Scalable Clustering by Truncated Fuzzy *c*-means

Guojun Gan, PhD, FSA

Department of Mathematics
University of Connecticut
Storrs, CT, 06269, USA

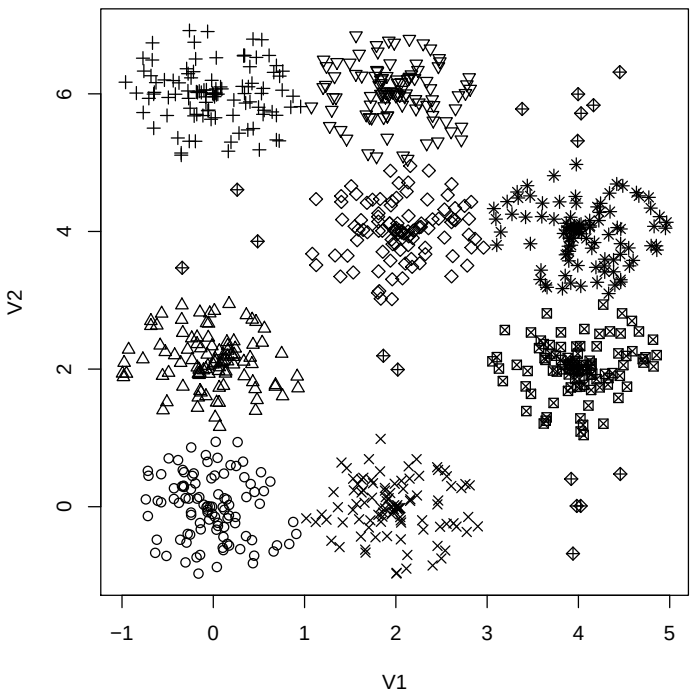
October 8, 2019

Outline

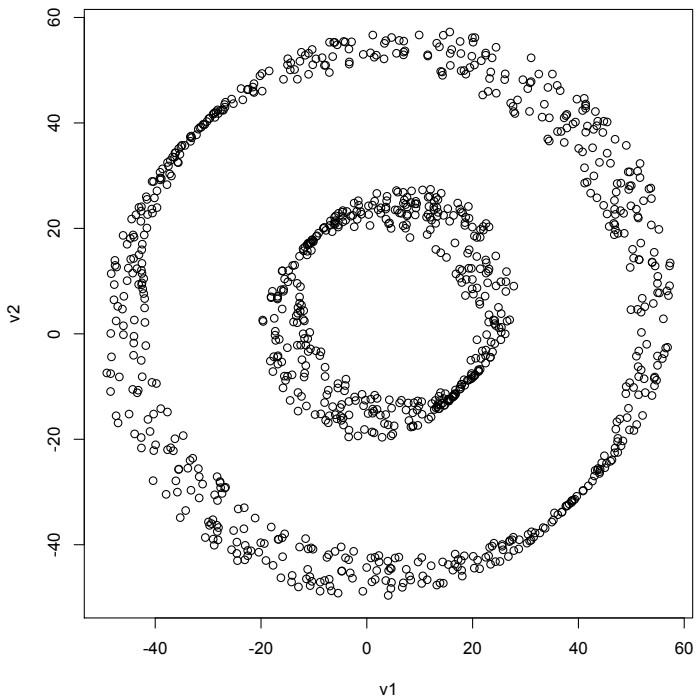
- A challenge in data clustering
- The TFCM algorithm
- Some numerical results

Data Clustering

Data clustering refers to a process of dividing a set of items into homogeneous groups or clusters such that items in the same cluster are similar to each other and items from different clusters are distinct (Gan et al., 2007; Gan, 2011).



(a)



(b)

Clustering Algorithms

Clustering algorithms can be divided into two groups:

- hard clustering algorithms: each item is assigned to one and only one cluster;
- fuzzy clustering algorithms: each item can be assigned to one or more clusters with some degrees of membership

Examples of hard clustering algorithms include the *k*-means algorithm (Macqueen, 1967). The FCM (Fuzzy *c*-means) algorithm (Dunn, 1973; Bezdek, 1981; Bezdek et al., 1984) is a popular fuzzy clustering algorithm.

Most algorithms are not efficient for dividing a large dataset into many clusters

- Divide millions of web pages into a thousand categories (Broder et al., 2014)
- Divide a portfolio of hundreds of thousands insurance policies into a thousand clusters in order to select a thousand representative policies (Gan, 2013; Gan and Lin, 2015)

FCM is slow for dividing a large dataset into many clusters

- The time complexity of the FCM algorithm is $O(nk^2d)$.
- Most existing improvements focus on large n.

Kolen and Hutcheson (2002) proposed a modified version of the FCM algorithm by eliminating the need to store the fuzzy partition matrix U. In the modified version, updating the cluster centers and updating the fuzzy memberships are combined into a single step. The algorithm proposed by Kolen and Hutcheson (2002) reduces the time complexity of the FCM algorithm from $O(nk^2d)$ to $O(nkd)$.

The truncated fuzzy c-means (TFCM) algorithm

- A subset of the full fuzzy partition matrix is stored
- The number of distance calculations at each iteration is reduced
- When k is large, a data point belongs to only a few clusters with high degrees of membership
- We can ignore the clusters with low degrees of membership while preserving the overall quality of the clustering.

The TFCM algorithm I

Let $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ be a dataset containing n points. Let k be the desired number of clusters. A fuzzy partition matrix $U = (u_{il})_{n \times k}$ of dividing X into k clusters is a $n \times k$ matrix that satisfies the following conditions

$$u_{il} \in [0, 1], \quad i = 1, 2, \ldots, n, \; l = 1, 2, \ldots, k,$$

and

$$\sum_{l=1}^{k} u_{il} = 1, \quad i = 1, 2, \ldots, n.$$

Let T be an integer such that $1 \leq T \leq k$. Let $\mathcal{U}_T$ be the set of fuzzy partition matrices U such that each row of U has at most T nonzero entries. In other words, $U \in \mathcal{U}_T$ if U is a fuzzy partition matrix such that for each $i = 1, 2, \ldots, n$,

$$|\{l : u_{il} > 0\}| \leq T, \tag{1}$$

The TFCM algorithm II

where $|\cdot|$ denote the number of elements in a set.
Then the objective function of the TFCM algorithm is defined as

$$P(U, Z) = \sum_{i=1}^{n} \sum_{l=1}^{k} u_{il}^{\alpha} \left(\|\mathbf{x}_i - \mathbf{z}_l\|^2 + \epsilon \right), \tag{2}$$

where $\alpha > 1$ is the fuzzifier, $U \in \mathcal{U}_T$, $Z = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_k\}$ is a set of cluster centers, $\|\cdot\|$ is the L^2-norm or Euclidean distance, and ϵ is a small positive number used to prevent division by zero. Let $I_i = \{l : u_{il} > 0\}$ for $i = 1, 2, \ldots, n$. Then we can rewrite the objective function (2) as

$$P(U, Z) = \sum_{i=1}^{n} \sum_{l \in I_i} u_{il}^{\alpha} \left(\|\mathbf{x}_i - \mathbf{z}_l\|^2 + \epsilon \right). \tag{3}$$

The TFCM algorithm III

Theorem

Given a set of centers Z. The fuzzy partition matrix $U \in \mathcal{U}_T$ that minimizes the objective function (2) *is given by*

$$u_{il} = \frac{\left(\|\mathbf{x}_i - \mathbf{z}_l\|^2 + \epsilon\right)^{-\frac{1}{\alpha-1}}}{\sum_{s\in I_i}\left(\|\mathbf{x}_i - \mathbf{z}_s\|^2 + \epsilon\right)^{-\frac{1}{\alpha-1}}}, \quad 1 \leq i \leq n,\ l \in I_i, \qquad (4)$$

where I_i is the set of indices of the T centers that are closest to $\mathbf{x}_i$, i.e.,

$$I_i = \{I_1, I_2, \ldots, I_T\} \qquad (5)$$

with $(I_1, I_2, \ldots, I_k)$ being a permutation of $(1, 2, \ldots, k)$ such that

$$\|\mathbf{x}_i - \mathbf{z}_{I_1}\| \leq \|\mathbf{x}_i - \mathbf{z}_{I_2}\| \leq \cdots \leq \|\mathbf{x}_i - \mathbf{z}_{I_k}\|.$$

The TFCM algorithm IV

Theorem

Given a fuzzy partition matrix $U \in \mathcal{U}_T$. The set of centers Z that minimizes the objective function (2) is given by

$$z_{lj} = \frac{\sum_{i=1}^{n} u_{il}^{\alpha} x_{ij}}{\sum_{i=1}^{n} u_{il}^{\alpha}} = \frac{\sum_{i \in C_l} u_{il}^{\alpha} x_{ij}}{\sum_{i \in C_l} u_{il}^{\alpha}}, \tag{6}$$

for $l = 1, 2, \ldots, k$ and $j = 1, 2, \ldots, d$, where d is the number of features, z_{lj} is the jth component of $\mathbf{z}_l$, and $C_l = \{i : u_{il} > 0\}$.

The TFCM algorithm V

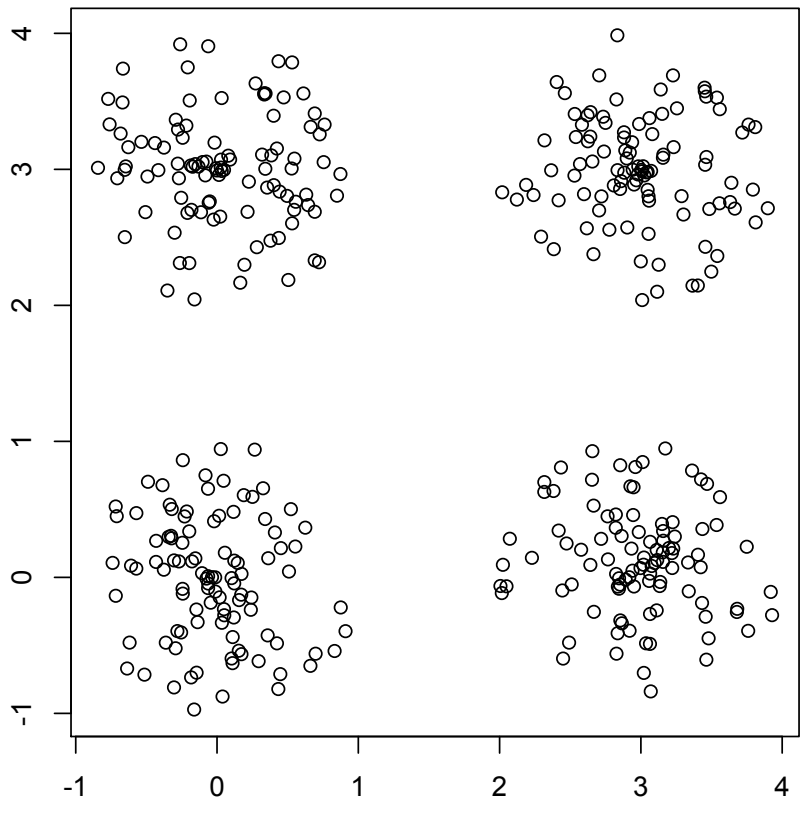
Input: $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, k, T, δ, N_{max}, α
Output: U, Z

1 Initialize the set of cluster centers $Z^{(0)}$ by selecting k data points from X randomly;
2 **for** $i = 1$ *to* n **do**
3 Calculate the distance between $\mathbf{x}_i$ and all k centers;
4 Let I_i be the subset of $\{1, 2, \ldots, k\}$ such that the corresponding T centers are closest to $\mathbf{x}_i$;
5 Update the weights $u_{il}^{(0)}$ for $l \in I_i$ according to Equation (4);
6 **end**
7 $s \leftarrow 0$;
8 $P^{(0)} \leftarrow 0$;

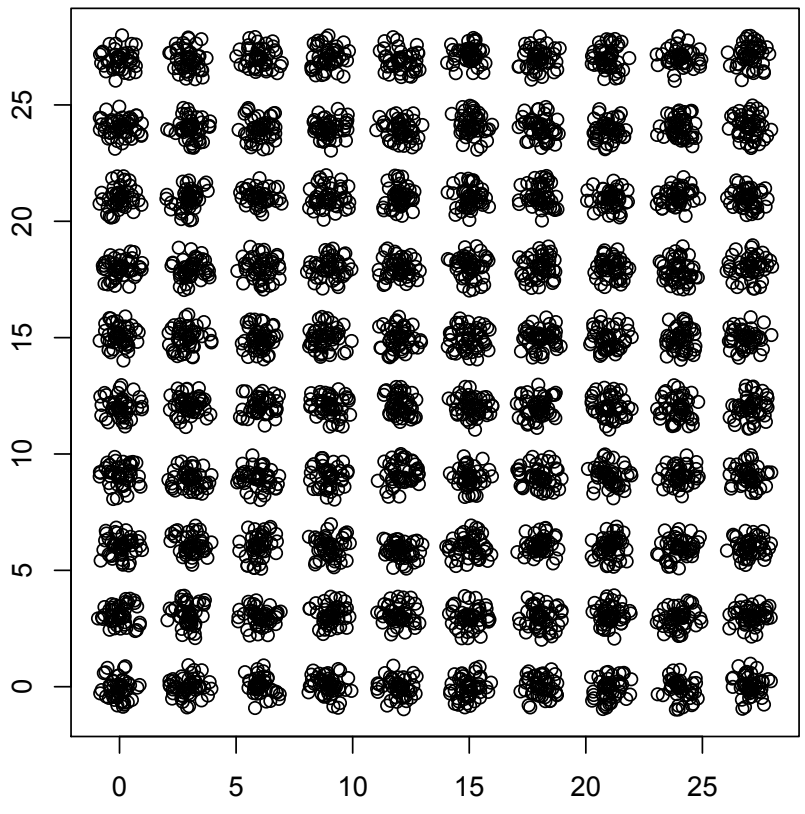
The TFCM algorithm VI

while *True* **do**
  Update the set of cluster centers $Z^{(s+1)}$ according to Equation (6);
  for $i = 1$ *to* n **do**
    Select T centers with indices in $\{1, 2, \ldots, k\}/I_i$ randomly;
    Calculate the distance between $\mathbf{x}_i$ and centers with indices in $I_i \cup J_i$;
    Update I_i with the indices of the T centers that are closest to $\mathbf{x}_i$;
    Update the weights $u_{il}^{(s+1)}$ for $l \in I_i$ according to Equation (4);
  end
  $P^{(s+1)} \leftarrow P\left(U^{(s+1)}, Z^{(s+1)}\right)$;
  if $\left|P^{(s+1)} - P^{(s)}\right| < \delta$ *or* $s \geq N_{max}$ **then**
    Break;
  end
  $s \leftarrow s + 1$;
end

Two synthetic datasets



(a)



(b)

Runtime and accuracy on the first synthetic dataset

k	Runtime	R_c
2	0.103(0.139)	0.433(0)
4	0.058(0.114)	1(0)
8	0.106(0.154)	0.682(0.023)

(a) TFCM

k	Runtime	R_c
2	0.044(0.061)	0.498(0)
4	0.05(0.058)	1(0)
8	0.176(0.143)	0.726(0.038)

(b) FCM

Runtime and accuracy on the second synthetic dataset

k	Runtime	R_c
50	6.869(6.65)	0.502(0.007)
100	5.084(1.97)	0.797(0.029)
200	20.639(7.879)	0.776(0.008)

(a) TFCM with $T = 3$

k	Runtime	R_c
50	5.269(1.574)	0.483(0.007)
100	4.348(1.887)	0.848(0.03)
200	20.184(9.307)	0.777(0.008)

(b) TFCM with $T = 6$

k	Runtime	R_c
50	71.877(16.729)	0.526(0.006)
100	26.341(18.1)	0.819(0.025)
200	53.683(26.543)	0.799(0.015)

(c) FCM

A real dataset

The variable annuity dataset was simulated by a Java program (Gan, 2015). The dataset contains 10,000 variable annuity contracts. The original dataset contains categorical variables. We converted the categorical variables into binary dummy variables and normalized all numerical variables to the interval [0,1]. The resulting dataset has 22 numerical features.

$$WSS = \sum_{l=1}^{k} \sum_{\mathbf{x} \in C_l} \sum_{j=1}^{d} \left(x_j - z_{lj} \right)^2 . \tag{7}$$

Runtime and accuracy on the real dataset

k	Runtime	*WSS*
100	6.417(1.9)	944.636(14.574)
200	16.167(5.565)	735.001(6.37)

(a) TFCM with $T = 3$

k	Runtime	*WSS*
100	16.734(5.133)	930.14(15.614)
200	31.871(19.216)	721.291(5.3)

(b) TFCM with $T = 6$

k	Runtime	*WSS*
100	71.185(22.023)	958.137(15.234)
200	87.918(22.641)	740.548(6.688)

(c) TFCM with $T = 12$

k	Runtime	*WSS*
100	164.02(57.612)	994.111(18.829)
200	219.695(51.104)	783.113(7.156)

(d) TFCM with $T = 23$

k	Runtime	*WSS*
100	280.137(70.577)	1049.864(24.202)
200	339.216(80.694)	822.988(8.866)

(e) TFCM with $T = 46$

k	Runtime	*WSS*
100	597.828(193.2)	895.205(16.264)
200	756.378(382.952)	697.841(6.736)

(f) FCM

Future work

- The initialization step calculates *nk* distances
- Combining the membership updating step and the center updating step by using the technique introduced by Kolen and Hutcheson (2002)
- Other efficient ways to divide a large dataset into many clusters

References I

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

Bezdek, J. C., Ehrlich, R., and Full, W. (1984). FCM: The fuzzy c-means clustering algorithm. *Computers & Geosciences*, 10(2-3):191 – 203.

Broder, A., Garcia-Pueyo, L., Josifovski, V., Vassilvitskii, S., and Venkatesan, S. (2014). Scalable k-means by ranked retrieval. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, WSDM '14, pages 233–242. ACM.

Dunn, J. C. (1973). A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3(3):32–57.

Gan, G. (2011). *Data Clustering in C++: An Object-Oriented Approach*. Data Mining and Knowledge Discovery Series. Chapman & Hall/CRC Press, Boca Raton, FL, USA.

Gan, G. (2013). Application of data clustering and machine learning in variable annuity valuation. *Insurance: Mathematics and Economics*, 53(3):795–801.

Gan, G. (2015). A multi-asset Monte Carlo simulation model for the valuation of variable annuities. In *Proceedings of the Winter Simulation Conference*, pages 3162–3163.

Gan, G. and Lin, S. (2015). Valuation of large variable annuity portfolios under nested simulation: A functional data approach. *Insurance: Mathematics and Economics*, 62:138–150.

References II

Gan, G., Ma, C., and Wu, J. (2007). *Data Clustering: Theory, Algorithms, and Applications*. SIAM Press, Philadelphia, PA.

Kolen, J. F. and Hutcheson, T. (2002). Reducing the time complexity of the fuzzy c-means algorithm. *IEEE Transactions on Fuzzy Systems*, 10(2):263–267.

Macqueen, J. (1967). Some methods for classification and analysis of multivariate observations. In LeCam, L. and Neyman, J., editors, *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics andProbability*, volume 1, pages 281–297, Berkely, CA, USA. University of California Press.