ISSN 1984-820X
Anais do CNMAC
v.3
PUBLICAÇÃO DA SBMAC
CNMAC2010
XXXIII CONGRESSO NACIONAL DE MATEMÁTICA APLICADA E COMPUTACIONAL

Uma Nova Forma de Calcular o Centro dos Clusters no Algoritmo Fuzzy C-Means

Rogério R. de Vargas,* **Benjamín R. C. Bedregal**,
Departamento de Matemática Aplicada e Informática, DIMAp, UFRN
59072-970, Natal, RN
E-mail: rogerio@ppgsc.ufrn.br, bedregal@dimap.ufrn.br

Resumo: *Agrupar dados é uma tarefa muito importante em mineração de dados, processamento de imagens e outros problemas de reconhecimento de padrões. O processo de agrupamento de dados Fuzzy podem ser demorados quando têm-se muitos objetos ou padrões para serem agrupados. Este artigo discute uma variante do algoritmo fuzzy c-means, o qual mostrou-se mais eficiente. Em vários testes realizados obteve resultados similares, mas com menor esforço computacional, diminuindo as iterações e consequentemente o tempo. Aqui apresentaremos, a modo de comparação, a utilização da base de dados IRIS pelos dois algoritmos: ckMeans e FCM. O algoritmo ckMeans permite reduzir o número de iterações e o tempo de processamento sem afetar na qualidade da partição. A redução é feita por calcular uma nova equação matemática para obter os centros dos clusters.*
Palavras-chave: *Agrupamento, fuzzy c-means, ckMeans*

1 Introdução

Agrupamento fuzzy é um método que pode capturar a incerteza em uma situação real. O agrupamento fuzzy pode obter um resultado robusto em relação de agrupamentos *hard* convencional.

Partições de agrupamento lida essencialmente com a tarefa de particionamento de um conjunto de entidades em um número de grupos homogêneos, com relação a uma medida de similaridade apropriada. Devido à natureza fuzzy de muitos problemas práticos, uma série de métodos de agrupamento fuzzy foram desenvolvidos após a teoria dos conjuntos fuzzy descrita de forma geral por [Zadeh 1965]. A principal diferença entre um agrupamento tradicional *hard* e um agrupamento fuzzy é que enquanto no agrupamento *hard* uma dado pertence a um único cluster, no agrupamento fuzzy um dado podem pertencer a mais de um grupo, mas com diferentes graus de pertinência [Nascimento 2000].

Estes métodos de agrupamento fuzzy têm sido amplamente aplicados em várias áreas, como processamento de imagem, a recuperação da informação, mineração de dados e outras [Carvalho 2007].

Técnicas de agrupamento podem serem divididas em métodos hierárquicos, separação e incremental. Métodos hierárquicos produzem grupos aninhados, métodos de separação são utilizados para a produção de grupos isolados e os métodos incrementais podem criar um novo grupo quando um novo registro é apresentado durante o processo de agrupamento [Jain et al. 1999].

Existem várias propostas diferentes de extensões para o algoritmo fuzzy c-means na literatura. Em [Zang 2009], por exemplo, é proposto uma nova métrica, utilizando a função exponencial para substituir a distância euclidiana no algoritmo fuzzy c-means (FCM). No artigo proposto por [Eschrich 2003] o objetivo principal é reduzir o tempo processamento e o número de iterações no algoritmo FCM, a redução é feita através da agregação de exemplos similares. No entanto, nenhum desses autores consideram uma nova forma de calcular os centros dos clusters.

Neste trabalho, propõe-se uma nova variante do algoritmo FCM, tendo como principal recurso à utilização de uma nova forma de calcular os centros dos clusters. A ideia é utilizar a matriz do grau de pertinência, a fim de obter uma matriz crisp que possibilite calcular os novos centros usando uma

*Bolsista de Doutorado Capes

ISSN 1984-820X
Anais do CNMAC v.3
PUBLICAÇÃO DA SBMAC
CNMAC2010 XXXIII CONGRESSO NACIONAL DE MATEMÁTICA APLICADA E COMPUTACIONAL

estratégia semelhante à do algoritmo k-means [MacQueen 1967]. Por este motivo, denominamos o algoritmo aqui proposto de ckMeans.

A seção 2 do artigo apresenta uma breve discussão do algoritmo fuzzy c-means. A seção 3 mostra a proposta do novo algoritmo proposto chamado ckMeans. Os experimentos são mostrados na seção 4 e finalmente, a seção 5 conclui o trabalho.

2 O algoritmo fuzzy c-means

Segundo [Zou et al. 2008], o algoritmo para agrupamento de dados fuzzy foi proposto por [Dunn 1974], e estendido por [Bezdek 1981]. A ideia basicamente é que o conjunto fuzzy $X = \{x_1, x_2, \ldots, x_n\}$ seja dividido em p clusters, μ_{ij} é o grau de pertinência da amostra x_i ao j-ésimo cluster e o resultado do agrupamento é expresso pelos graus de pertinência na matriz μ.

O algoritmo FCM tenta encontrar conjuntos nos dados, minimizando a função objetiva mostrada na equação (1):

$$J = \sum_{i=1}^{n} \sum_{j=1}^{p} \mu_{ij}^{m} d\left(x_i; c_j\right)^2 \tag{1}$$

onde:

- n é o número de dados;
- p é o número de clusters considerados no algoritmo o qual deve ser decidido antes da execução;
- $m > 1$ é o parâmetro da fuzzificação[1]. Usualmente, m esta no intervalo de $[1,25; 2]$ [Cox 2005];
- x_i um vetor de dados de treinamento, onde $i = 1, 2, \ldots, n$. Onde cada posição no vetor representa um atributo do dado;
- c_j é o centro de um agrupamento fuzzy ($j = 1, 2, \ldots, p$);
- $d\left(x_i; c_j\right)$ é a distância[2] entre x_i and c_j;

A entrada do algoritmo são os n dados, o número de clusters p e o valor de m. Os passos são:

1. Inicialize μ com um valor aleatório contínuo entre zero (nenhuma pertinência) e um (pertinência total), onde a soma das pertinências deve ser um;
2. Calcule o centro do *cluster* j da seguinte maneira:

$$c_j = \frac{\sum_{i=1}^{n} \mu_{ij}^{m} x_i}{\sum_{i=1}^{n} \mu_{ij}^{m}} \tag{2}$$

3. Calcule um valor inicial para J usando a equação (1);
4. Calcule a tabela da função de pertinência fuzzy μ conforme mostrado na equação (3)

$$\mu_{ij} = \frac{\left(\frac{1}{d(x_i;c_j)}\right)^{\frac{2}{m-1}}}{\sum_{k=1}^{p} \left(\frac{1}{d\left(x_i; c_k\right)}\right)^{\frac{2}{m-1}}} \tag{3}$$

[1] Considerando somente valores racionais para simplificar o cálculo das equações (1), (2) e (3). Uma vez que na prática, são usados m racionais.

[2] Quando são valores numéricos, normalmente é usado a distância Euclidiana

ISSN 1984-820X
Anais do CNMAC v.3
PUBLICAÇÃO DA SBMAC
CNMAC2010 XXXIII CONGRESSO NACIONAL DE MATEMÁTICA APLICADA E COMPUTACIONAL

5. Retornar à etapa 2 até que uma condição de parada seja alcançada.

Algumas condições de parada possíveis são:

- Um número de iterações pré-fixado for executado;
- O usuário informa um valor de parada $\epsilon > 0$, e se

$$d\left(J_U; J_A\right) \leq \epsilon$$

então pára, onde J_A é a função objetiva (equação (1)) calculada na iteração anterior e J_U é a função objetiva da última iteração.

3 O algoritmo ckMeans

O algoritmo k-means, proposto por [MacQueen 1967], é um método de particionamento (método não-hierárquico) que divide as observações dos dados em k clusters mutuamente exclusivos.

Esse algoritmo considera como centro de um grupo o seu centróide. O centróide de um grupo é definido como o vetor soma de todos os vetores correspondentes aos objetos associados a este grupo. Então, a tarefa do algoritmo k-means é minimizar a função objetivo correspondente à distância total entre os objetos e os centróides dos grupos aos quais esses objetos foram associados.

O algoritmo ckMeans proposto segue a mesma estrutura do algoritmo FCM, porém, a única alteração deu-se em como calcular o centro dos clusters, ou seja, o c_j.

Devido utilizar a mesma forma de calcular o centro de cada cluster do algoritmo k-means, nomeou-se o algoritmo proposto neste artigo de ckMeans.

Mas para isto, antes, é criada uma nova matriz μ, chamada de μCrisp contendo valores 1 ou 0. Cada linha dessa nova matriz tem 1 na posição do maior valor dessa linha na matriz μ e zero nas demais posições da linha. Quando uma coluna da matriz μCrisp, for toda com zeros, é atribuído o valor 1 na posição que corresponde ao maior valor dessa mesma coluna na matriz μ.

O algoritmo ckMeans retorna uma matriz μCrisp com valores em $\{0,\ 1\}$ conforme é mostrado na equação (4). Ou seja, μCrisp é a matriz enquanto μCrisp_{ij} é o conteúdo dessa matriz na posição (ij).

$$\mu Crisp_{ij} = \max\left(\left\lfloor \frac{\mu_{ij}}{\max\limits_{l=1}^{p} \mu_{il}} \right\rfloor, \left\lfloor \frac{\mu_{ij}}{\max\limits_{l=1}^{n} \mu_{lj}} \right\rfloor\right) \tag{4}$$

O primeiro argumento do max tem que cada dado tenha o valor 1 no cluster ao qual pertence com maior grau de pertinência e grau de pertinência zero nos demais. O segundo argumento tem por objetivo que o maior grau de cada coluna (cluster) seja 1. Para assim garantir que todo cluster tenha pelo menos um elemento. Dessa forma, em raras ocasiões, pode acontecer que uma linha tenha mais de uma valor 1 (o que não ocorre o algoritmo k-means original), mas como esta matriz é apenas auxiliar, não ocasionará qualquer transtorno.

Os passos do algoritmo para calcular o μCrisp_{ij}[3] é realizado da seguinte forma:

1. Leia μ;
2. Em cada linha encontrar o maior valor da matrix μ e atribuir 1 a essa mesma posição em μCrisp e zero nas restantes;
3. Armazenar em um vetor a quantidade de 1's que cada coluna de μCrisp possui.

[3]Pode ocorrer uma situação onde o resultado de μCrisp_{ij} não esteja completamente fiel à equação (4). O maior valor da coluna μ_{ij} terá 1 em μCrisp_{ij}.

Se uma coluna não tiver 1's marque sumariamente com 1 a posição onde está o maior valor.

Após calculada a matriz μCrisp calculam-se os novos centros dos clusters conforme a equação (5).

$$c_j = \frac{\Sigma_{i=1}^{n} x_i \mu Crisp_{ij}}{\Sigma_{i=1}^{n} \mu Crisp_{ij}} \tag{5}$$

O c_j é calculado pela somatória dos dados que pertencem ao cluster (de forma crisp) e dividido pela quantidade de objetos classificados como 1 na matriz μCrisp deste cluster.

4 Experimentos

Inicialmente implementou-se o algoritmo FCM (tradicional) em C++ (com a biblioteca C-XSC) baseada na implementação[4] de [deGruijter and McBratney 1988], disponível em `http://www.usyd.edu.au/agric/acpa/fkme/program.html`.

Todos os algoritmos aqui discutidos, foram executados e desenvolvidos em C++ (Versão 4.4.1) usando a biblioteca C-XSC (versão 2.2), usando um microcomputador Pentium IV, 3.0 GHz PC, com 512 KB cache e 1 GB de memória principal, usando o sistema operacional Linux (Kernel 2.6.31-20-generic, GNOME 2.28.1, Ubuntu 9.10). Os gráficos foram obtidos usando Gnuplot (versão 4.2 patchlevel 5).

4.1 A base de dados IRIS

A base de dados IRIS [Fisher 1936] é talvez o banco de dados mais utilizado na literatura no reconhecimento de padrões.

Testou-se o algoritmo ckMeans com o banco de dados IRIS (da UCI Repositório [Asuncion 2007]). Esta base de dados contém 3 séries de 50 instâncias, cada conjunto correspondente a uma das três classes da planta íris (Iris setosa, Iris Versicolour e Iris virginica).

Cada registro é descrito em termos de 4 variáveis numéricas (1. comprimento da sépala, 2. largura da sépala, 3. comprimento da pétala e 4. largura da pétala) todos os dados em centímetros.

Utilizou-se esta base de dados para discutir os resultados entre os algoritmos FCM e ckMeans.

4.2 Parâmetros de Inicialização

Os parâmetros de entrada são 150 dados (obtidos da base de dados) e estes dados referem-se à classe (1-50 Iris Setosa, 51-100 Iris Versicolour e 101-150 Iris Virginica). O número de clusters são 3, o valor de fuzziness é $m = 1.25$ e $\epsilon = 0.001$. Estes parâmetros foram usados em ambas as configurações dos algoritmos apresentados.

Os valores iniciais de μ_{ij} são números aleatórios. Usou-se os mesmos valores para inicializar os algoritmos FCM e ckMeans.

5 Resultados Comparativos

O resultado final da classificação de c_j nos algoritmos FCM e ckMeans é mostrado na tabela 1.

Observe que o centro dos clusters em todos os clusters são similares.

[4]De fato, essa implementação reporta exatamente os mesmo valores de [deGruijter and McBratney 1988].

ISSN 1984-820X
Anais do CNMAC
v.3
PUBLICAÇÃO DA SBMAC
CNMAC2010
XXXIII CONGRESSO NACIONAL DE MATEMÁTICA APLICADA E COMPUTACIONAL

Tabela 1: c_j Resultado com FCM e ckMeans

	comprimento da sépala		largura da sépala		comprimento da pétala		largura da pétala	
	FCM	**ckMeans**	**FCM**	**ckMeans**	**FCM**	**ckMeans**	**FCM**	**ckMeans**
Cluster 1	5.006	5.006	3.422	3.428	1.472	1.462	0.251	0.246
Cluster 2	6.866	6.870	3.085	3.086	5.733	5.746	2.083	2.089
Cluster 3	5.901	5.905	2.746	2.746	4.414	4.413	1.433	1.433

A tabela 2 mostra a média da diferença, entre os dois métodos. Para o cluster 1 é praticamente zero, para o cluster 2 é de 0,05 e cluster 3 é praticamente zero, uma vez que o desvio-padrão para o cluster 1 é praticamente zero, o cluster 2 é de 0,2 e o cluster é 3 é praticamente zero.

Tabela 2: Comparação dos Algoritmos

	Cluster 1	**Cluster 2**	**Cluster 3**
Média da diferença	0.0001	0.0591	0.0009
Desvio Padrão	0.0006	0.2427	0.0030

O número de instâncias classificadas usando os algoritmos FCM e ckMeans são os mesmos, conforme mostrado na tabela 3.

Tabela 3: Instâncias Agrupadas

Cluster	**Instâncias**	**Porcentagem**
Cluster 1	50	33.33%
Cluster 2	37	24.66%
Cluster 3	63	42%

A tabela 4 mostra a classificação dos dados em cada classe. O número de clusters classificados incorretamente são 17, que corresponde a 11.33%.

Tabela 4: Objetos Classificados

Cluster Atribuído	**Cluster 1**	**Cluster 2**	**Cluster 3**
Iris Setosa	50	0	0
Iris Versicolor	0	2	48
Iris Virginica	0	35	15

A tabela 5 mostra a quantidade de iterações, a média do tempo de processamento de cada iteração em segundos e o tempo total em segundos que o algoritmo levou para convergir.

Tabela 5: Performance

	FCM	**ckMeans**
Quantidade de iterações	18	13
Tempo médio de cada iteração	0.08	0.06
Tempo total para convergir	1.42	0.76

Observe que o algoritmo FCM convergiu com 18 iterações enquanto o algoritmo ckMeans convergiu com 13 iterações. Observe também que o tempo de processamento no algoritmo ckMeans foi menor do que o algoritmo FCM, com 0,76 e 1,42 segundos, respectivamente.

A tabela 6 mostra a função objetiva de J em ambos os algoritmos aqui discutidos. É mostrado na primeira coluna o valor da iteração de forma sequencial, na segunda coluna, o valor de J no algoritmo FCM e na terceira coluna mostra-se o valor de J no algoritmo ckMeans.

Tabela 6: Função Objetiva de J

Iteração	J in **FCM**	J in **ckMeans**
1	0.5029	0.5209
2	17.5569	29.2891
3	13.6251	0.0034
4	0.0717	0.0266
5	0.0098	0.0183
6	0.0234	0.0470
7	0.0322	0.0817
8	0.0440	0.0866
9	0.0697	0.1590
10	0.1060	0.0836
11	0.1250	0.0390
12	0.1003	0.0000
13	0.0505	**0.0000**
14	0.0192	
15	0.0065	
16	0.0021	
17	0.0006	
18	**0.0002**	

A primeira iteração tanto do algoritmo FCM quanto do algoritmo ckMeans o valor de J ficou em 0.5029. A segunda iteração houve um valor mais elevado para J (17.5569 no algoritmo FCM e 29.2891 no algoritmo ckMeans). O valor de J na terceira iteração no algoritmo FCM ficou em torno de 13, enquanto no algoritmo ckMeans ficou próximo de zero. Somente na quarta iteração o algoritmo FCM teve o J próximo de zero.

Na iteração 12 e 13 no algoritmo ckMeans o valor de J foi zero (considerando 50 casas decimais).

6 Conclusões

Neste trabalho foi proposto um novo método para calcular os centros dos clusters do algoritmo fuzzy c-means, reduzindo o tempo de processamento e o número de iterações. O algoritmo ckMeans fornece uma aceleração substancial perante a aplicação FCM tradicional.

Com as mesmas condições de software e hardware o algoritmo FCM usou quase o dobro de tempo do que o algoritmo ckMeans, obtendo resultados idênticos em termo de classificação.

Compreende-se que a expressão para o cálculo da função de objetiva e os centros dos cluster no algoritmo FCM é uma derivação matemática de uma função objetiva. Porém, não se tem essa preocupação no algoritmo ckMeans, a tabela 6 mostra que os valores de J (função objetivo) é um pouco menor no algoritmo ckMeans do que no algoritmo FCM, e portanto, na prática o objetivo de minimizar J também pode ser alcançado pelo algoritmo ckMeans.

Os experimentos mostram que a classificação do grau de pertinência com o algoritmo ckMeans em relação ao cluster é similar do que com o algoritmo FCM (considerando o caso estudado). O número de

iterações em relação à convergência em todos os cluster usando o algoritmo FCM foram 18 iterações. No entanto, usando o algoritmo ckMeans obteve-se a convergência com 13 iterações.

Como trabalho futuro a intenção é aplicar o algoritmo ckMeans à outras bases de dados e comparar com outras variantes do algoritmo FCM.

Referências

[Asuncion 2007] Asuncion, A., Newman, D (2007). UCI Machine Learning Repository [`http://www.ics.uci.edu/~mlearn/MLRepository.html`]. *Irvine, CA: University of California, Department of Information and Computer Science.*

[Bezdek 1981] Bezdek, J. (1981). *Pattern Recognition with Fuzzy Objective Function Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA.

[Carvalho 2007] Carvalho, F. (2007). Fuzzy C-Means Clustering Methods for Symbolic Interval Data. *Pattern Recogn. Lett.*, 28(4):423–437.

[Cox 2005] Cox, E. (2005). Fuzzy Modelling and Genetic Algorithms for Data Mining and Exploration. Morgan Kaufmann, 2005.

[deGruijter and McBratney 1988] deGruijter, J. and McBratney (1988). A modified fuzzy K-means for predictive classification. *Classification and Related Methods of Data Analysis*, H.H. Bock, ed., Elsevier Science, Amsterdam, 1988, pp. 97-104.

[Dunn 1974] Dunn, J. (1974). A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. *Journal of Cybernetics*, 3:32–57.

[Eschrich 2003] Eschrich, S., Ke, J., Hall, L. and Goldgof, D. (2003) Fast Accurate Fuzzy Clustering Through Data Reduction. IEEE Transactions on Fuzzy Systems, vol. 11, pp. 262-270, 2003.

[Fisher 1936] Fisher, R. (1936) The Use of Multiple Measurements in Taxonomic Problems. Annals Eugen., vol. 7, pp. 179-188.

[Hofschuster and Kramer 2003] Hofschuster, W. and Kramer, W. (2003). C-XSC 2.0 - A C++ Library for eXtended Scientific Computing. In *Numerical Software with Result Verification: International Dagstuhl Seminar, Dagstuhl*, pages 15–35. Springer.

[Jain et al. 1999] Jain, A. K., Murty, M. N., and Flynn, P. J. (1999). Data Clustering: a Review. *ACM Comput. Surv.*, 31(3):264–323.

[MacQueen 1967] MacQueen, J. (1967). Some Methods for Classification and Analysis of Multivariate Observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, Berkeley, CA. University of California Press.

[Nascimento 2000] Nascimento, S., Mirkin, B., Moura-Pires, F. (2000) A Fuzzy Clustering Model of Data and Fuzzy C-Means. Fuzzy Systems, 2000. FUZZ IEEE 2000. The Ninth IEEE International Conference on vol.1, no., pp.302-307 vol.1, 7-10

[Zadeh 1965] Zadeh, L. (1965). Fuzzy Sets. *Information and Control*, 8, pp, 409–416.

[Zang 2009] Zang, K., Li, B., Xu, J., and Wu, L. (2009). New Modification of Fuzzy c-Means Clustering Algorithm. Fuzzy Information and Engineering vol:1, pages 448–445.

[Zou et al. 2008] Zou, K., Wang, Z., and Hu, M. (2008). An New Initialization Method for Fuzzy C-means Algorithm. *Fuzzy Optimization and Decision Making*, 7(4):409–416.