Avaliação de Algoritmos para Estimação do Número de Grupos em Problemas de Mineração de Dados

Silva,Tassyo Tchesco [1], Abreu, Andrêssa Finzi de [2],
[1,2] Universidade Federal de Uberlândia
tassyo@comp.ufu.br

1 - Introdução

Mineração de Dados (MD) é um ramo da computação que teve início nos anos 80, quando os profissionais das empresas e organizações começaram a se preocupar com as grandes quantidades de dados informáticos estocados e inutilizados dentro da empresa. De acordo com Tan (2005), MD é o processo de automaticamente descobrir informação útil em grandes repositórios de dados. A mineração é um ramo multidisciplinar, que envolve tecnologia de banco de dados, aprendizado de máquina, visualização de dados, estatística, entre outras áreas.

Agrupamento de dados é uma das principais tarefas realizadas pela área de MD. Organizar dados em grupos é um dos modos mais fundamentais de compreensão e aprendizagem (Jain e Dubes, 1988). A análise de grupos divide os dados em grupos que são significativos, úteis ou ambos, usando somente informações encontradas nos dados que descrevem os objetos e seus relacionamentos (Tan, 2005). O objetivo do agrupamento é encontrar uma organização conveniente e válida dos dados.

Segundo Everitt (2001), um grupo é uma agregação de objetos no espaço de tal forma que a distância entre quaisquer dois objetos em um grupo é menor que a distância entre qualquer objeto do grupo e qualquer objeto que não esteja no grupo.

Formalmente, o problema de agrupamento é definido da seguinte forma (Dias e Ochi, 2003): Dado um conjunto *X* com *n* elementos, $X = \{X_1, X_2, ..., X_n\}$, o problema de agrupamento consiste na obtenção de um conjunto de k grupos, $C = \{C_1, C_2, ..., C_k\}$, tal que os elementos contidos em um grupo C_i possuam uma maior similaridade entre si do que com os elementos de qualquer um dos demais grupos do conjunto *C*. O conjunto *C* é considerado um agrupamento com k grupos.

Existem diferentes algoritmos de agrupamento disponíveis na literatura, dentre eles, pode-se destacar o algoritmo *K-Means*. Um dos problemas encontrados neste algoritmo é que o número de grupos *k* deve ser fornecido previamente pelo usuário. Em muitos problemas reais, o número de grupos não é conhecido previamente, dificultando a execução do algoritmo de agrupamento (Peleg and Moore, 2000).

Para resolver este problema, existem diferentes algoritmos para a escolha eficiente do número de grupos *k* como, o *X-Means* e o *OMRk.* O objetivo deste trabalho é analisar os algoritmos *X-Means* e *OMRk* na escolha do melhor número de grupos, usando bases de dados artificiais e de problemas reais. A medida de qualidade usada para avaliar o agrupamento é a silhueta simplificada.

2 - Materiais e métodos

A metodologia a ser usada consiste na comparação de dois algoritmos para a escolha do valor de k, *X-Means* e *OMRk*. Cada algoritmo será executado em diferentes bases de dados, a fim de que o melhor número de grupos para cada base seja encontrado. A seguir, o algoritmo *K-Means* será executado, usando o valor de k encontrado por cada algoritmo, e algumas medidas da qualidade do agrupamento serão obtidas.

2-1 - X-Means

O *X-Means* (Peleg and Moore, 2000) tem como entrada um intervalo de possíveis valores de *k* (maior e menor valor possível para *k*). As saídas são os centros de cada grupo e o valor do *k*.

A idéia principal do algoritmo é definida da seguinte forma: i) iniciar o algoritmo com *k* sendo o menor valor do intervalo e ii) acrescentar novos grupos até o maior valor do intervalo. Durante o processo, o conjunto de grupos com o melhor resultado é armazenado.

O algoritmo do *X-Means* é definido da seguinte forma:

Algoritmo *X-Means*:

```
Entrada: minK, maxK
Saída: melhorK
1: Para K = minK até maxK faça
2:   Melhorar_Parametros();
3:   Melhorar_Estrutura();
4:   Registra_Melhor_Resultado();
5: Fim Para
6: melhorK = Retornar_NroGrupos();
```

O procedimento Melhorar_Parametros() executa o *K-means*, a fim de obter um agrupamento. O procedimento que descobre se, e onde os novos grupos devem aparecer é o Melhorar_Estrutura(). Nessa etapa também ocorre também a divisão de grupos em dois novos grupos, usando duas diferentes estratégias propostas no trabalho. O método Registrar_Melhor_Resultado() armazena o melhor valor de k encontrado. O método Retornar_NroGrupos() retorna o número de grupos finais encontrados pelo algoritmo.

2-2- OMRk

O algoritmo *OMRK* (Naldi et al.,2011) (*Ordered Multiple Runs of K-Means*) é uma abordagem pseudo-exaustiva, que executa o *K-Means* repetidamente vezes, em busca do agrupamento que minimize um índice relativo de validação de grupos. O índice utilizado neste trabalho é a silhueta simplificada (Vendramin et al.,2010).

O algoritmo tem como entrada o maior valor possível para k. O retorno é o melhor valor de k para a base de dados. A execução do *K-Means* necessita da escolha dos k centros iniciais. A escolha destes k centros foi realizada usando o algoritmo *K-Means*++ (Arthur e Vassilvitskii, 2007).

O algoritmo *OMRk* é descrito a seguir:

Algoritmo *OMRk*

```
Entrada: maxK
Saída: melhorK
1: Para k=2 até maxK  faça
2:    Res= Executa_KMeans(k);
3:    Calcula_Silhueta(Res);
4: }
5: melhorK=grupMelhor_Silhueta();
```

A silhueta é definida de acordo com a Equação 1 e 2.

$$Sx_i = \frac{b_{p,i} - a_{p,i}}{\max(a_{p,i}, b_{p,i})} \qquad (1)$$

$$Silhueta = \frac{1}{N}\sum_{i=1}^{N} Sx_i \qquad (2)$$

Sendo $a_{p,i}$ dissimilaridade média do i-ésimo objeto ao seu cluster e $b_{p,i}$ dissimilaridade média do i-ésimo objeto ao cluster mais próximo.

2-3- Bases de Dados utilizadas

Neste trabalho foram utilizadas cinco bases de dados, disponíveis no site do UCI (UCI, 2010), e uma base de dados gerada artificialmente utilizando o framework MOA (MOA, 2010).

Tabela 1 - Informações sobre as bases utilizadas

Bases	Instâncias	Atributos	Tipo
Iris	150	4	real
Balance	625	4	real
Haberman	306	3	inteiro
Breast	699	9	inteiro
Dermatology	366	34	inteiro
Base1	1000	4	real

2-4 - Medidas de Avaliação de Agrupamento

A avaliação de grupos pelo critério da silhueta simplificada é dada por um cálculo que utiliza da dissimilaridade média dos objetos ao centro do seu grupo. A fórmula da silhueta simplificada é a mesma da fórmula da silhueta apresentada nas Equações 1 e 2, porém $a_{p,i}$ e $b_{p,i}$ são calculados como a distância do i-ésimo objeto ao centróide do grupo em questão.

3 - Resultados e discussão

Os testes realizados com o algoritmo *X-Means* foram feitos através da ferramenta *WEKA* (Hall et al., 2009) que é um programa de código livre feito em Java. Os testes realizados com o algoritmo *OMRk* foram feitos em uma implementação em Java. Primeiramente foram testadas todas as bases de dados no *X-Means* e no *OMRk* obtendo desta forma, o valor de K.

Para avaliar o desempenho de ambos os algoritmos, usou-se como medida de avaliação o método da silhueta simplificada (Vendramin et al.,2010).

Tabela 2 - Resultado da Avaliação do *X-Means*

Bases	Valor K	Silhueta
Iris	2	0.83970
Balance	2	0.51527
Haberman	2	0.51256
Breast	2	0.75462
Dermatology	2	0.65479
Base1	2	0.79679

Tabela 3 - Resultado da Avaliação do *OMRk*

Bases	Valor K	Silhueta
Iris	2	0.83970
Balance	8	0.55809
Haberman	4	0.60229
Breast	2	0.75462
Dermatology	2	0.65479
Base1	2	0.79679

Observando os dados descritos nas tabelas 2 e 3, nota-se que houve discrepância entre os algoritmos no valor de k em duas bases, *Balance* e *Haberman*. Na primeira, usando o algoritmo *X-Means* o valor de k é igual a 2, enquanto que no segundo o valor de k é igual a 8. Na base *Haberman* usando o algoritmo *X-Means* o valor de k é igual a 2, enquanto que no algoritmo *OMRk* o valor de k é igual a 4.

Quando diferentes valores para k são obtidos, analisa-se o valor da medida da silhueta. Nas bases citadas anteriormente é possível concluir que o algoritmo *OMRk* obteve melhor desempenho para ambas as bases, pois comparando-se o valor da medida da silhueta do *X-Means* e do *OMRk*, o valor do último é maior, ou seja, mais próximo de 1.0. Observa-se que o melhor desempenho obtido pelo algoritmo *OMRk* foram em bases com poucos atributos e várias instâncias.

4 - Considerações finais

A análise dos algoritmos, *X-Means* e *OMRk* inferiu que em bases com poucos atributos e várias instâncias, o *OMRk* obteve melhor desempenho, enquanto que para as outras bases os dois algoritmos obtiveram desempenhos semelhantes.

Levando em consideração o número reduzido de bases testadas, os resultados obtidos são prévios, necessitando de testes mais profundos que serão abordados em trabalhos futuros. Também estão sendo estudadas novas medidas de avaliação do agrupamento e o uso de testes estatísticos para validar os resultados.

5 - Referências

Tan, P., Steinbach, M. and Kumar, V. Introduction to Data Mining. Addison-Wesley. 2005.

Jain, A., Dubes, R. Algorithms for Clustering Data. Prentice-Hall. Englewood Cliffs. 1988.

Ochi, L. S., Dias, C. D. and Soares, S. S. F. Clusterização em Mineração de Dados. 2003.

Everitt, B. S., Landau S. and Leese, M. Cluster Analysis. Edward Arnold. 2001

Peleg, D. and Moore, A. W. X-means: extending K-means with efficient estimation of the number of clusters. School of Computer Science. 2000.

Arthur, D. and Vassilvitskii, S. K-means++: the advantages of careful seeding. Symposium on Discrete Algorithms. 2007.

Vendramin, L., Campello, R. J. G. B. and Hruschka, E. R. Relative Clustering Validity Criteria: A Comparative Overview. Statistical Analysis and Data Mining. A Wiley Company. Vol. 03. pp. 209 – 235. 2010.

Naldi, M. C., Campello, R. J. G. B., Hruschka, E. R. and Carvalho, A. C. P. L. F. Efficiency Issues of Evolutionary k-Means. Applied Soft Computing. Elsevier. Vol. 11. p. 1938 – 1952. 2011.

Hall, M., Frank, E., Holmes, G.,Pfahringer, B., Reutemann, P., Witten, I. H. The WEKA Data Mining Software: An Update. SIGKDD Explorations. Volume 11. Issue 1. 2009

Agradecimentos

Os autores são bolsistas do Programa de Educação Tutorial / MEC e agradecem ao mesmo pela oportunidade de desenvolver trabalhos científicos.