- Parâmetros:
  - *Numlocal* – número de iterações de busca
  - *Maxneighbor* – número máximo de vizinhos analisados a cada passo
- *mincost* (custo atual da melhor partição) ← infinito
- 1. *i* ← 1 (primeira iteração de busca)
- 2. Setar *current* para um vértice arbitrário
- 3. *j* ← 1 (primeiro vizinho analisado)
- 4. Escolher um vizinho aleatório e calcular custo diferencial
- 5. Se vizinho tem custo menor, *current* é setado como vizinho e volta ao passo 3
- 6. Senão, *j*++. Se *j* < *maxneighbor*, vai para passo 4
- 7. Armazena *current* como *bestnode* e custo como *mincost*
- i++. Se *i* > numlocal, retorna *bestnode* como saída, senão retorna ao passo 2

Parâmetros

- *Maxneighbor*
  - Quanto maior, mais próximo o algoritmo se torna do PAM
- Experimentos mostram um valor satisfatório para [2]:
  - *Numlocal* – 2
  - *Maxneighbor* – max{0.0125 * k (n – k), 250}

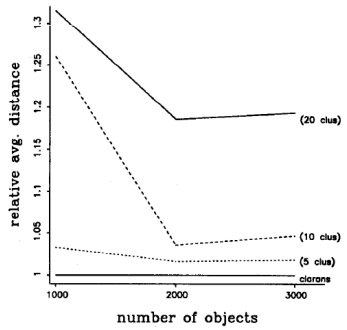
Comparação de Eficiência

- CLARANS x PAM
  - Qualidade dos grupos é semelhante (5 grupos)



Comparação de Eficiência

- CLARANS x CLARA
  - Para o mesmo tempo de processamento



Conclusão

- CLARANS é um método baseado em busca em grafo
- Busca é limitada a um subconjunto de vizinhos
  - Eficiência de tempo com relação a PAM
  - Eficiência de qualidade com relação a CLARA

Referências

- [1] Ng, Raymond T and Han, Jiawei. Efficient and Effective Clustering Methods for Spatial Data Mining, 1994.
- [2] G. Piatetsky-Shapiro and W.J. Frawley. Knowledge Discovery in Databases, AAAI/MIT Press, 1991.