國 立 交 通 大 學

電信工程學系

碩 士 論 文

決定群中心個數 k 與位置的
分裂 K-均值分群演算法

Divisive *K*-Means Clustering Algorithm for Determining *k* and Positions of Cluster Centers

研 究 生　：林佑信

指導教授　：李程輝 教授

中 華 民 國 九 十 八 年 六 月

決定群中心個數 k 與位置的分裂 K-均值分群演算法

Divisive *K*-Means Clustering Algorithm for Determining *k* and Positions of Cluster Centers

研 究 生　：林佑信　　Student: You-Shin Lin

指導教授　：李程輝 教授　　Advisor: Dr. Tsern-Huei Lee

國 立 交 通 大 學

電 信 工 程 學 系 碩 士 班

碩 士 論 文

A Thesis

Submitted to Department of Communication Engineering

College of Electrical and Computer Engineering

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of

Master of Science

in

Communication Engineering

June 2009

Hsinchu, Taiwan

中 華 民 國 九 十 八 年 六 月

決定群中心個數 *k* 與位置的分裂 *K*-均值分群演算法

學生：林佑信　　指導教授：李程輝 教授

國立交通大學電信工程學系碩士班

中文摘要

分群法(clustering)近來是一個眾所周知的研究主題，而且它也被廣泛的應用在許多的領域中。在眾多的分群演算法之中，*k*-均值演算法(*k*-means algorithm)是一個通俗、簡單且快速的分群演算法。然而在 *k*-均值演算法的應用上，卻有兩個主要的問題：第一，在一個真實的資料集合中，確切的 k 值是未知的；第二，*k*-均值演算法很難有效的去選擇初始的群聚中心點，而且群聚中心點的初始位置的選擇會大大影響了分群的結果。為了解決這兩個主要的問題，我們提出了一個新的演算法，其主要是在 *k*-均值演算法的目標函數上多加了一個衝突的項，使得這分群過程對於初始群中心的選擇不會那麼敏感。結合分群的驗證方法，我們能夠決定最佳的群聚中心個數與其所在的位置。我們在許多自創的資料組裡作模擬，都能夠有效的得到最佳的分群結果。

Divisive *K*-Means Clustering Algorithm for Determining *k* and Positions of Cluster Centers

Student: You-Shin Lin Advisor: Dr. Tsern-Huei Lee

Institute of Communication Engineering
National Chiao Tung University

Abstract

Clustering is a well-known research topic, which applied widely in many fields. Among of the clustering algorithms, *k*-means algorithm is one of the most popular, simple, and fast clustering algorithm. However, there are two major problems in the application of the *k*-means algorithm. First, the right value of *k* is usually unknown in a real data set. Second, it is difficult to select effectively initial cluster centers, and the clustering result is sensitive to the initial cluster centers. In order to solve the two problems, we propose a new algorithm which extends the standard *k*-means algorithm by introducing a conflict term to the objective function to make the clustering process not sensitive to the initial cluster centers. Combined with the cluster validation technique, we can determine the optimal *k* and the positions of cluster centers. Simulation results on synthetic data sets show the effectiveness of the proposed algorithm in determining the number and positions of the cluster centers.

誌謝

能完成這篇論文，首先我要感謝我的指導教授－李程輝教授。在我這兩年的研究生活中，教導我許多作研究的方法以及該有的態度，使我從中獲得許多寶貴的經驗，成長了許多。

感謝我的父母－林穎聰先生與陳素真女士。感謝父母對我的養育之恩，並且在我的求學生涯裡，一路上對我的支持與鼓勵。感謝我的兄長－林建宏先生。從小對我的照顧與勉勵。

感謝實驗室的學長姐－景融、郁文、迺倫、耀誼、勁文、世弘、凱文、明鑫。在我的研究生涯中，對我的照顧與熱心指導。感謝同窗好友－家豪、俊德、小汪、松晏、均傑、敬堯、晨屹。陪我一起渡過研究生的酸甜苦辣。感謝實驗室的學弟妹－小机、曉薇、韋儒、建碩、惠雅。帶給實驗室活力與歡笑。

感謝我的大學好友－士展、蔡包、中敬、黑倫、貞慶、小靠、國書、阿慶、鹿哥、小飛、大支。一直以來給我精神上的支持與鼓勵，且在我煩惱不順時，陪我舒壓解悶。

感謝你們讓我有這麼精彩的碩士生活！

最後謹將此論文獻給身邊所有愛我的人及我愛的人。

林佑信　2009 年 6 月　於風城交大

Contents

List of Tables

List of Figures

Chapter 1.
Introduction

Clustering has been one of the most widely studied topics in data mining and pattern recognition. The task of clustering is to group a set of objects into clusters so that objects from the same cluster are more similar to each other than objects from different clusters. Various types of clustering methods have been proposed and developed, for instances, [1], [2], [3], and [4]. The *k*-means algorithm is one of the most popular, simple and fast clustering algorithms. *K*-means algorithm was proposed by MacQueen in 1967 [5]. Its basic idea is that, given the cluster number *k* and a set of initial cluster centers stochastically, an iterative algorithm is used to improve the partition of the clusters through moving the cluster centers continually until the best partition result is obtained.

There are two major problems in the application of the *k*-means algorithm in cluster analysis. First, the number of clusters *k* needs to be determined in advance as an input to the *k*-means algorithm. In a real data

set, k is usually unknown. Second, its performance heavily depends on the initial starting conditions [6]. The k-means algorithm requires a set of initial cluster centers to start and often end up with different clustering results from different sets of initial cluster centers. In other words, the k-means algorithm is very sensitive to the initial cluster centers [7], [8].

Several papers had been proposed to address the issue of choosing the initial cluster centers for a known value of k [9]-[13]. And those simulation results are well to solve such problem. But the value of k is usually unknown in a real data set. In [14], Hamerly and Elkan have proposed statistical methods to learn k in k-means algorithm. In [15], Li et al. have proposed an agglomerative fuzzy k-means clustering algorithm to obtain the exactly number of cluster centers. In this algorithm, the initial number of cluster centers must be set to be larger than the true number of cluster centers in a data set.

In this thesis, we propose a new algorithm to solve the above two problems with the application of the k-means clustering algorithm. The new algorithm extends the standard k-means algorithm by introducing a

conflict term to the objective function to make the clustering process not sensitive to the initial cluster centers. The new algorithm does not need to know the true number of the cluster centers in advance. It runs with the value of $k=1$ at the beginning, and the value of k increases by degrees. When the least objective function value is found, the best positions of cluster centers will be obtained. So we do not need to select a set of cluster centers randomly in advance, we just need to calculate the mean value of the data set and take it as the initial cluster center. Combined with cluster validation techniques, the new algorithm can determine the optimal number of clusters and the positions of the cluster centers in a data set. Simulation results have demonstrated the effectiveness of the proposed algorithm in producing the consistent clustering results and determining the correct number of clusters in different data sets.

In Chapter 2, we introduce background of *k*-means algorithm. In Chapter 3, we briefly review the related work. In Chapter 4, we formulate the proposed algorithm to select the number and positions of clusters. In Chapter 5, simulation results are given to illustrate the effectiveness of the new algorithm. The last Section summarizes our concluding remarks.

Chapter 2.
Related Works

2.1. *K*-Means Algorithm

The *k*-means algorithm is one of the simplest unsupervised learning algorithms that provide solutions to the clustering problem. Let $X = \{X_1, X_2, \ldots, X_n\}$ be a data set of *n* objects in which each object X_i is represented as $[x_{i,1}, x_{i,2}, \ldots, x_{i,m}]$, where *m* is the number of dimensions, and $Z = [z_1, z_2, \ldots, z_k]^T$ is an *k*-by-*m* matrix containing the cluster centers. The basic idea of the *k*-means algorithm is as follows. Given the number of cluster centers *k* and selected arbitrarily *k* cluster centers at the beginning. The next step is to partition the objects to the nearest cluster center to form a cluster. When no objects are pending, the second step is completed. The third step is to compute the mean value of each cluster and make it as the new cluster center. Then it is iterative continually executing the above of second and third steps until the positions of the cluster centers have no changes. The overall clustering process of the *k*-means algorithm aims at minimizing the objective function:

$$J = \sum_{j=1}^{k} \sum_{x_i \in C_j} \left\| x_i - z_j \right\|^2 \quad (1)$$

where C_j denotes cluster j and z_j is the cluster center of C_j. The smaller J is, the more similar within group data is. The standard k-means algorithm is described as follows:

Input: Number of clusters k and data set X.

Output: The final clustering result with k clusters.

Step1: Select arbitrarily k initial cluster centers.

Step2: Partition all the objects to the cluster that has the closest center.

Step3: Compute the mean value of each cluster and renew the cluster centers.

Step4: Repeat Step 2 and Step 3 until the centers no longer change.

Step5: Output.

2.2. An Efficient *K*-means Clustering Algorithm Based on Influence Factors [16]

Leng at al. [16] have proposed an efficient k-means clustering algorithm based on influence factors to solve the clustering problem with unknown value of *k*. The algorithm has two major steps. The first step is to select initial cluster centers based on the threshold ε. The second step is to merge clusters based on the influence factor until no influence factor satisfies the merging condition. In this step, it will calculate the influence factors between each cluster pairs and merge them if any of influence factor is larger than the threshold $\alpha_{\min}$. When there is no influence factor larger than the threshold $\alpha_{\min}$, it will update the cluster centers and run the standard *k*-means algorithm to achieve the final clustering results. In the simulation result, it shows that the algorithm has high quality and obtains a well clustering result when given the best value of threshold ε and $\alpha_{\min}$.

Chapter 3.
Our Proposed Algorithm

3.1 The Proposed Algorithm

In order to handle the problem of clustering, we propose a new algorithm. The proposed algorithm aims at minimizing the following objective function:

$$P(U,Z)=\sum_{j=1}^{k}\sum_{i=1}^{n}u_{i,j}\cdot\left\|x_i-z_j\right\|^2+\lambda\cdot\frac{n}{k}\sum_{j=1}^{k}\sum_{i=1}^{k}\left\|z_i-z_j\right\|^2 \qquad (2)$$

subject to

$$u_{i,j}=\begin{cases}1, & \text{if } j=\min_{l\in V_i} l,\ 1\le i\le n\\ 0, & \text{otherwise}\end{cases},\quad V_i=\left\{j\left|\arg\min_{1\le j\le k}\left\|x_i-z_j\right\|\right.\right\} \qquad (3)$$

where $U=\left[u_{i,j}\right]$ is an n-by-k partition matrix.

The first term in (2) is the cost function of the standard k-means algorithm. If the objective function only has the first term and the value of k is unknown, the algorithm will tend to have the larger value of k to minimize the objective function. When the value of k equals to n, the clusters will be the most compact and the objective function value will be

minimum. The second term is the square summation of the Euclidean distance for the all cluster center pairs. Therefore, it tends to have the smaller value of k to minimize the objective function. So the second term is added to provide the strength of the reverse to make the value of k do not increase unlimited. And we will want to find the exactly value of k which makes the summation of the two terms in (2) is minimum.

The coefficient n/k of the second term in (2) is added in order to make the two terms have the same quantity of items. Because the first term has the items of $n*k$, but the second term only has the items of $k*k$. If the value of n is large more than k, the value of the first term must dominate the objective function.

The main idea of the proposed algorithm is as follows. At the beginning, we must set the value of the conflict factor λ. Then we use a cluster center to find the opti mum clustering result by running the standard k-means algorithm in a given data set. In other word, its initial optimum position of the cluster center is the barycenter of all objects in the data set. The initial objective function value P can also be obtained.

The next step is to choose arbitrarily an object from the cluster which has the most number of objects and set it as a new cluster center. Then we also try to find the optimum clustering result by taking the two cluster centers as input to run the standard *k*-means algorithm. After finished the standard *k*-means algorithm, we can compute the optimum objective function value P_{new} and compare with the initial value of P. If $P_{new} < P$, we continue to run the above steps by increasing the value of k continually until we obtain the minimum objective function value. When we find the minimum value of P, we will obtain the optimum number and positions of the cluster centers. The overall flowchart of the proposed algorithm is shown in Fig. 1.

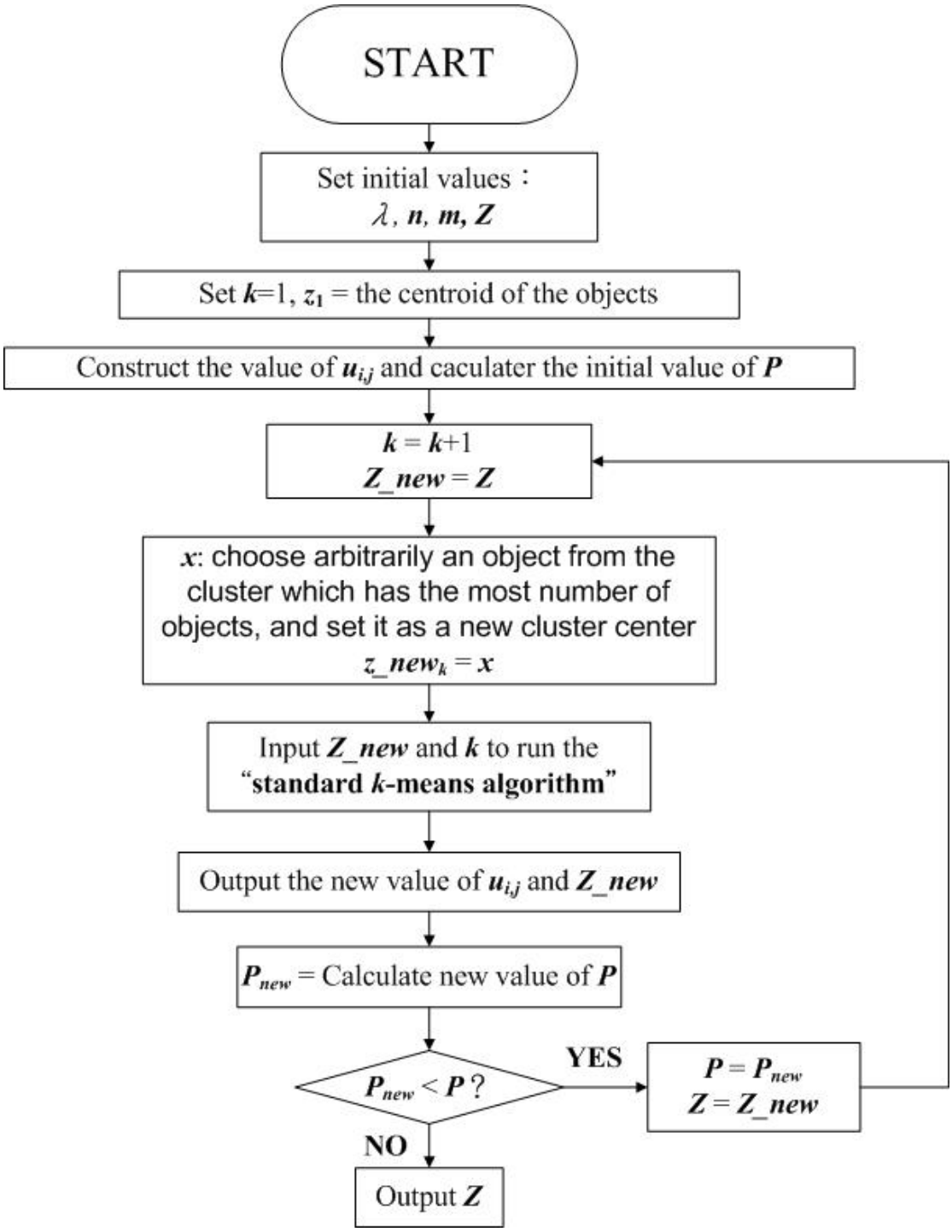

Fig. 1. The flowchart of the proposed algorithm.

3.2 The Properties of The Proposed Algorithm

In the clustering process, the proposed algorithm tries to minimize the within cluster dispersion and the separations between cluster centers. In order to balance the two factors, the parameter λ plays an important role in the minimization process. The parameter λ has the following properties to control the clustering process.

- When λ is small such that

$$\sum_{j=1}^{k}\sum_{i=1}^{n} u_{i,j} D_{i,j} \gg \lambda \cdot \frac{n}{k} \sum_{j=1}^{k}\sum_{i=1}^{k} \left\| z_i - z_j \right\|^2$$

The first term $\sum_{j=1}^{k}\sum_{i=1}^{n} u_{i,j} D_{i,j}$ will dominate the objective function. So the clustering process will tend to have the larger value of k to minimize the within cluster dispersion.

- When λ is large such that

$$\sum_{j=1}^{k}\sum_{i=1}^{n} u_{i,j} D_{i,j} \ll \lambda \cdot \frac{n}{k} \sum_{j=1}^{k}\sum_{i=1}^{k} \left\| z_i - z_j \right\|^2$$

The second term $\lambda \cdot \frac{n}{k} \sum_{j=1}^{k}\sum_{i=1}^{k} \left\| z_i - z_j \right\|^2$ will dominate the objective function. So the clustering process will tend to have the small value of k to minimize the distances between cluster centers.

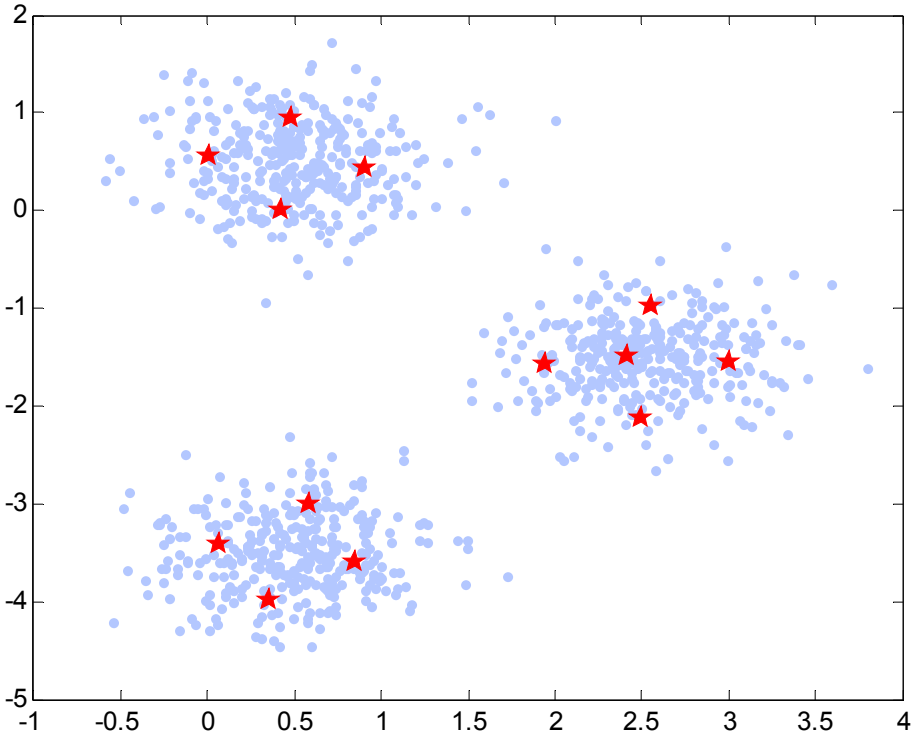

Fig. 2. The result obtained via the proposed algorithm with $\lambda = 0.001$.

For example, we run a synthetic data set of 1000 objects in a two dimension space. Fig. 2 shows the result obtained via the proposed algorithm with $\lambda = 0.001$. We can see that when λ is very small, the number of cluster centers generated by the proposed algorithm was more larger than true number of cluster centers.

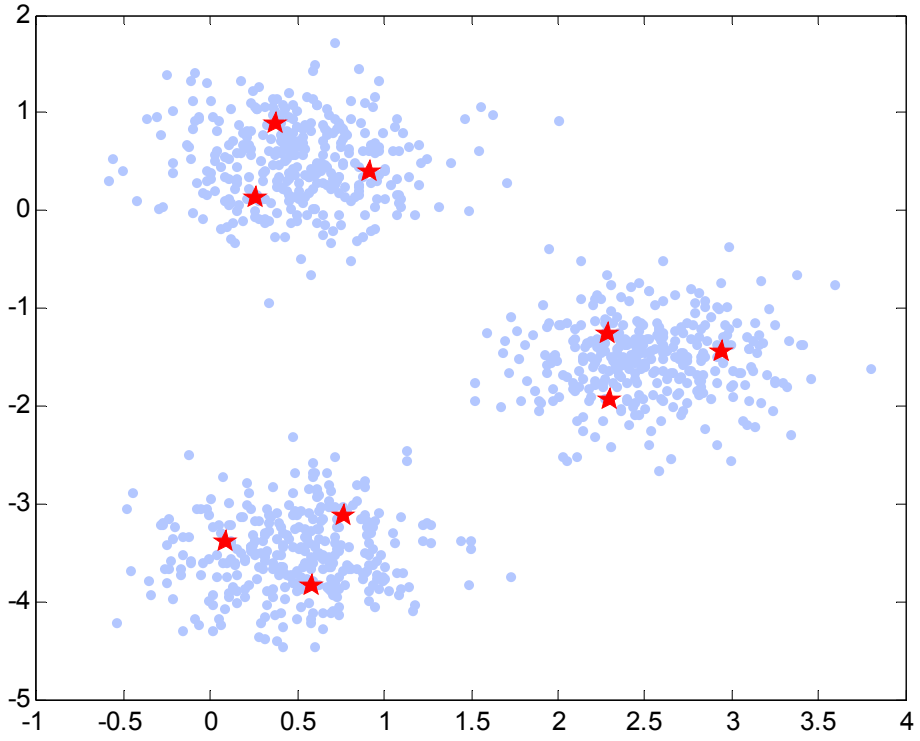

Fig. 3. The result obtained via the proposed algorithm with $\lambda = 0.002$.

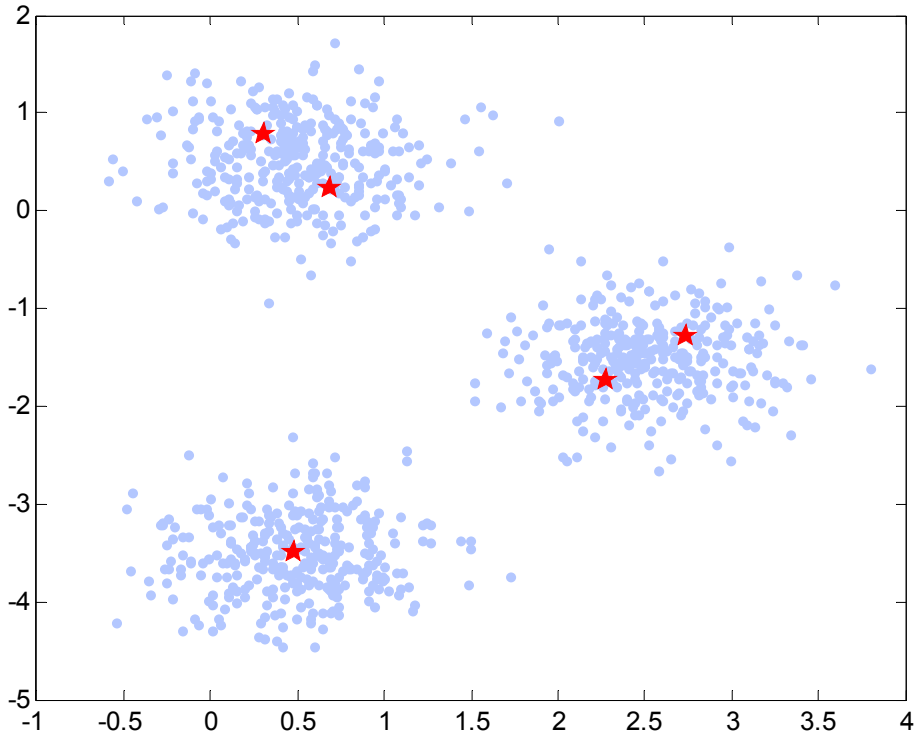

Fig. 4 The result obtained via the proposed algorithm with $\lambda = 0.003$.

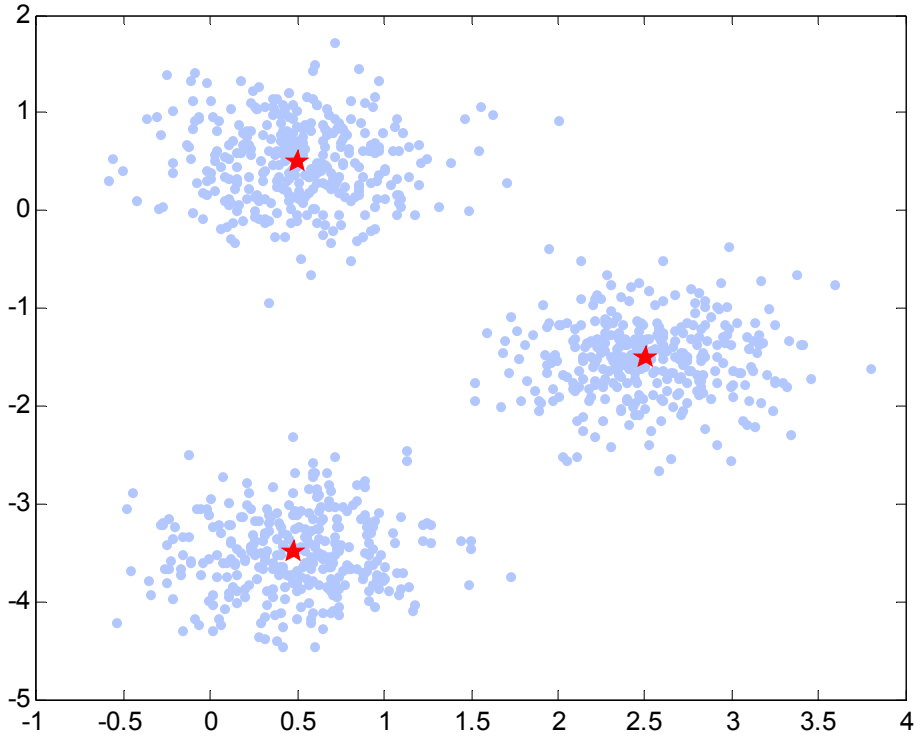

Fig. 5 The result obtained via the proposed algorithm with $\lambda = 0.01$.

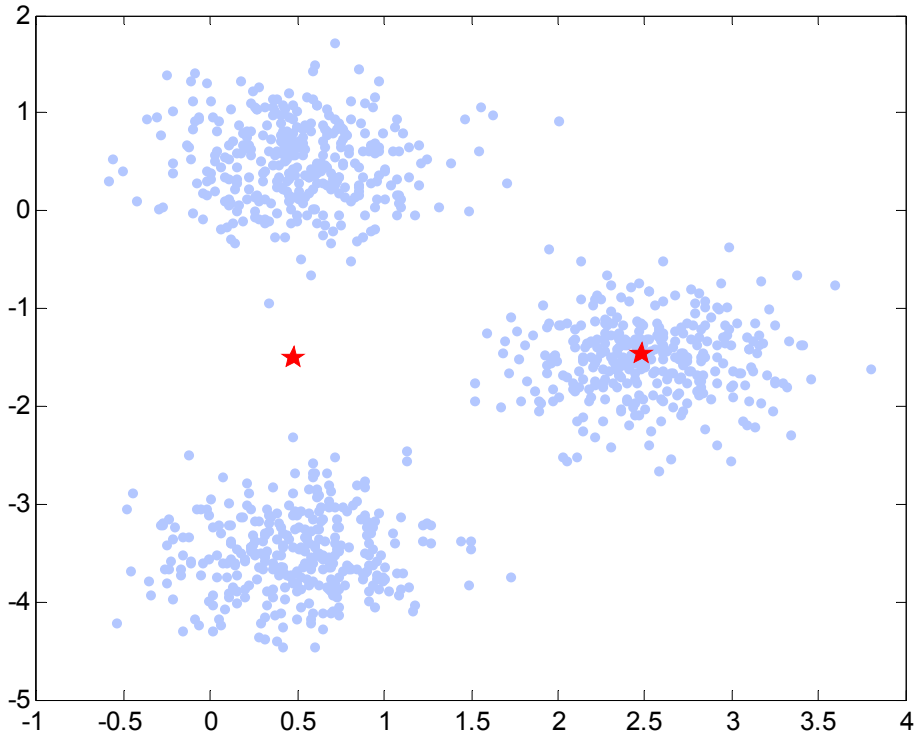

Fig. 6 The result obtained via the proposed algorithm with $\lambda = 0.18$.

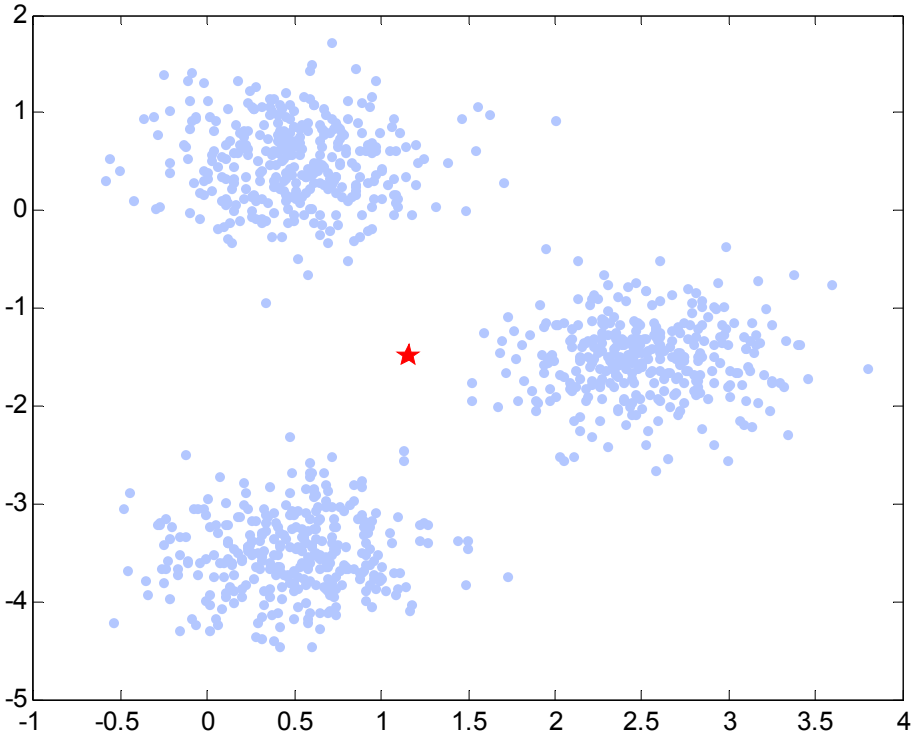

Fig. 7 The result obtained via the proposed algorithm with $\lambda = 0.3$.

Form the Fig. 3 to Fig.7, we can see that the result of the cluster centers changes in a decreasing order while the value of λ changes in an increasing order. As λ increased, the second term will gradually dominate the objective function, and the clustering result will tend to have small value of *k*. However, when λ increased to certain level, the number of cluster centers was same as the number of true cluster centers. This indicates that the value of λ at this time was right in finding the true cluster centers. Fig. 8 shows the clustering result of the data set of the example by running the proposed algorithm with different value of λ.

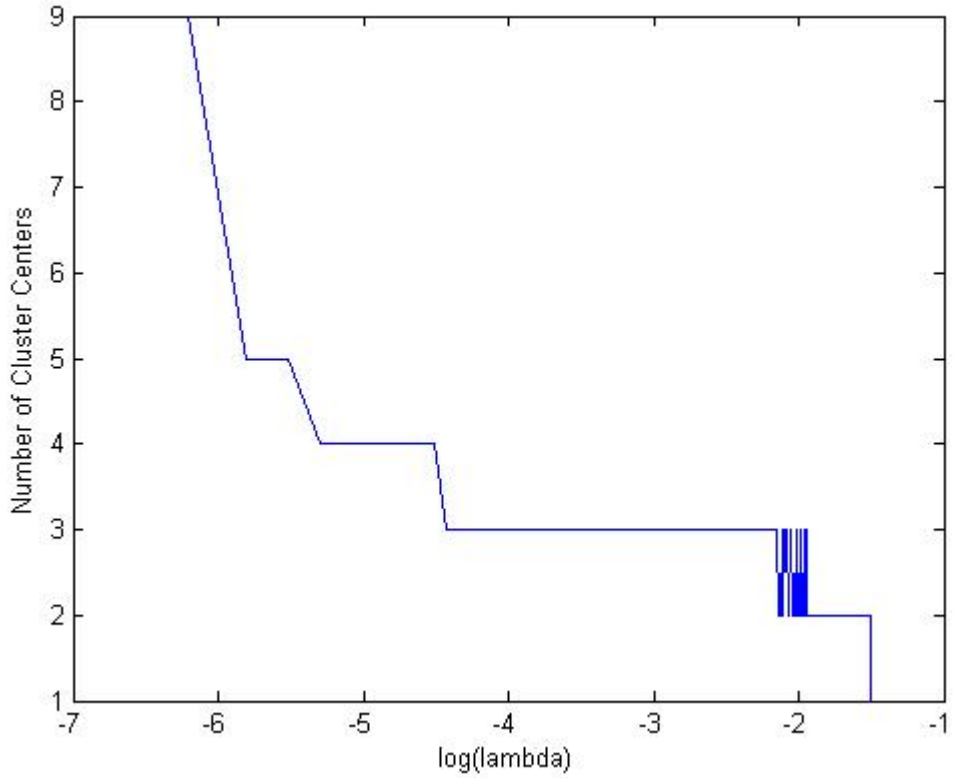

Fig. 8. The number of cluster centers with respect to different value of λ

3.3 The Overall Implementation

The overall implementation of the algorithm is shown in Fig. 9, and it automatically run the proposed algorithm to find the best number and positions of cluster centers.

In the implementation, there are two major loops. In the first loop, we find the value of conflict factor $\lambda_{\min}$ such that the proposed algorithm will produce k cluster centers, and the value of k must large or equal to the threshold α which we defined. If the true number of cluster centers is not

larger than α, the first loop will guarantee the best number of the cluster centers not be missed. In this simulation, we define the value of $\alpha = 30$ because there are not a lot of clusters in a real data set generally. In the second loop, the number of cluster centers k is changed in a decreasing order while λ increases slowly. Because the second term in (2) will dominate the objective function by degrees. We consider that the value of λ increases from $\lambda_{\min} : \lambda = \lambda_{\min} \times t$, where t = 2, 3, …, and run the proposed algorithm for each λ. In this loop, we further add a clustering validation step to validate the clustering result and record the clustering result and validation value. The clustering validation index will be defined and studied in a later. When the number of the cluster centers equals to 1, we will stop the loop, and the clustering result with the least validation index value will be obtained in the output of the implementation.

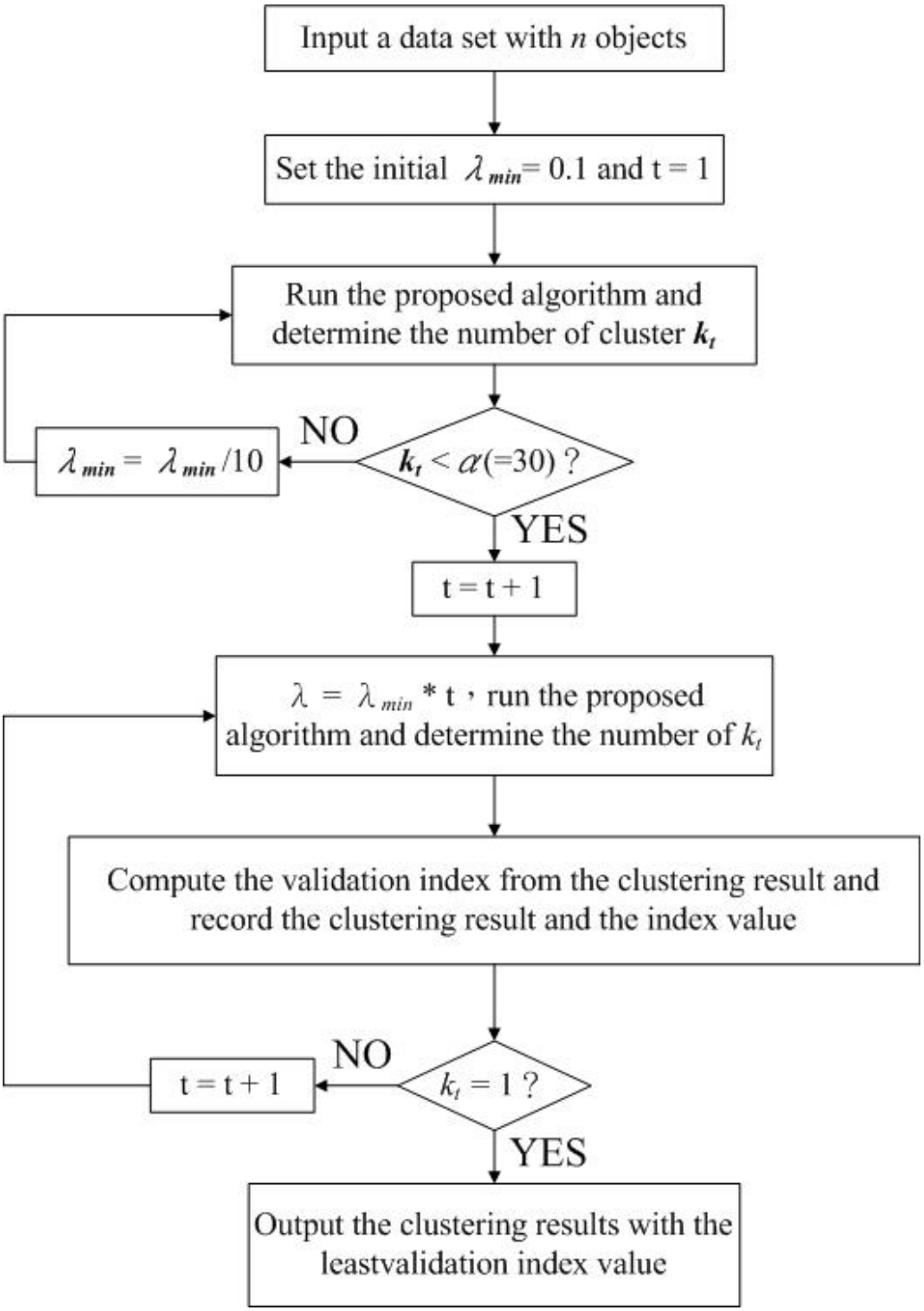

Fig. 9. The flowchart of the overall implementation.

3.4 The Validation Index

The validation index we used is proposed by Sun et al. [17]. The validation index is constructed based on the average compactness of the within clusters and separations between clusters. The validation index is proposed as following form:

$$V(U,Z,k) = Scat(k) + \frac{Dist(k)}{Dist(k_{\max})} \tag{4}$$

where the first term is defined as follows:

$$Scat(k) = \frac{\frac{1}{k}\sum_{i=1}^{k}\left\|\sigma(z_i)\right\|}{\left\|\sigma(X)\right\|} \tag{5}$$

where

$$\sigma(X) = \left[\sigma_1(X), \sigma_2(X), \ldots, \sigma_m(X)\right]^T, \tag{6}$$

$$\sigma_m(X) = \frac{1}{n}\sum_{i=1}^{n}\left(x_{i,m} - \bar{x}_m\right)^2, \tag{7}$$

$$\bar{x}_m = \frac{1}{n}\sum_{i=1}^{n} x_{i,j}, \tag{8}$$

$$\sigma(z_l) = \left[\sigma_1(z_l), \sigma_2(z_l), \ldots, \sigma_m(z_l)\right]^T, \text{ and} \tag{9}$$

$$\sigma_m(z_l) = \frac{1}{n}\sum_{i=1}^{n} u_{i,l}\left(x_{i,m} - z_{l,m}\right)^2 \tag{10}$$

The first term in (4) represents the compactness of the within clusters.

The value of the $Scat(k)$ generally decreases when k increases because the clusters become more compact. The second term $Dist(k)$ represents the separations between clusters, and it is defined as follows:

$$Dist(k) = \frac{D_{\max}^2}{D_{\min}^2} \sum_{i=1}^{k} \left(\sum_{j=1}^{k} \left\| z_i - z_j \right\|^2 \right)^{-1} \tag{11}$$

where

$$D_{\min} = \min_{i \neq j} \left\| z_i - z_j \right\| \quad \text{and} \quad D_{\max} = \max_{i \neq j} \left\| z_i - z_j \right\| \tag{12}$$

So we can know that the smaller value of V, the better clustering result is

Chapter 4.
Simulation Results

In the first simulation, we randomly generate a synthetic data set with $n = 1000$, $m = 2$, $k = 6$. Each dimension of each cluster of the data sets is generated as the normal distribution with the standard derivation $\sigma = 1$. Fig. 10 and Fig. 11 show the results of the number of cluster centers *k* and the validation index *v* with respect to different value of conflict factor λ. In the two figures, we can see that the minimum value of *v* will be obtained when the value of $k = 6$. Fig. 12 shows the clustering result, and these cluster centers are very close to the true centers of the six synthetic clusters in a data set.

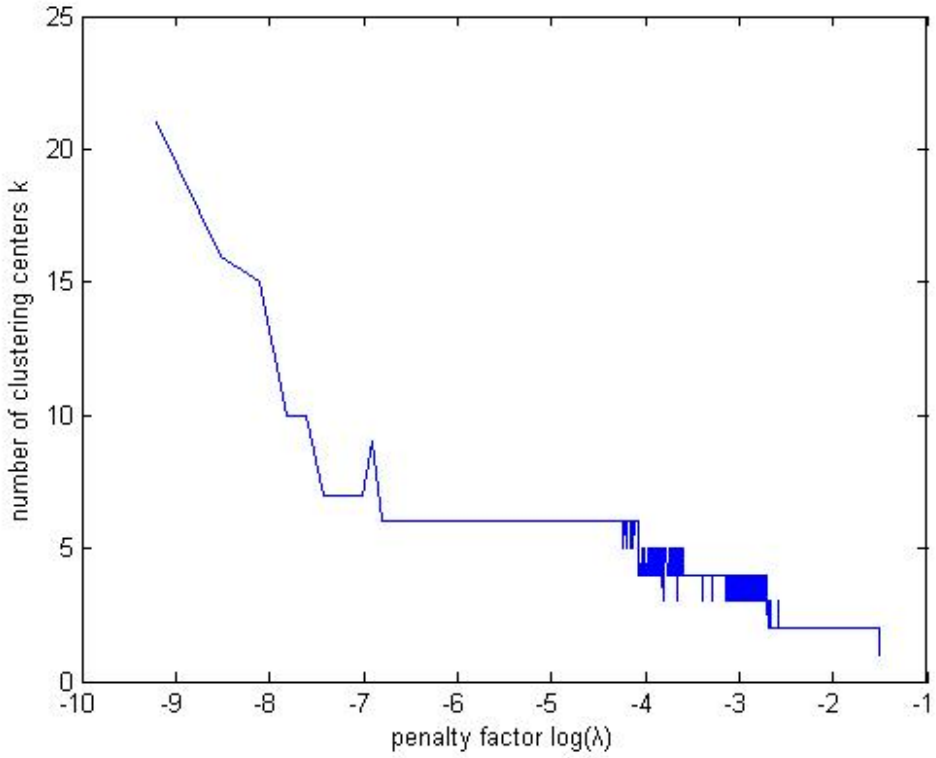

Fig. 10. The number of cluster centers with respect to different value of λ.

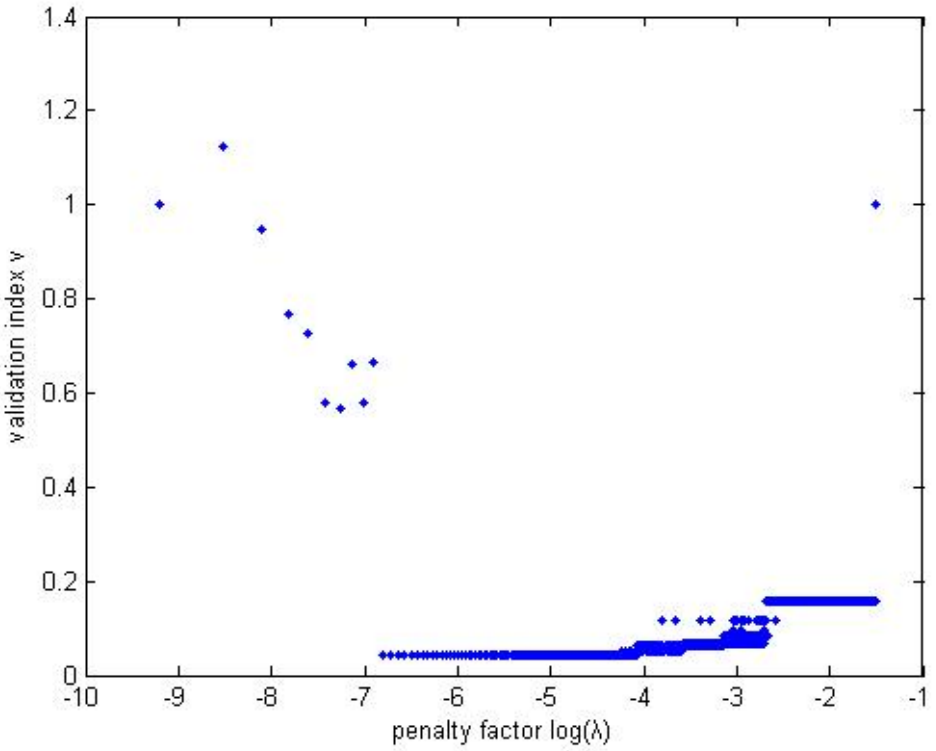

Fig. 11. The validation index V with respect to different value of λ.

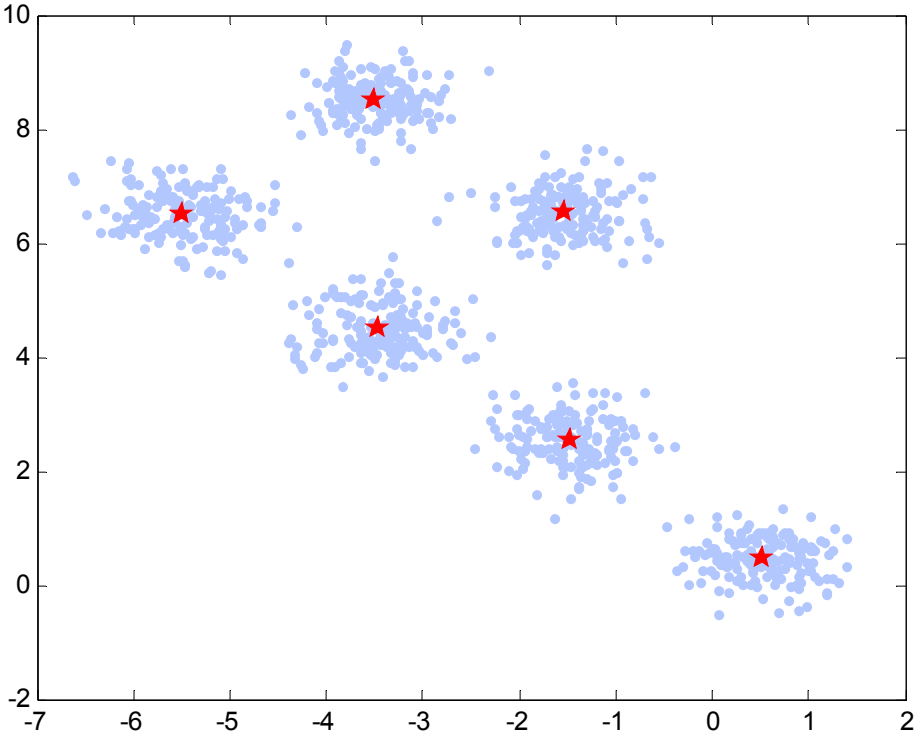

Fig. 12. The final position of the cluster centers.

For the comparison, we also use the standard *k*-means algorithm to generate the clustering result. We run the algorithms in 100 times, and Table 1 lists the comparison on times of the best positions of the determined cluster centers. In the Table 1, we can see that the proposed algorithm generates more consistent clustering results in different clustering runs. By contrast, the standard *k*-means algorithm only has 39% opportunity to obtain the best positions of the cluster centers because it is very sensitive to the initial cluster center. The best positions are list in the Table 2.

Table 1.The comparison on times of the best positions of determined cluster centers.

	The proposed algorithm	The standard *k*-means algorithm
Times of running algorithms	100	100
Times of obtaining the best positions	100	39
Proportion of obtaining the best positions	100%	39%

Table 2.The determined cluster centers by using the proposed algorithm and the standard *k*-means algorithm.

	The positions of true cluster centers	The best positions of determined cluster centers using the proposed algorithm	The best positions of determined cluster centers using the standard *k*-means algorithm
Cluster 1	(0.5,0.5)	(0.496,0.536)	(0.496,0.536)
Cluster 2	(-1.5,2.5)	(-1.469,2.482)	(-1.469,2.482)
Cluster 3	(-3.5,4.5)	(-3.501,4.538)	(-3.501,4.538)
Cluster 4	(-5.5,6.5)	(-5.495,6.488)	(-5.495,6.488)
Cluster 5	(-3.5,8.5)	(-3.473,8.521)	(-3.473,8.521)
Cluster 6	(-1.5,6.5)	(-1.477,6.515)	(-1.477,6.515)

Table 2 lists the comparison of best positions of determined cluster centers using the proposed algorithm and the true cluster centers. We can see that the best positions of determined cluster centers using the proposed algorithm are very close to the true positions of the cluster centers.

Table 3. The true value of k and the determined value of k by using the proposed algorithm in the different data sets.

Dimensions	Objects	The true value of k	The determined value of k by using the proposed algorithm
2	500	3	3
		6	6
	5000	3	3
		6	6
3	500	3	3
		6	6
	5000	3	3
		6	6
4	500	3	3
		6	6
	5000	3	3
		6	6

In the second simulation, we also randomly generate synthetic data sets with different number of dimensions, objects, and cluster centers to run the proposed algorithm. The result is listed in the Table 3, and we can see that the proposed algorithm performed very well. The best number of clusters k can be selected from the different data sets by using the proposed algorithm. Comparing with the standard k-means algorithm, it need to know the exactly value of k in advance. If the value of k is determined,

the standard *k*-means algorithm also does not exactly obtain the best cluster centers. But the proposed algorithm does not have such problems.

In the third simulation, we run the algorithm which Leng at al. [16] have proposed by using the above synthetic data sets. On the condition that the exactly value of ε and $\alpha_{\min}$ are given, it can obtain the best number and positions of cluster centers. By contrast, the proposed algorithm does not need to be given a parameter in advance. It can automatically find out the exactly value of λ by using the validation index. In the simulation result, it also can obtain the best number and positions of cluster centers as same as the algorithm in [16]. The algorithm in [16] must be given the values of two parameters in advance. So, it will need to spend some time re-learning the values of two parameters again when the data set is changed. But the proposed algorithm also does not have the problem.

Chapter 5.
Conclusion

In this thesis, we have presented a new algorithm for numerous data sets to determine the number and the positions of cluster centers. The proposed algorithm does not need to know the true value of k in advance, and it makes the clustering process not sensitive to the initial cluster centers. The proposed algorithm runs with the value of $k=1$ initially, and it increases the value of k by degrees. When the least objective function value is found, the best number and positions of cluster centers will be obtained. So we do not need to select a set of cluster centers randomly in advance, we just need to calculate the mean value of the data set and take it as the initial cluster center. The proposed algorithm aims at minimizing the objective function, which is the sum of the objective function of the standard k-means algorithm and the function of the summation of distances between cluster centers. Combined with the cluster validation technique, we can determine the exactly value of k and theirs positions of the cluster centers.

Our simulation results have shown the effectiveness of the proposed algorithm in different data sets. We can determine the best *k* and positions of the cluster centers in the data sets with different number of dimensions, objects, and cluster centers

Bibliography

[1] A.K. Jain and R.C. Dubes, *Algorithm for clustering Data*. Prentice Hall, 1988.

[2] V. Capoyleas, G. Rote, and G. Woeginger, "Geometric Clusterings," *J. Algorithms,* vol. 12, pp. 341-356, 1991.

[3] M.N. Murty, A.K. Jain, and P.J. Flynn, "Data Clustering: A Review," *ACM Computing Surveys,* vol. 31, no. 3, pp. 264-323, 1999.

[4] L. Kaufman and P.J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. New York: John Wiley&Sons, 1990.

[5] J.B. MacQueen, "Some Methods for Classification an Analysis of Multivariate Observations," *Proc. Fifth Symp. Math. Statistics and Probability*, vol. 1, AD 669871, pp. 281-297, 1967.

[6] J. M. Peña, J. A. Lozano, and P. Larrañaga, "An empirical comparison of four initialization methods for the k-means algorithm. Pattern Recognition Letters, "1999, 20 (10), 1027-1040.

[7] L. Bottou and Y. Bengio, "Convergence Properties of the k-Means Algorithms," *Advances in Neural Information Processing Systems*, 7, G.

Tesauro and D. Touretzky, eds., pp. 585-592, MIT Press, 1995.

[8] D. Pollard, “A Central Limit Theorem for k-Means Clustering,” *Annals of Probability*, vol. 10, pp. 919-926, 1982.

[9] G. Babu and M. Murty, “A near-optimal initial seed value selection in K-Means algorithm using a genetic algorithm,” *Pattern Recognition letters*, 14:763-769, 1993.

[10] P. S. Bradley, U. M. Fayyad, “Refining Initial Points for K-means, Clustering Advances in Knowledge Discovery and Data Mining”, *Proc. 15th International Conf. on Machine Learning*, 1998.

[11] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, A. Y. Wu, "An Efficient k-Means Clustering Algorithm: Analysis and Implementation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 881-892, 2002.

[12] J. He, M. Lan, C. L. Tan, S. Y. Sung, and H. BoonLow, “Initialization of Cluster refinement algorithms: a review and comparative study,” *Proceeding of International Joint Conference on Neural Networks*, 2004.

[13] F. Yuag, Z. H. Meng, H. X. Zhang, and C. R. Dong, “A New Algorithm to Get the Initial Centroids,” *Proceeding of the Third International Conference on Machine Learning and Cybernetics*, 2004.

[14] G. Hamerly and C. Elkan, "Learning the k in k-Means," *proc. 17th Ann. Conf. Neural Information Processing Systems*, 2003.

[15] M. J. Li, M. K. Ng, Y. Cheung, and J. Z. Huang, "Agglomerative fuzzy k-means clustering algorithm with selection of number of clusters," *IEEE Transactions on Knowledge and Data Eng.*, 2008.

[16] M. Leng, H. Tang, and X. Chen, "An Efficient K-means Clustering Algorithm Based on Influence Factors," *Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, 2007.

[17] H. Sun, S. Wang, and Q. Jiang, "FCM-Based Model Selection Algorithms for Determining the Number of Clusters," *Pattern Recognition*, vol. 37, pp. 2027-2037, 2004.