Similarity Based Fuzzy and Possibilistic c-means Algorithm

Chunhui Zhang[1] Yiming Zhou[2] Trevor Martin[3]

[1,2]School of Computer Science and Technology
Beihang University
Beijing,China,100191
prospring@163.com, zhouyiming@buaa.edu.cn
[3]Deptartment of Engineering Maths
University of Bristol
Bristol, UK, BS8 1TR
Trevor.Martin@bris.ac.uk

Abstract

A similarity based fuzzy and possibilistic c-means algorithm called *SFPCM* is presented in this paper. It is derived from original fuzzy and possibilistic c-means algorithm(*FPCM*) which was proposed by Bezdek. The difference between the two algorithms is that the proposed SFPCM algorithm processes relational data, and the original FPCM algorithm processes propositional data. Experiments are performed on 22 data sets from the UCI repository to compare SFPCM with FPCM. The results show that these two algorithms can generate similar results on the same data sets. SFPCM performs a little better than FPCM in the sense of classification accuracy, and it also converges more quickly than FPCM on these data sets.

Keywords: similarity, fuzzy clustering

1. Introduction

Clustering as a type of machine learning methods has been applied to many theoretical and practical problems. There are mainly two kinds of clustering algorithms, one is hard clustering, the other is fuzzy clustering. Hard clustering generates crisp clusters, and fuzzy clustering generates fuzzy clusters, which don't have crisp boundaries. Fuzzy clustering methods are good at dealing with "might be" situations in the real world.

There is a series of famous fuzzy clustering methods, such as fuzzy c-means(FCM)[9], possibilistic c-means(PCM)[10], fuzzy-possibilistic c-means model(FPCM)[7], possibilistic fuzzy c-means clustering algorithm(PFCM)[8]. There are also many derived methods. However, most fuzzy clustering methods process propositional data, only a small portion can process relational data. Relational data are typically n^2 pair-wise similarity or dissimilarity matrix between all pairs of objects in a dataset. This kind of fuzzy clustering algorithms are called relational fuzzy clustering algorithms. Relational fuzzy c-means (RFCM) [3], non-Euclidean RFCM (NERFCM) [4], (Gaussian) Kernelized Non-Euclidean Relational

Fuzzy c-Means[11], fuzzy relational data clustering (FRC) [5], robust RFCM (R-RFCM) [5], robust NERFCM (R-NERFCM)[6],robust-FRC (R-FRC) [5] and SPCM[2] are some relational fuzzy clustering algorithms. They are mostly derived from FCM, and SPCM is derived from PCM.

The proposed SFPCM is derived from FPCM and can process relational data. The initial purpose of the SFPCM is to cluster a set of terms, to find which are domain terms and which are noisy terms. FPCM is a powerful fuzzy clustering method, we want to use it to perform this task. Because only similarity matrix on terms can be obtained, and FPCM cannot handle relational data, we need to modify it. However, we don't list the experiments to cluster domain terms by SFPCM in this paper because of pages limitation. We just concentrate on experiments on 22 data sets from UCI[12] to compare FPCM and SFPCM.

The rest of this paper is organized as follows. Section 2 describes the proposed SFPCM algorithm; Section 3 shows some experiments on 22 data sets from the UCI machine learning repository to compare FPCM and the proposed SFPCM; Section 4 makes conclusion and talks about future research directions.

2. SFPCM algorithm

The original FPCM proposed by Bezdek[7] is based on distance computation, and the input of the algorithm is propositional data set. We try to modify FPCM to process relational data and call the new algorithm SFPCM.

SFPCM algorithm is an optimization problem solved by iterative processes.

The objective function is

$$J_{max} = \sum_{i=1}^{n}\sum_{j=1}^{c}(u_{ij}^{m} + t_{ij}^{\eta})(sim(x_i, c_j))^2 \quad (1)$$

where $1 < m < \infty$ and $1 < \eta < \infty$. The whole algorithm is as follows.

Input:($M_{n\times n}$,c,m,η,ε)

Where M is similarity matrix on data set X. $X = \{x_1, x_2, x_2, \ldots, x_n\}$, each object can be represented by a unit vector of length n. For example

$x_1 = (1, 0, 0, \ldots, 0)$

$x_2 = (0, 1, 0, \ldots, 0)$

...

$x_n = (0, \ldots, 0, 1)$

The entry $m_{ij} \in M$ is the similarity between x_i and x_j, and can also be written as $sim(x_i, x_j)$. c is predefined cluster number. m and η are two fuzzy exponents. ε is an extreme small positive number which controls the termination of the algorithm.

Output: (U,T,V). U is membership matrix, T is possibility matrix, and V is cluster center matrix.

Initialization: We first select c objects randomly from X as initial cluster centers, we call them seeds. $seeds_j \in X, \quad j = 1 \ldots c$ is the jth initial cluster center. And the initial u_{ij}^0, t_{ij}^0 and v_j^0 are computed as:

$$u_{ij}^0 = (\sum_{k=1}^{c}(\frac{sim(x_i, seeds_k)}{sim(x_i, seeds_j)})^{\frac{2}{m-1}})^{-1} = (\sum_{k=1}^{c}(\frac{m_{i,seeds_k}}{m_{i,seeds_j}})^{\frac{2}{m-1}})^{-1} \quad (2)$$

$$t_{ij}^0 = (\sum_{k=1}^{n}(\frac{sim(x_k, seeds_j)}{sim(x_i, seeds_j)})^{\frac{2}{\eta-1}})^{-1} = (\sum_{k=1}^{n}(\frac{m_{k,seeds_j}}{m_{i,seeds_j}})^{\frac{2}{\eta-1}})^{-1} \quad (3)$$

if $m_{i,seeds_j} = 1$, which means x_i is the jth initial cluster center, u_{ij} and t_{ij} will

be corrected to 1.

$$v_j^0 = \frac{\sum_{k=1}^{n}(u_{kj}^m + t_{kj}^\eta) \times x_k}{\sum_{k=1}^{n}(u_{kj}^m + t_{kj}^\eta)} \quad (4)$$

Iteration: In the following iterations, U,V,T are updated.

$$u_{ij} = (\sum_{k=1}^{c}(\frac{sim(x_i, c_k)}{sim(x_i, c_j)})^{\frac{2}{m-1}})^{-1} \quad (5)$$

$$t_{ij} = (\sum_{k=1}^{n}(\frac{sim(x_k, c_j)}{sim(x_i, c_j)})^{\frac{2}{\eta-1}})^{-1} \quad (6)$$

$$v_j = \frac{\sum_{k=1}^{n}(u_{kj}^m + t_{kj}^\eta)x_k}{\sum_{k=1}^{n}(u_{kj}^m + t_{kj}^\eta)} \quad (7)$$

In the above equations, $1 \leq i \leq n$, $1 \leq j \leq c$. In equation5 and equation6 , $sim(x_i, c_j)$ is defined as:

$$sim(x_i, c_j) = \frac{\sum_{k=1}^{n}(u_{kj}^m + t_{kj}^\eta) \times m_{i,k}}{\sum_{k=1}^{n}(u_{kj}^m + t_{kj}^\eta)} \quad (8)$$

m_{il} is the similarity between x_i and x_l. u_{lj} is the membership degree of x_l to c_j and t_{lj} is the possibility of x_l to c_j.

When $|V - V_{old}| < \varepsilon$, the iteration process terminates.

SFPCM satisfies the constraints:

$$\sum_{j=1}^{c} u_{ij} = 1 \quad i = 1, \ldots, n \quad (9)$$

and

$$\sum_{i=1}^{n} t_{ij} = 1 \quad j = 1, \ldots, c \quad (10)$$

3. Experiments

We have carried out experiments on 22 data sets selected from UCI[12] to compare SFPCM and FPCM. If the original data set has separate training and testing sets, the training set is used, otherwise, the whole set is used. Besides, if the original data set is categorical, we change it to numerical data set by replacing each category of a categorical attribute with a number. You can refer to the UCI repository website[12] for more detailed information about the data sets we used.

The similarity between two objects is computed by the reciprocal of their Euclidean distance. And the similarities are normalized. Because SFPCM is a derivation of FPCM, it should generate similar results with FPCM on same data sets. We use same initial clusters for both algorithms, and ε is set to be 0.001 for both algorithms. We use the following method to compute the relatedness of the two fuzzy partitions(resultant clusters) obtained by FPCM and SFPCM.

For each pair of objects (o_1, o_2) of a data set O, there are four cases in the two partitions.

(1) o_1 and o_2 belong to the same cluster in the first partition, they also belong the same cluster in the second partition.

(2) o_1 and o_2 belong to the same cluster in the first partition, but they belong to different clusters in the second partition.

(3) o_1 and o_2 belong to different clusters in the first partition, and they belong to the same cluster in the second partition.

(4) o_1 and o_2 belong to different clusters in the first partition, and they also belong to different clusters in the second partition.

Then for all pairs of objects in O, we count the number of each case, which are n_1,n_2,n_3 and n_4. The relatedness or similarity of the two partitions is computed as:

$$R = \frac{n_1 + n_4}{n_1 + n_2 + n_3 + n_4} \qquad (11)$$

A higher R means better similarity.

The results are shown in Table1, the second column is R value, the third and fourth columns are classification accuracy of FPCM and SFPCM respectively. In order to compute accuracy, we first defuzzy the resultant fuzzy partition to a crisp partition. If an object has the maximal membership degree and possibility value for a cluster than other clusters, then this object will be assigned to the cluster. For example, if an object with maximal membership for cluster 1, but maximal possibility value for cluster 2, then it won't be assigned to any cluster. Then we map the resultant clusters to real classes of the data set. If most objects of a cluster are from Class i, then this cluster will be labeled to Class i. Finally, classification accuracy can be computed.

From Table1, we can see that, the two algorithms obtain same accuracy for *iris* and *lung*. They obtain same partition for *iris*, because the R value is one. For data sets *balance*, *australian* and *vehicleaa*, FPCM's accuracy is higher than SFPCM's. And for other seventeen data sets, the accuracy of SFPCM is higher, which are labeled in bold type in Table1. Among these data sets, there are six data sets on which the accuracy of SFPCM exceeds 10% than FPCM's. The most prominent one is *thyroid*, the accuracy of SFPCM is greater than FPCM's by 31.16%

Table2 shows that SFPCM converges more quickly than FPCM nearly on all data sets except *hepatitis* and *lung*. The reason may lie in that for SFPCM, $|V - V_{old}|$ is only affected by U and T, but for FPCM, $|V - V_{old}|$ is affected by U,T and X. Sometimes, even U and T change little, the absolute value between V and V_{old} is bigger than the threshold ε. We then do experiments using a bigger $\varepsilon = 0.01$ to see its effect on accuracy and iteration times. We find that the iteration times of FPCM reduce remarkably but are still bigger than SFPCM's. For example, on *breast*, the iteration times of FPCM is 37 and SFPCM's is 7; and on *bupa*, their iteration times are 35 and 7 respectively. But the accuracy of SFPCM on *breast* is 0.8283, and better than FPCM's accuracy which is 0.6624. And on *bupa*, FPCM's accuracy is 0.4406 and SFPCM's accuracy is 0.6237. SFPCM doesn't lead to less classification accuracy in spite of much fewer iteration times.

We also use Gaussian kernel function

Data set	R	FPCM	SFPCM
iris	1	0.96	0.96
glass	0.8284	0.4346	**0.5981**
soybean	0.9722	0.9574	**1**
waveform	0.7924	0.5518	**0.6690**
breast	0.8281	0.6738	**0.8283**
cleveland	0.9267	0.4698	**0.5570**
pima	0.8548	0.5560	**0.5638**
hepatitis	0.8188	0.5430	**0.6490**
balance	0.8680	0.6352	0.6176
ionosphere	0.8249	0.5812	**0.6439**
bupa	0.8531	0.4363	**0.6237**
thyroid	0.6982	0.5442	**0.8558**
wine	0.9763	0.8989	**0.9157**
hayes	0.6073	0.3788	**0.4394**
lenses	0.6196	0.6667	**0.7083**
lung	0.6553	0.6296	0.6296
spambase	0.6478	0.5082	**0.5838**
australian	0.4942	0.7768	0.5551
heart	0.7252	0.7074	**0.7667**
satimage	0.9398	0.7024	0.6859
segment	0.9608	0.6229	**0.6433**
vehicleaa	0.8277	0.4043	0.3723

Table 1: Relatedness and Accuracy with $\varepsilon = 10^{-3}$

data set	FPCM	SFPCM
iris	8	**5**
glass	39	**17**
soybean	13	**7**
waveform	14	**2**
breast	100	**7**
cleveland	48	**31**
pima	10	**3**
hepatitis	12	16
balance	50	**5**
ionosphere	7	**5**
bupa	44	**5**
thyroid	36	**16**
wine	9	**5**
hayes	61	**9**
lenses	47	**30**
lung	19	20
spambase	9	**2**
australian	18	**2**
heart	37	**8**
satimage	19	**6**
segment	20	**9**
vehicleaa	25	**21**

Table 2: Iteration times with $\varepsilon = 10^{-3}$

Data set	Accuracy	Iteration
iris	0.9467	**5**
glass	**0.4393**	**22**
soybean	**1**	**3**
waveform	**0.7268**	**2**
breast	0.4835	**14**
cleveland	**0.5805**	**11**
pima	**0.5911**	**3**
hepatitis	**0.6026**	**7**
balance	0.6128	**3**
ionosphere	**0.6610**	**4**
bupa	0.4206	**9**
thyroid	**0.5721**	**8**
wine	**0.9231**	**5**
hayes	**0.4015**	**31**
lenses	0.6667	**8**
lung	0.6296	**6**
spambase	0.4184	**1**
australian	0.5551	**1**
heart	0.6741	**14**
satimage	**0.7039**	**5**
segment	**0.6649**	**7**
vehicleaa	0.3085	25

Table 3: Results for SFPCM using Gaussian Kernel Function and $\varepsilon = 10^{-3}$

to compute similarity matrix for data sets. Accuracy and iteration times of SFPCM are shown in Table3. We also label better accuracy and smaller iteration times in comparison with FPCM in bold type. It can be seen that SFPCM's accuracy for twelve data sets reduce, three unchange, and seven other data sets improved. However, the big and small relationship between the accuracy of FPCM and SFPCM doesn't change much from Table1.

4. Conclusion and future work

In this paper, we derive an algorithm called SFPCM from FPCM, which can process relational data. We do experiments on 22 data sets from UCI repository to test SFPCM. For nearly all data sets, SFPCM converges more quickly than FPCM. But fewer iteration times don't lead to worse accuracy for SFPCM. The quick convergence is an advantage of SFPCM and is especially useful for processing large data sets in practical applications.

However, there are still many problems to explore in the future. For example, the performance of SFPCM should be tested on more data sets with different choices of input parameters. We also need to analyze which type of data is suited to use SFPCM. The impact of different similarity measures on clustering result of SFPCM should also be studied.

References

[1] G.Castellano, A.M. Fanelli, C.Mencar and M.A.Torsello. Similarity-based Fuzzy clustering for user profiling. *International Conferences on Web Intelligence and Intelligent Agent Technology*. 2007 IEEE/WIC/ACM

[2] V.S.Tseng and C. Kao. A Novel Similarity-Based Fuzzy Clustering Algorithm by Integrating PCM and Mountain Method. *IEEE TRANSACTIONS ON FUZZY SYSTEMS*, 15:1188-1196. DECEMBER,2007

[3] R.J.Hathaway, J.W.Davenport, and J.C.Bezdek. Relational duals of the c-Means clustering algorithms. *Pattern Recognition*, vol. 22, no.2, pp. 205-212, 1989.

[4] R. J. Hathaway and J. C. Bezdek, NerF c-means: Non-Euclidean relational fuzzy clustering, *Pattern Recognition*, vol.27, no.3, pp.429-437, 1994.

[5] R. N. Dave and S. Sen, Robust fuzzy clustering of relational data, *IEEE Trans on Fuzzy Systems*, vol. 10, no. 6, pp. 713-727, 2002.

[6] J. W. Davenport and R. J. Hathaway, Possibilistic c-means clustering for relational data, in *Proceedings of the 1st International Conference on Neural, Parallel & Scientific Computations*, 1995, vol.1, pp.139-142.

[7] N.R.Pal, and J.C.Bezdek. A mixed c-means clustering model. In *IEEE Int.Conf.Fuzzy Systems*, pages 11-21, Spain,1997.

[8] N.R.Pal, K.Pal, J.M.Keller and J.C.Bezdek. A Possibilistic Fuzzy c-Means Clustering Algorithm. *IEEE TRANSACTIONS ON FUZZY SYSTEMS* 13:517-530. AUGUST 2005.

[9] J. C. Bezdek. Pattern Recognition With Fuzzy Objective Function Algorithms. New York: Plenum, 1981.

[10] R. Krishnapuram and J. Keller. A possibilistic approach to clustering. *IEEE Trans. Fuzzy Syst.*, vol. 1, no. 2, pp. 98-110, Apr. 1993.

[11] R.J.Hathaway, J.M. Huband, and J.C.Bezdek. A Kernelized Non-Euclidean Relational Fuzzy c-Means Algorithm. *Proceedings of FUZZ-IEEE 2005* IEEE Press, Piscataway, N.J., pp. 414-419, 2005.

[12] http://mlearn.ics.uci.edu/MLRepository.html