Projected Clustering Algorithm

Emanuel Matos – 5560105
Disciplina : Análise de Agrupamentos
Prof. Dr. Ricardo Campello
Dez/2010

View

- Problema
- Objetivo
- Definições/Premissas
- Algoritmo
- Acurácia
- Escalabilidade
- Conclusões

Problema

- Dado um conjunto de objetos no espaço, encontrar uma partição de pontos em clusters cujos pontos em cada cluster estejam próximos um do outro.
- Em altas dimensões é dificil encontrar Cluster em função do espaçamento entre os dados.
- Redução de dimensões incorre em possível perda de informação.

3

Objetivo

- Método de encontrar Clusters numa pequena projeção de subespaço para dados de alta dimensionalidade.
- Reduzir a dimensionalidade , selecionando as dimensões de interesse.

4

Definições / Premissas – 1

- Assumimos que temos como parâmetros ***k e l***

  ***k** numero de clusters e **l** numero médio de dimensoes no cluster e $k * l$ tem que ser inteiro*

- **N** total de objetos e **d** dimensionalidade
- $C = \{x_1, x_2, \ldots, x_t\}$, conjunto de objetos do Cluster
- Centroide é uma média algébrica dos objetos:

$$\overline{x}_c = \sum_{i=1}^{t} x_i / t$$

- Radius : distância média d(.,.) de cada ponto ao Centróide

$$r_c = \sum_{i=1}^{t} d(\overline{x}_c, x_i)/t$$

5

Definições / Premissas – 2

- Varias distâncias podem ser utilizadas, dependem de dominio do problema.
- Para exemplificar, duas se destacam, a Distância Euclidiana e a distância de Manhattan – Ambas são derivações das Normas.

$$d'_p(x_1, x_2) = \left(\sum_{i=1}^{d} |x_{1,i} - x_{2,i}|^p \right)^{1/p}$$

- Para este algoritmo foi utilizada uma derivação da Distância de Manhattan, foi chamada Distância Segmentada de Manhattan, qual foi definida em função da Dimensionalidade D

6

Manhattan segmental distance

- Dados dois pontos $x_1 = \{x_{1,1}, x_{1,2}, \dots, x_{1,d}\}$ e $x_2 = \{x_{2,1}, x_{2,2}, \dots, x_{2,d}\}$
- Para qualquer tamanho de dimensão D, |D| ≤ d , a distância relativa Manhatan Segmental Distance entre dois pontos é dada por :

$$d_D(x_1, x_2) = \left(\sum_{i \epsilon D} |x_{1,i} - x_{2,i}|\right) / |D|$$

- Útil na comparação de diferentes clusters tendo numero de dimensões variados pois acaba por normalizar pelas dimensões.

7

Algoritmo PROCLUS

- Fase 1 – Inicialização
- Fase 2 – Iteratividade
- Fase 3 - Refinamento

8

{Fase 1 – Inicialização}

- Já temos definido o numero de Clusters que queremos (Parametro k) e a Média de Dimensões (Parametro l)
- A proposta da fase de inicialização é de redução do tamanho do conjunto de dados para simultaneamente selecionar dados representativos do conjunto.
- Utilizando uma técnica de "Hill Climbing" , num processo de sucessivos ganhos para o conjunto de medoides.

9

{Fase 2 – Iteratividade}

- Já com a primeira formação dos clusters varre-se todo o conjunto de dados e progressivamente vai "qualificando " os melhores Clusters e Dimensões que ficarão no modelo.

10

{Fase 3 – Refinamento}

- Depois de encontrar o "melhores Clusters", faz-se mais um passo onde se utiliza a distribuição dos pontos diferente da fase anterior que utiliza a localização dos centroides.
- Outliers tambem são tratados neste ultimo passo.

Para cada medoide m_i e novo conjunto de dimensões $\mathcal{D}_i$ encontramos a menor distância de Manhattan Segmentada Δ_i para todos os outros (k-1) medoides da dimensão $\mathcal{D}_i$:

$$\Delta_i = \min_{j \neq i} d_{\mathcal{D}_i}(m_i, m_j)$$

11

Algoritmo PROCLUS

- Deve encontrar os centros dos clusters e as dimensões apropriadas.
  - I. Utilizando K-Medoides acham-se os pontos centrais dos clusters.
  - II. Qualquer Cluster que N/k * Desvio minimo menor que 1, (foi escolhido 0.1) ,considera-se um cluster ruim na formação)
  - FI

Algorithm PROCLUS (N. de Clusters: k, Média de dimensões: l)

$\{\mathcal{C}_i$ é o $i-$ esimo $cluster\}$
$\{\mathcal{D}_i$ é a $i-$ esima dimensão associada ao $\mathcal{C}_i$ $cluster\}$
$\{\mathcal{M}_{current}$ são os medoides da iteração em andamento$\}$
$\{\mathcal{M}_{best}$ é o melhor conjunto de medoides encontrados$\}$
$\{\mathcal{N}$ é o conjunto final de medóides com associação às dimensões$\}$
$\{\mathcal{A}, \mathcal{B}$ são constantes e inteiros$\}$

$\{\boldsymbol{Fase\ 1 - Inicialização}\}$
S = amostra aleatória tamanho $\mathcal{A} * k$
$\mathcal{M}$ = GREEDY $(\mathcal{S}, \mathcal{B} * k)$

$\{\boldsymbol{Fase\ 2 - Iteratividade}\}$
BestObjective = ∞
$\mathcal{M}_{current}$ = medoides aleatórios $\{m_1, m_2, \ldots m_k\} \subset \mathcal{M}$
Repetir
  Para cada medoide $m_i \in \mathcal{M}_{current}$ faça
    Seja δ_i a distancia para o próximo medoide de m_i
    $\mathcal{L}_i$ = pontos da esfera centrados em m_i com raio δ_i
    fim;
    $\mathcal{L} = \{\mathcal{L}_1, \ldots, \mathcal{L}_k\}$
    $(\mathcal{D}_1, \mathcal{D}_2, \ldots \mathcal{D}_k)$ = Procuradimensao$(k, l, \mathcal{L})$
    { formação de clusters}
      $(\mathcal{C}_1, \ldots, \mathcal{C}_k)$ = AcessaPontos$(\mathcal{D}_1, \ldots, \mathcal{D}_k)$
      FuncaoObjetivo = Avaliaclusters$(\mathcal{C}_1, \ldots, \mathcal{C}_k, \mathcal{D}_1, \ldots, \mathcal{D}_k)$
        Se FuncaoObjetivo < BestObjective então
        Faça
        BestObjective = FuncaoObjetivo
        $\mathcal{M}_{best} = \mathcal{M}_{current}$
        Guarde $\mathcal{M}_{best}$ com medoides
        Fim
      Guarde $\mathcal{M}_{current}$ com valores de $\mathcal{M}_{best}$
      Atualize $\mathcal{M}_{best}$ com valores aleatórios de $\mathcal{M}$
Até todos medoides tenham sido varridos

$\{\boldsymbol{Fase\ 3 - Refinamento}\}$
$\mathcal{L} = \{\mathcal{L}_1, \ldots, \mathcal{L}_k\}$
$(\mathcal{D}_1, \mathcal{D}_2, \ldots \mathcal{D}_k)$ = Procuradimensao$(k, l, \mathcal{L})$
$(\mathcal{C}_1, \ldots, \mathcal{C}_k)$ = AcessaPontos$(\mathcal{D}_1, \ldots, \mathcal{D}_k)$
$(\mathcal{N} = (\mathcal{M}_{best} = (\mathcal{D}_1, \ldots, \mathcal{D}_k))$
  Retorna$(\mathcal{N})$

12

GREEDY

Algorithm GREEDY (Set of points: S, Numero de medoides: k)

$\{d(.,.)$ *é a função distância*$\}$

Inicio
$\mathcal{M} = \{m_1\}$ $\{m_1$ *é um ponto aleatório de* $S(dados)\}$
$\{calculo\ da\ distância\ entre\ os\ pontos\ e\ m_1\}$
Para cada $x \in S \setminus \mathcal{M}$
$dist(x) = d(x, m_1)$
Para $i = 2$ *até* k
Faça
$\{escolha\ dos\ m_i\ medoides\ que\ sejam\ longe\ do\ anterior\}$
Seja $m_i \in S \setminus \mathcal{M}$
$dist(m_i) = \max\{dist(x) \mid x \in S \setminus \mathcal{M}\}$
$\mathcal{M} = \mathcal{M} \cup \{m_i\}$
{Calculo das distâncias de cada ponto ao medóide mais próximo}
Para cada $x \in S \setminus \mathcal{M}$
$dist(x) = \max\{dist(x), d(x, m_i)\}$
Fim
Retorne $\mathcal{M}$
Fim

Procuradimensao (FindDimension)

Algorithm Procuradimensao $(k, l, \mathcal{L})$

Inicio

$\{d$ *é o total de numero de dimensões*$\}$
$\{X_{i,j}$ *é a distância média dos pontos em* $\mathcal{L}_i$ *para o medoide* m_i *na dimensão* $j\}$
Para cada medoide m_i faça:
Inicio
$Y_i = \frac{\sum_{j=1}^{d} X_{i,j}}{d}$
$\mathcal{D}_i = \emptyset$
$\sigma_i = \sqrt{\frac{\sum_{j=1}^{d} (X_{i,j} - Y_i)^2}{d-1}}$
Para cada dimensão j faça $Z_{i,j} = (X_{i,j} - Y_i)/\sigma_i$
Fim
Capture $k * l$ numeros no mínimo de valores de $Z_{i,j}$ sujeito
a restrição de no mínimo 2 dimensões para cada cluster.
Se $Z_{i,j}$ é selecionado então some a dimensão j em $\mathcal{D}_i$
Retorne $(\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_k)$
FIM

Acessapontos (AssignPoints)

Algorithm Acessapontos $(\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_k)$
Inicio
Para cada $i \in \{1, \dots, k\}$ faça $\mathcal{C}_i = \emptyset$
Para cada *ponto p* faça
    Inicio
    Seja $d_{\mathcal{D}_i}(p, m_i)$ Manhattan Segmental Distance do ponto
    p *ao medoide* m_i *relativa a dimensão* $\mathcal{D}_i$;
    Encontre o i de menor valor de $d_{\mathcal{D}_i}(p, m_i)$ e adicione p em $\mathcal{C}_i$;
    Fim
Retorne $(\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_k)$
Fim

Avaliacluster (EvaluateCluster)

Algorithm Avaliacluster $(\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_k, \mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_k)$
Inicio
Para cada $\mathcal{C}_i$ faça:
    Inicio
    Para cada dimensão $j \in \mathcal{D}_i$ faça:
        Inicio
        $Y_{i,j}$ = *Distância média dos pontos em* $\mathcal{C}_i$ *para o centroide de* $\mathcal{C}_i$
        pertencente a dimensão j
        Fim
    $w_i = \frac{\sum_j Y_{i,j}}{|\mathcal{D}_i|}$
    Fim
Retorne $\frac{\sum_{i=1}^{k} |\mathcal{C}_i| * w_i}{\mathcal{N}}$
Fim

Acurácia

- Foi realizado o seguinte experimento:
  - Foram gerados 2 set de dados de 100.000 objetos e 20 dimensões e k=5.
    - 1. Case, l=7 (todos iguais) e k=5 na tabela abaixo os inputs e outputs,e a Matriz de Confusão:

Case 1

Input	Dimensions	Points
A	3,4,7,9,14,16,17	21391
B	3,4,7,12,13,14,17	23278
C	4,6,11,13,14,17,19	18245
D	4,7,9,13,14,16,17	15728
E	3,4,9,12,14,16,17	16357
Outliers	-	5000

Found	Dimensions	Points
1	4,6,11,13,14,17,19	18701
2	3,4,7,9,14,16,17	21915
3	3,4,7,12,13,14,17	23975
4	4,7,9,13,14,16,17	16018
5	3,4,9,12,14,16,17	16995
Outliers	-	2396

Case 1

Output \ Input	A	B	C	D	E	Outlier
1	0	0	18245	0	0	456
2	21391	0	0	0	0	523
3	1	23278	0	101	0	697
4	0	0	0	15728	0	290
5	0	0	0	0	16357	638
Outliers	0	0	0	0	0	2396

Acurácia

2. Case, l=4 (2:2,1:3,1:6 e 1:7) e k=5 na tabela abaixo os inputs e outputs, e a Matriz de Confusão:

Case 2

Input	Dimensions	Points
A	2,3,4,9,11,14,18	21391
B	2,3,7	23278
C	2,12	18245
D	2,3,4,12,13,17	15728
E	2,4	16357
Outliers	-	5000

Found	Dimensions	Points
1	2,3,7	22051
2	2,4	16800
3	2,3,4,12,13,17	15387
4	2,12	18970
5	2,3,4,9,11,14,18	21498
Outliers	-	5294

Case 2

Output \ Input	A	B	C	D	E	Outlier
1	0	20992	267	416	18	358
2	34	0	0	0	16097	669
3	0	9	1	15309	10	58
4	0	2256	16536	0	0	178
5	21357	0	0	2	10	129
Outliers	0	21	1441	1	222	3609

Acurácia - Resultado

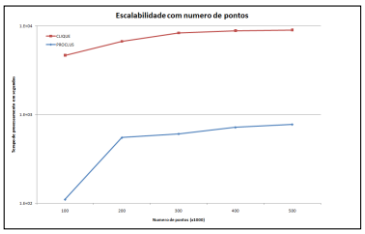
A porcentagem de erro de classificação é insignificante depois de aplicar os teste.

Existe uma perfeita correspondência entre as dimensões das saidas dos clusters e entradas.

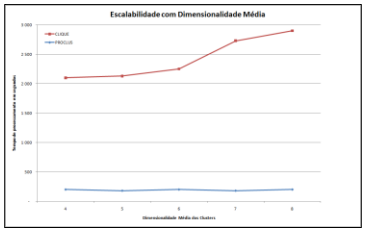
O Resultado importante é que não requer uma boa partição dos dados e sim informação de quantas dimensões ou atributos são relevantes para cada partição.

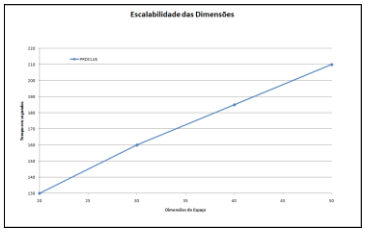
19

Escalabilidade - Comparativa



Para cada iteração do PROCLUS tivemos O(N.k.l), e no máximo teriamos O(N.k.d). O aumento da dimensionalidade é basicamente linear ao tempo de processamento no PROCLUS.





20

Conclusões

- PROCLUS ou projected clustering foi proposto para descobrir em altas dimensões grupos interessantes.
- Não trabalha com todos os tipos de objetos.
- Comparativo ao CLIQUE, que é outro para alta dimensionalidade, o PROCLUS se sai melhor quando se deseja partições e analises de tendência tem melhor interpretabilidade dos resultados.

21

Referência

- Fast Algorithms for Projected Clustering - Aggarwal,C.C.;Wolf,J,L;Yu,P. S.;Procopiuc,C.;Park,J. S. – ACM Digital Library – 1999

- Gráficos e Tabelas gerados a partir da referência.

22