# The Optimal Estimation of Fuzziness Parameter in Fuzzy C-Means Algorithm

Hsun-Chih Kuo[1] and Yu-Jau Lin[2(✉)]

[1] Department of Statistics, National Chengchi University, Taipei City, Taiwan, ROC
seankuo@nccu.edu.tw

[2] Department of Applied Mathematics, Chung Yuan Christian University,
Taoyuan City, Taiwan, ROC
yujaulin@cycu.edu.tw

**Abstract.** The fuzziness parameter $m$ is an extra parameter that facilitates the iterative formulas of Fuzzy c-means (FCM). However, the parameter $m$, commonly set to be 2.0, is an important factor that effects the effectiveness of FCM. In literatures, the statistical study of $m$ is so far not available. Viewing $m$ as a random variable, we propose a novel idea to optimize the fuzziness parameter $m$. For the model selection, a modified cluster validity index is defined as the optimal function of $m$ and improve the effectiveness of FCM. Then the simulated annealing algorithm is applied to approximate its estimate.

**Keywords:** Fuzzy c-means · Xie-Beni index · Simulated annealing · Markov chain

## 1 Introduction

Clustering methods [3] can be roughly divided into two groups: hierarchical and classification methods. Classification method aims to find the best partition of data into $c$ clusters in such a way that one criterion is optimized. Here we consider the fuzzy classification and use the Fuzzy C-Means (FCM) algorithm [1,2,6,7]. In addition to the specification of the number $c$ of clusters in the data set, FCM method requires to choose the fuzziness parameter $m$, an important factor that influences the effectiveness of FCM. Note that the study of $m$ has not been completely investigated in literatures. Pal and Bezdek [4] suggested $m \in [1.5, 2.5]$, and Yu et al. [10] proposed a theoretical upper bound for $m$ to prevent the sample mean from being the unique optimizer of an FCM objective function. Wu [8] showed that the parameter $m$ influenced the robustness of FCM and $m \in [1.5, 4]$. For a large theoretical upper bound, they suggested the implementation of the FCM with a suitable large $m$ value. Also, the value $m = 4$ is recommended for FCM when the data contains noise and outliers. In practical use purpose, $m$ is commonly fixed to 2. This choice allows an easy computation of the membership values.

L. Polkowski et al. (Eds.): IJCRS 2017, Part I, LNAI 10313, pp. 566–575, 2017.
DOI: 10.1007/978-3-319-60837-2_45

# 2 Methodology

## 2.1 The FCM Algorithm

For a given number of $c$ clusters and the fuzzifier $m > 1$, the FCM algorithm is an iterative procedure that minimizes the objective function

$$J(c) = \sum_{k=1}^{c} \sum_{i=1}^{N} u_{ki}^{m} \, d^2(x_i, a_k), \tag{1}$$

where $d(x_i, a_k)$ is the distance (dissimilarity) between the cluster center $a_k$, $k = 1, 2, \cdots, c$ and the data $x_i$, $i = 1, 2, \cdots, N$(number of sample size), and $u_{ki}$ denotes the fuzzy membership value of object $x_i$ to the cluster $k$ that satisfies the following conditions

$$0 \leq u_{ki} \leq 1 \text{ and } \sum_{k=1}^{c} u_{ik} = 1 \tag{2}$$

FCM algorithm is then minimized (1) by the following iterative equations.

$$a_k = \frac{\sum_{i=1}^{N} u_{ki}^{m} x_i}{\sum_{i=1}^{N} u_{ki}^{m}} \tag{3}$$

$$u_{ki} = \frac{1}{\sum_{j=1}^{c} \left( \frac{d(x_i, a_k)}{d(x_i, a_j)} \right)^{\frac{1}{m-1}}} \tag{4}$$

Fuzzy partitioning is carried out through an iterative optimization (minimizing) of the objective function $J(c)$ by alternatively updating the membership $\mu_{ij}$ and the cluster center $a_k$.

## 2.2 The XB Index

Among the existing validity indices to evaluate the goodness of clustering according to a given number of clusters, the Xie–Beni (XB) index [9] is a credible fuzzy-validity criterion based on a validity function which identifies overall compact and separate fuzzy c-partitions. This function depends upon the data set, geometric distance measure, and distance between cluster centroids and fuzzy partition, irrespective of any fuzzy algorithm used. For evaluating the goodness of the data partition, both cluster compactness and intercluster separation should be taken into account. For FCM algorithm with $m = 2.0$, the XB index can be shown to be

$$\mathrm{XB}(c) = \frac{J(c)}{N d_{\min}} \tag{5}$$

where $d_{\min}$ is the minimum distance between cluster centroids. The more separate the clusters are, the larger $d_{\min}$ and the smaller $\mathrm{XB}(c)$.

### 2.3 The Parameter $m$ as a Random Variable

The following numerical example shows that different values of $m$ yield to different models according to XB index. To demonstrate the class clustering, we generate a pseudo dataset from 4 clusters centered at (5, 5), (5,−5), (−5, 5) and (−5, −5), each has 12 observations and they follow the two dimensional independent normal distribution. A realization of simulated data, denoted by $D_1$, is shown in Fig. 1. Each elements of the same cluster are marked with the same color points. Visually, the number of clusters are likely to be 4 and possibly 3.

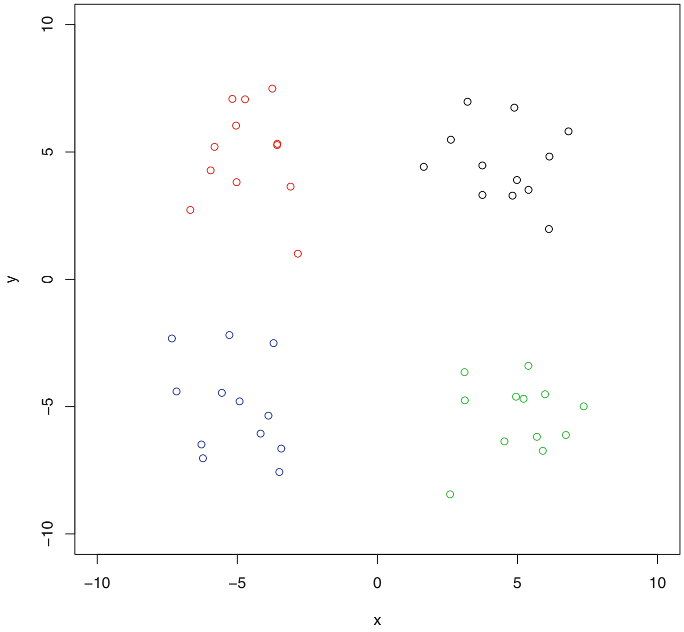


**Fig. 1.** The scatter plot of data set $D_1$

Table 1 lists the summary of the model suggested by XB indices.

We see, when $m = 6.0$, the suggested cluster number by XB index is $c = 3$. But it is incorrect. This implies that XB index is somehow not perfect since it depends on $m$.

Different from the classical analysis, our novel idea is to view the fuzziness parameter $m$ as a random variable in the XB index. That is, given the data, XB index consists of two parameter $m$ and $c$.

$$\mathrm{XB}(c, m) = \frac{J(c)}{N d_{\min}} \tag{6}$$

Next, we apply the simulated annealing algorithm to find the maximum likelihood estimator of $m$.

**Table 1.** Different values of $m$ yield to different models.

| $m$ | $c$ (suggested by XB) | Correct or not |
|---|---|---|
| 2.0 | 4 | Yes |
| 3.0 | 4 | Yes |
| 4.0 | 4 | Yes |
| 5.0 | 4 | Yes |
| 6.0 | 3 | No |
| 7.0 | 3 | No |

### 2.4 Simulated Annealing Algorithm

The simulated annealing algorithm (SA) [5] which employs a probabilistic procedure can approximate the minimizer $m$ of an optimal function. It originally simulates the process of slow cooling of molten metal to achieve the minimum function value in a minimization problem. The cooling phenomenon of the molten metal is simulated by introducing a temperature like parameter and cooling it down using the concept of Boltzmann's probability distribution. The Boltzmann's probability distribution implies that the energy $E$ of a system in thermal equilibrium at temperature T is distributed probabilistically according to the relation $P(E) = e^{-E/kT}$, where $P(E)$ denotes the probability of achieving the energy level E, and k is called the Boltzmann's constant [5]. SA's major advantage over other methods is an ability to avoid becoming trapped at local minima. The algorithm employs a random search for which not only accepts changes that decrease objective function but also some changes that increase it. The latter are accepted with probability $p = e^{-\triangle F/t_n}$, where $\triangle F = F_n - F_{n-1}$ is the increase or the decrease in objective function value and $F_n$ is a control parameter, which by analogy with the original application is known as the system "temperature" irrespective of the objective function involved.

## 3 The Estimation of $m$

The theoretical distribution of $m$ is so far not available since $m$ depends on the data and the corresponding objective functions. For example, if the optimal function is to minimize $J(c, m) = J(c)$ in (1), we consider $m$ as a random variable that has the probability density function $f(m|c)$ of the form

$$m \sim f(m|c) \propto e^{-J(c,m)}$$

We see that the estimate, so called maximum likelihood estimate in statistics, of maximizing a probability density function with kernel $e^{-J(c,m)}$ is equivalent the optimal estimate of minimizing $J(c, m)$.

We define the cluster validity function as the objective function of $m$ in our analysis.

$$G(k) = \frac{\mathrm{XB}(k, m)}{\min_{i \neq k}\{\mathrm{XB}(k, m)\}} \tag{7}$$

In determining the number of clusters $c$, the model with minimum $\mathrm{XB}(c, m)$ is preferred. However, It's difficult to optimize both the number of cluster $c$ and the fuzziness parameter $m$ together in the same procedure since the problem of choosing best $c$ has been a hard problem in classification. Given a likely $c$, the new validity function $G(k)$ in (7) can differentiates the best model between other clusters.

With vague information when $m = 2.0$ in dataset $D_1$, the preferred model can be identified to be 4. We emphasize the difference between the best and the second best models in terms of XB index by taking the ratio of them. Since small $\mathrm{XB}(c, m)$ indicates a better number of cluster $c$, the minimum of $G(c, m)$ is desired. And SA, the general procedure in optimizing an objective function with different initial values of cluster center $a_k$ and $\mu_{ij}$, can approximate the fuzziness parameter $m$ as a sequence of Markov chain that ultimately converges to its minimizer.

### 3.1 Simulation Study

Using the data used in the numerical example, we see the *correct* model $c = 4$, and we wish to estimate $m$ given the data and possible clusters $(2, 3, 4, 5)$. According to XB index, the best model $c^* = 4$. Then the minimizer $m^*$ of objective function $G_4(m)$ is the estimate.

$$G_4(m) = \frac{\mathrm{XB}(4, m)}{\min\{\mathrm{XB}(2, m), \mathrm{XB}(3, m), \mathrm{XB}(5, m)\}} \tag{8}$$

We propose the following algorithm: start from $m = 2.0$. The number of clusters is firstly determined by the one, say $c^*$, with minimum XB index among possible clusters. Note that $\hat{c}$ is a ball park figure. Next, SA is applied to locate the estimate of $\hat{m}$ of the cluster validity function $G_4(m)$. That is,

$$G_4(\hat{m}) = \min_{m>1} G_4(m, c^*).$$

Finally, set $m = \hat{m}$, proceed FCM and double check XB indices for possible models to ensure $c^*$ is the cluster with smallest XB index. If not, run the above procedure again until the estimate $\hat{m}$ agrees with the indicator XB.

**FCM** algorithm

(a) Pre-set the cluster number $c$ and the fuzziness parameter $m$.

(b) Set initial values of cluster center $a_k$ and fuzziness membership $u_{ki}$, $i = 1, 2, \cdots, N$(number of sample size), $k = 1, 2, \cdots, c$.

(c) $u_{ki} = \frac{1}{\sum_{j=1}^{c} \left( \frac{d(\mathbf{x}_\mathbf{i}, a_k)}{d(x_i, a_j)} \right)^{\frac{1}{m-1}}}$

(d) $a_k = \frac{\sum_{i=1}^{N} u_{ki}^m x_i}{\sum_{i=1}^{N} u_{ki}^m}$

(e) $J(c) = \sum_{k=1}^{c} \sum_{i=1}^{N} u_{ki}^m \, d^2(x_i, a_k) < \epsilon = 10^{-4}$, stop;

else, go to Step (b).

**SA** algorithm

1 Set the starting values of $m^{(0)} = 2.0$.

2 Calculate XB($k$, $m^{(0)}$), $k = 1, 2, \cdots, c$, by FCM algorithm and XB($c, m$) in (6) and determine the one with smallest XB index as the preferred number of clusters, say $c^*$.

3 At state $j$, propose a candidate $m^* \sim \mathrm{N}(m^{(j)}, 0.05^2)$.

4 Let $u$ follow uniform distribution$(0, 1)$.

5 Accept the candidate $m^*$ as the next state value of Markov chain $\{m^{(j)}\}_{j=1,2,\cdots}$ with probability

$$\alpha_1 = e^{\frac{(-1)}{T_j}\left(G(m^*) - G(m^{(j-1)}\right)}.$$

That is

$$m^{(j+1)} = \begin{cases} m^* & \text{if } u < \alpha \\ m^{(j)} & \text{otherwise.} \end{cases}$$

where $T_j = \dfrac{100}{j\log(j)}$, $G(m) = G(m, c^*)$ is defined in (7).

6 If $j < n_2$, say $n_2 = 20,000$, then go to Step 3.

7 Calculate XB(2,$\hat{m}$), XB(3, $\hat{m}$), XB(4, $\hat{m}$) and XB(5,$\hat{m}$). If $c^*$ is the one with smallest XB index, then STOP; otherwise set $j = 0$ and $m^{(0)}$ and go to Step 2.

$$\hat{m} = m^{(n_2)}\text{is the minimier of } G(m)$$

$$\hat{c} = c^*\text{is the number of clusters.}$$

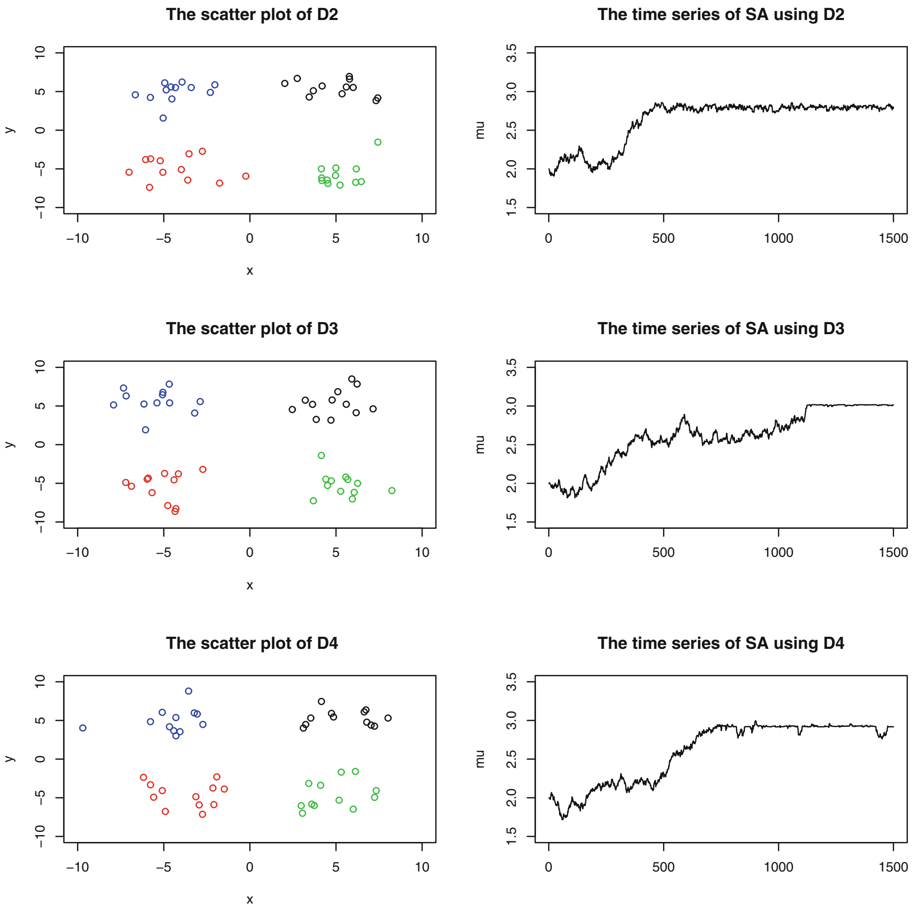
8 The cluster center $a_k$, $k = 1, 2, \cdots, \hat{c}$ can be obtained by FCM with $m = \hat{m}, c = \hat{c}$.

## 4 The Numerical Experiment

We apply our proposed method to the data with obvious clusters and see how well it performs. To make comparisons, we simulate data based on the statistical distribution of data set $D_1$ in Sect. 2.3.

### The Data Scheme (I): Standard Derivation $\sigma = 1.5$

Three data sets, $D_2, D_3, D_4$, are simulated from the same target distribution as $D_1$ in Fig. 1, each has 12 observations and they follow the 2 dimensional independent normal distribution with same standard deviation $\sigma = 1.5$, $N_2(\mu_i = (5, 5), (-5, -5), (5, -5), (-5, 5), \sigma)$. Together with the time series plots of $m$ in applying SA algorithm, the scatter plots of the three data sets are shown in Fig. 2.


The scatter plot of D2

The time series of SA using D2

The scatter plot of D3

The time series of SA using D3

The scatter plot of D4

The time series of SA using D4

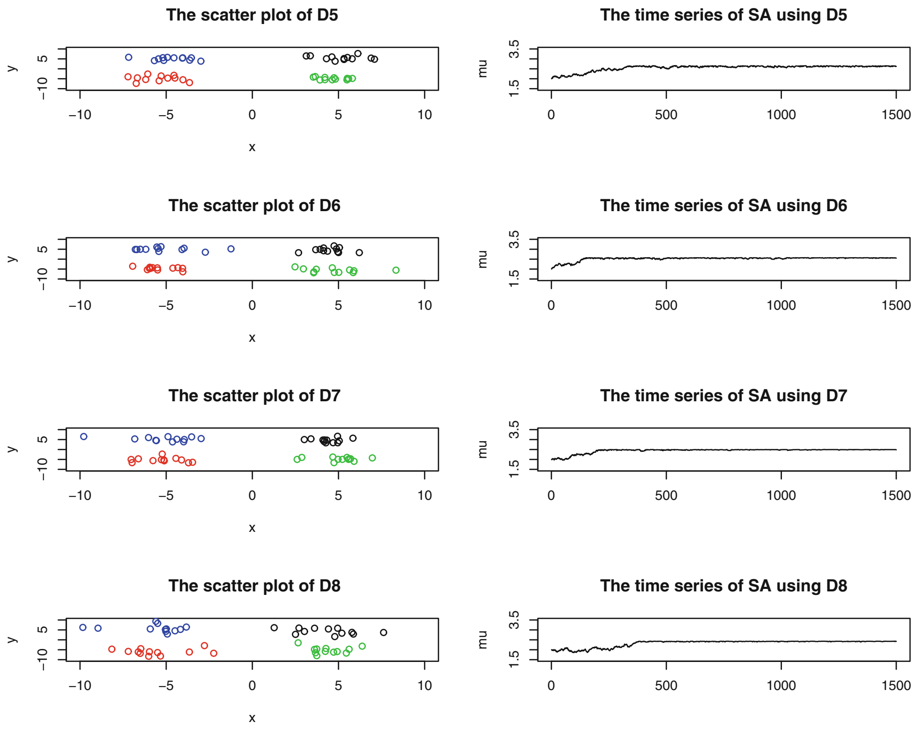
**Fig. 2.** .

## The Data Scheme (II): $\sigma = 1.0, 1.0, 1.2, 1.8$

Four data sets, $D_5, D_6, D_7, D_8$, are simulated from 4 clusters, the same mean points as $D_1$, each has 12 observations and they follow the 2 dimensional independent normal distribution with standard deviations $\sigma = 1.0, 1.0, 1.2, 1.8$ $N_2(\mu_i = (5,5), (-5,-5), (5,-5), (-5,5),\ \sigma\ )$. The scatter plots and the time series plots of $m$ given by SA algorithm using the data sets are shown in Fig. 3.

## The Data Scheme (III): $c = 2, 3$

Two data sets, $D_9, D_{10}$, are simulated from 2 and 3 clusters, each has 12 observations and they follows the 2 dimensional independent normal distributions with same standard deviation $\sigma = 1.5$, $N_2(\mu_i = (5,5),\ (-5{,}5),\ \sigma = 1.5)$ and


The scatter plot of D5

The time series of SA using D5

The scatter plot of D6

The time series of SA using D6

The scatter plot of D7

The time series of SA using D7

The scatter plot of D8

The time series of SA using D8

**Fig. 3.** .

$N_2(\mu_i = (5, 5)$,$(-5,\ -5)$, $(-5$,$5)$, $\sigma = 1.5)$, respectively. The scatter plots of the data sets and the time series plots of $m$ are shown in Fig. 4.

## 4.1 Concluding Remarks

As seen the the time series plots of $m$ given by SA algorithm, their Markov chains converge. For example in $D_1$, starting from $m = 2.0$, the last 10 realizations of $m^{(j)}_{j=1,2,\cdots,1500}$ are

| | | | | |
|---|---|---|---|---|
| 2.7847 | 2.7912 | 2.7912 | 2.7644 | 2.7832 |
| 2.8023 | 2.7930 | 2.7930 | 2.8007 | 2.7822 |

The sequence of Markov chains of $m$ given by SA algorithm using the data set $D_1$ are getting close to a fixed point $\hat{=}2.7822$. And the suggested number of clusters indicates the correct model is $c = 4$ with cluster centers $(5.91, 5.09)$, $(4.92, -6.12)$, $(-4.69, 5.17)$, $(-5.04,-5.41)$.

Repeated numerical experiments using data sets $D_i, i = 1, 2, \cdots, 10$, show that our proposed work well. The suggested clusters $\hat{c}$ by XB indices are as good

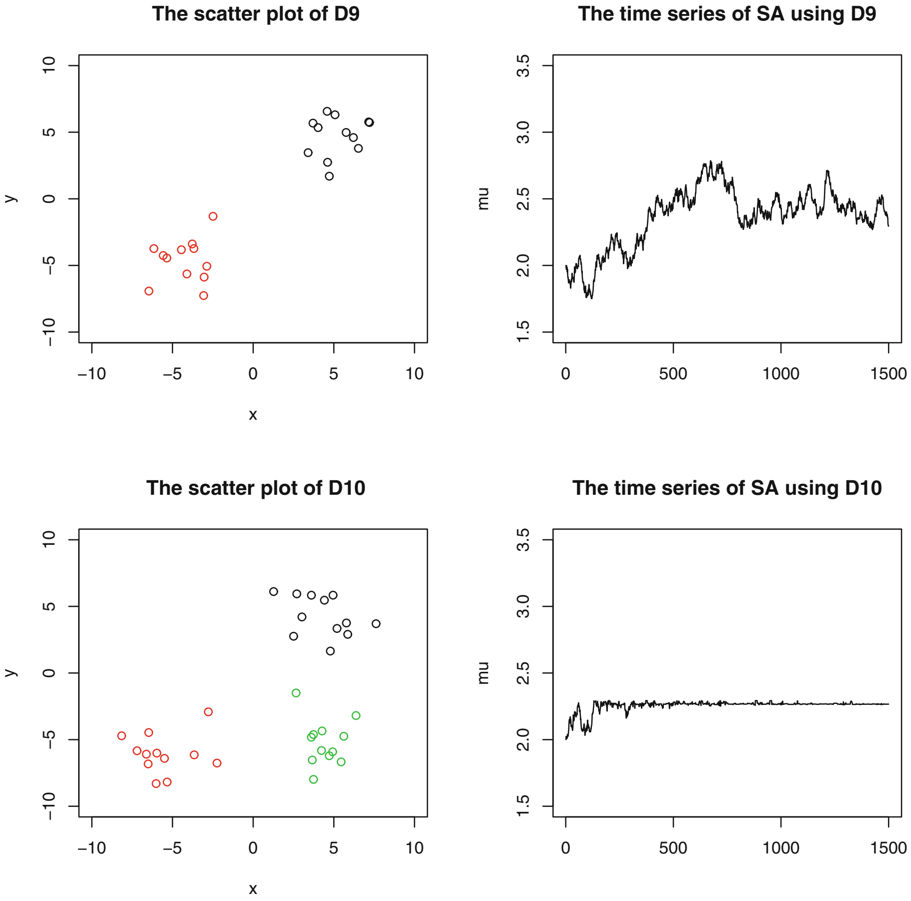


**Fig. 4.** .

| Data | c | $\hat{c}$ | $\hat{m}$ |
|---|---|---|---|
| $D_1$ | 4 | 4 | 2.7822 |
| $D_2$ | 4 | 4 | 3.0219 |
| $D_3$ | 4 | 4 | 2.5896 |
| $D_4$ | 4 | 4 | 2.6703 |
| $D_5$ | 4 | 4 | 2.6221 |
| $D_6$ | 4 | 4 | 2.5525 |
| $D_7$ | 4 | 4 | 2.4842 |
| $D_8$ | 4 | 4 | 2.4221 |
| $D_9$ | 2 | 2 | 2.2946 |
| $D_{10}$ | 3 | 3 | 2.5525 |

as expected. The SA estimates of $m$ are eventually convergent. The estimates $\hat{m}$ range from 2.1 to 3.0. See the following table for the summary.

Since the optimal estimates of $m$ are not far from the commonly used value, 2.0, the cluster centers computed by FCM are about the same.

## 5 Conclusion

Fuzzy c-means is a common, fast and useful method of clustering classification. In applying FCM algorithm, the fuzziness parameter $m$, originally designed to facilitate the iterative formulas of FCM, is usually set to be 2.

In this paper, we first show that $m$ indeed an important factor in determining the cluster validity. We then view $m$ as a random variable and apply SA algorithm to approximate the optimal estimate of $m$ based on the modified XB index.

Even though the results of our numerical experiments are not surprising, our approach is novel. We successfully delete the effect due to the extra parameter $m$ by finding its minimizer and thus guarantee the effectiveness of FCM. Furthermore, the statistical distribution of $m$ can be possibly available by the Markov chain Monte Carlo in the future.

## References

1. Bezdek, J.C.: Numerical taxonomy with fuzzy sets. J. Math. Biol. **1**, 57–71 (1974)
2. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York (1981)
3. Gath, I., Geva, A.B.: Unsupervised optimal fuzzy clustering. IEEE Trans. Pattern Anal. Mach. Intell. **11**, 773–781 (1989)
4. Pal, N.R., Bezdek, J.C.: On cluster validity for fuzzy c-means model. IEEE Trans. Fuzzy Syst. **1**, 370–379 (1995)
5. Wang, X.Y., Garibaldi, J.M.: Simulated annealing fuzzy clustering in cancer diagnosis. Informatica **29**, 61–70 (2005)
6. Wang, W., Zhang, Y.: On fuzzy cluster validity indices. Elsevier Fuzzy Sets Syst. **158**, 2095–2117 (2007)
7. Wu, K.L., Yang, M.S.: A cluster validity index for fuzzy clustering. Pattern Recogn. Lett. **26**, 1275–1291 (2005)
8. Wu, K.L.: Analysis of parameter selections for fuzzy c-means. Elsevier Pattern Recogn. **45**, 407–415 (2012)
9. Xie, X.L., Beni, G.A.: A validity measure for fuzzy clustering. IEEE Trans. Pattern Anal. Mach. Intell. **13**, 841–847 (1991)
10. Yu, J., Cheng, Q., Huang, H.: Analysis of the weighting exponent in the FCM. IEEE Trans. Syst. Man Cybern. B **34**, 634–639 (2004)