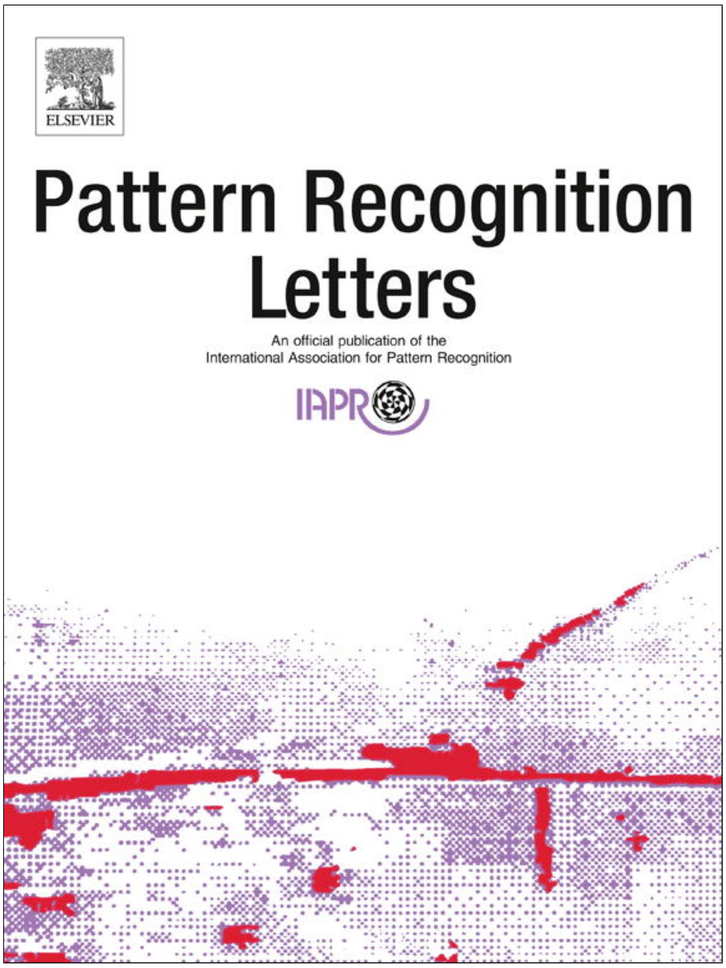

ELSEVIER

Pattern Recognition Letters

An official publication of the
International Association for Pattern Recognition

IAPR

(This is a sample cover image for this issue. The actual cover is not yet available at this time.)

Pattern Recognition Letters 34 (2013) 580–586

Contents lists available at SciVerse ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

k'-Means algorithms for clustering analysis with frequency sensitive discrepancy metrics

Chonglun Fang, Wei Jin, Jinwen Ma *

Department of Information Science, School of Mathematical Sciences and LMAM, Peking University, Beijing 100871, China

ARTICLE INFO

Article history:
Received 17 August 2010
Available online 28 November 2012

Communicated by S. Sarkar

Keywords:
Clustering analysis
k-Means
Cluster number
Competitive learning
Discrepancy metric

ABSTRACT

This paper proposes a new kind of k'-means algorithms for clustering analysis with three frequency sensitive (data) discrepancy metrics in the cases that the exact number of clusters in a dataset is not pre-known. That is, by setting the number k of seed-points for learning clusters to be larger than the true number k' of actual clusters in the dataset, i.e., $k > k'$, these algorithms can locate the centers of k' actual clusters by k' converged seed-points, respectively, with the extra $k - k'$ seed-points corresponding to empty clusters, namely containing no winning points in the competition according to the underlying frequency sensitive discrepancy metrics. It is demonstrated by the experiments on both synthetic and real-world datasets that these three new k'-means clustering algorithms can detect the number of actual clusters in a dataset with a classification accuracy rate as high as or higher than that of the original k'-means algorithm. Moreover, they converge more quickly than the original one.

1. Introduction

Clustering analysis, aiming at discovering the hidden data structure of a dataset, is a powerful technique applied in many areas of data analysis and information processing, such as data mining and compression, pattern recognition, vector quantization and signal processing, etc. Actually, it is a process of distributing the original data points into a number of distinctive clusters. Naturally, according to a certain discrepancy metric or distance, points in the same cluster are similar to each other, while points from different clusters are dissimilar. In fact, there have already been a variety of clustering algorithms in literature (e.g., MacQueen, 1967; Xu et al., 1993; Ester et al., 1996; Ma and Liu, 2007).

Mathematically, the clustering problem can be described as follows: given a dataset containing N points in the d-dimensional space, named by $\mathcal{R}_d$, as well as its cluster number $k(< N)$, we need to select k seed-points or cluster centers for k clusters in the data space with a certain data discrepancy metric or criterion according to which, the data points are assigned into one of the k clusters. As for the classical k-means algorithm (MacQueen, 1967), the criterion is just to minimize the sum of the mean squared distances between the data points and their nearest centers. Actually, each x_t is assigned to a cluster via the classification membership function given by

$$I(x_t, i) = \begin{cases} 1, & \text{if } i = \arg\min_j \|x_t - c_j\|^2,\ j = 1, 2, \ldots, k. \\ 0, & \text{otherwise}, \end{cases} \tag{1}$$

where $\|\cdot\|$ is the Euclidean norm. In each iteration, the ith cluster center c_i is updated by the following rule:

$$c_i = \frac{1}{|C_i|} \sum_{x_t \in C_i} x_t, \tag{2}$$

where $|C_i|$ denotes the number of the data points in C_i, i.e., Cluster i.

Due to its simplicity, the k-means algorithm is widely used for clustering analysis. Moreover, there are many investigations and improvements on the implementation of the k-means algorithm (e.g., Bradley and Fayyad, 1998; Kanungo et al., 2002). Generally, the k-means algorithm can lead to a good classification on a dataset when $k = k'$, i.e., k is selected to be the exact number of clusters in the dataset. So, the number of clusters, i.e., k, must be known in advance. Since the k-means algorithm just leads to k clusters for a dataset, the clustering result is certainly wrong if k is not equal to the exact number k' of clusters in the dataset. However, in a common case, we probably do not know the exact number of clusters in a dataset in advance. If k is not selected properly, the k-means algorithm will lead to a wrong clustering result. Therefore, it is rather important to select the correct number of clusters for a dataset. As a matter of fact, this is a rather difficult problem. In order to solve this problem, many approaches have been proposed and can be divided into two categories.

In the first category, one tries to increase the number of clusters one by one until the correct number of clusters is finally reached

* Corresponding author. Tel.: +86 10 62760609; fax: +86 10 62751801.
E-mail address: jwma@math.pku.edu.cn (J. Ma).

0167-8655/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.patrec.2012.11.004

(Zhang and Liu, 2002; Likas et al., 2003; Li and Ma, 2008). Generally, this kind of clustering algorithm starts from one cluster or a small number of clusters. According to a certain criterion, some improper cluster splits into two clusters and this procedure will go on step by step until the correct number of clusters is finally reached. In the second category, the rewarding and penalizing competitive learning mechanism is introduced. The main idea is to set a larger number of clusters at first. The rewarding and penalized competitive learning will push out the extra cluster centers, and keep the correct number of cluster centers in the field of the data. A typical example of this approach is the rival penalized competitive learning (RPCL) algorithm proposed in (Xu et al., 1993) and further investigated and developed in (Ma and Wang, 2006; Ma and Cao, 2006). Essentially in either case, the criterion for determining the correct number of clusters for a dataset plays an important role. In fact, many such criteria have been proposed from different aspects, such as Akaike's Information Criterion (AIC) (Akaike, 1974), Bayesian Inference Criterion (BIC) (Schwarz, 1978), Minimum Message Length (MML) (Wallace and Dowe, 1999), and Bayesian Ying-Yang (BYY) harmony learning principle (Ma et al., 2004; Cheung, 2003; Ma and Liu, 2007). However, these approaches generally involve in complicated mathematical models as well as large amounts of computation.

Recently, Zalik (2008) proposed a new kind of k-means algorithm called k'-means algorithm which essentially implements a rewarding and penalizing competitive learning mechanism for selecting the correct number of clusters in a dataset by just using a new kind of data discrepancy metric between an data point and a cluster center instead of the Euclidean distance used in Eq. (1). Firstly, it sets the number of initial cluster centers to be larger than the correct one. With the special discrepancy metric, the competition learning is implemented among these estimated clusters such that those extra seed-points will finally have no winning data point and correspond to empty clusters, while the correct number of clusters will be finally obtained. It was demonstrated by the experiments in (Zalik, 2008) that the k'-means algorithm can efficiently determine the number of actual clusters in some synthetic and real-world datasets. However, this kind of data discrepancy metric has not been investigated from different ways. Moreover, it has been further found by the experiments that this k'-means algorithm is not so good in the large-scale real-world dataset. Furthermore, Fang and Ma (2009) proposed another k'-means algorithm based on a data discrepancy metric with a penalty of average cluster weight, but its performance is not so stable with the initial values of the seed-points.

In the current paper, we propose a new kind of k'-means algorithms with three frequency sensitive discrepancy metrics. Although these k'-means algorithms perform in the same way as the classical k-means algorithm, their learning mechanisms are similar to that of the rival penalized competitive learning algorithm which tries to push the extra seed-points to infinity and thus make the corresponding clusters empty. So, it is not necessary to know the exact number of clusters in advance. But we should have an overestimate of k' in advance, which is relatively easy in practice. The experiments on both synthetic and real-world datasets show that these three new k'-means algorithms can detect the number of actual clusters in a dataset with a classification accuracy rate as high as or better than that of the original k'-means algorithm. Moreover, they converge more quickly than the original one.

The rest of this paper is organized as follows. The new frequency sensitive discrepancy metrics and the corresponding k'-means algorithms are presented in Section 2. The experimental results are contained in Section 3, including clustering analysis on both the synthetic and real-world datasets as well as an application to unsupervised color image segmentation. Finally, a brief conclusion is made in Section 4.

2. Three frequency sensitive data metrics for the k'-means algorithm

As described in (Zalik, 2008), the k'-means algorithm works in the same paradigm as the classical k-means algorithm, but differs in the expression of the cluster membership function $I(x_t, i)$ defined via a specific discrepancy metric between an input data point x_t and the cluster center c_i. In this situation, k is assumed to be larger than the exact number k' of the clusters in the dataset, i.e., $k > k'$. In order to eliminate those extra clusters, we should introduce the rewarding and penalizing competitive learning mechanism into the learning process through the cluster membership functions $I(x_t, i)$ defined via the underlying discrepancy metric. Actually, such a data discrepancy metric should have the following two characteristics in the clustering or learning process:

(i) The clusters with few data points should be penalized and become empty in the sequential iterations.
(ii) The cluster seed-points try to locate in the places where the data points are densely accumulated.

In fact, a typical example of this kind of discrepancy metric is just the data metric $dm(x_t, c_i)$ used in the original k'-means algorithm (Zalik, 2008), which is defined by

$$dm(x_t, c_i) = \|x - c_i\|^2 - E\log_2 p_i, \tag{3}$$

where $E > 0$ is a constant serving as a penalty factor and $p_i = P(C_i)$ is the frequency that an input data point is in the C_i cluster (subset).

Inspired by this special discrepancy metric, we investigate this kind of discrepancy metric from different points of view and construct the three feasible ones as follows.

Our first discrepancy metric is defined by

$$d_1(x, c_i) = \|x - c_i\|^2 + \lambda/p_i, \tag{4}$$

where λ serves as a penalty factor, being a positive constant. When some p_i becomes zero, $d_1(x, c_i)$ is considered as the positive infinity for any input data point x and this cluster becomes empty and withdraws from the competition. Clearly, the first term in Eq. (4) is just the mean square error between the data points x and the corresponding cluster centers c_i, while the second term is based on the frequency of the data point belong to the ith cluster. Functionally, the first term tries to attract the seed-points to the centers of the actual clusters, respectively, while the second term tries to penalize the clusters with few data points and make them empty. In comparison with Zalik's metric given by Eq. (3), we just use $1/p_i$ instead of $\log_2 p_i$. In this way, the penalty mechanism may be more active and efficient.

We further define the second discrepancy metric by

$$d_2(x, c_i) = p_i\|x - c_i\|^2 + \lambda/p_i, \tag{5}$$

where $\lambda > 0$ is also a penalty factor. Clearly, the second term is as the same as that of the first discrepancy metric $d_1(x, C_i)$, but the first term is the product of the C_i's frequency p_i and the point mean square error $\|x - c_i\|^2$. If p_i is small, the first term becomes small, while the second term is large. However, if p_i is large, the second term may be too small. So, we can use this kind of discrepancy metric to make a balance between the first and second items, which makes the rewarding and penalizing competitive learning in a more harmonic way.

Finally, we define the third discrepancy metric by

$$d_3(x, c_i) = p_i\|x - c_i\|^2 - \lambda\log p_i, \tag{6}$$

where $\lambda > 0$ is again a penalty factor. In comparison with the second discrepancy metric, we just use $-\log_2 p_i$ in stead of $1/p_i$. If $p_i = 0,\ -\log p_i$ is also considered as the positive infinity. Clearly, the second term is restored to that of Zalik's metric. Since p_i is now acted in the first term, this discrepancy metric may lead to another good manner of rewarding and penalizing competitive learning for some situations.

As these three discrepancy metrics are sensitive to the frequencies of the clusters in the competition, we refer to them as the frequency sensitive discrepancy metrics in this paper. With each of the three frequency sensitive discrepancy metrics, say $d(x_t, c_j)$, we can compute the corresponding cluster membership function by

$$I(x_t, i) = \begin{cases} 1 & \text{if } i = \arg\min d(x_t, c_j),\ j = 1, \ldots, k; \\ 0 & \text{otherwise.} \end{cases} \qquad (7)$$

That is, x_t belongs to C_i if and only if $I(x_t, i) = 1$. Then, we can implement the k'-means algorithm with a such cluster membership function $I(x_t, i)$. For clarity, we further refer to the k'-means algorithm with the ith frequency sensitive discrepancy metric as the k'-means algorithm i in the following analysis and discussions. Just as the original k'-means algorithm, these k'-means algorithms with the three frequency sensitive discrepancy metrics also have a rewarding and penalizing competitive learning paradigm. For certain clusters, there may be no data point belonging to them after a number of iterations. According to each of the three discrepancy metric, these clusters will be always empty in the sequential iterations. It can be demonstrated by the experiments in the next section that all the extra clusters can become empty after the convergence of the k'-means algorithm. It can be also found that the penalty factor λ should be carefully selected. If λ is too large, there may be only one left cluster containing all the data points, with all the other clusters being empty. If λ is too small, the extra clusters may be always kept. The penalty factor will be discussed in detail in Section 3.

3. Experimental results

In this section, various experiments on both synthetic and real-world datasets are carried out to test the classification performances of the k'-means algorithms with three frequency sensitive discrepancy metrics, being compared with those of the original k'-means algorithm, DBSCAN (Ester et al., 1996), as well as MML and AIC based clustering methods. Moreover, these k'-means algorithms are successfully applied to unsupervised color image segmentation. In each new discrepancy metric, the penalty factor λ is the unique parameter and should be selected properly. As p_i is a frequency being varied in $[0, 1]$, λ may depend on the scale of sample data. In order to overcome this incertainty and find out the optimal value of λ, we can normalize the original sample data via a certain linear scale transform. That is, the processed sample data will have zero mean and unit variance. Thus, in each experiment, the sample data will be normalized firstly and the algorithm will be implemented on the normalized data.

3.1. On the synthetic datasets

3.1.1. Classification performances on four typical synthetic datasets

We begin to test the classification performances of the new k'-means algorithms with the three frequency sensitive discrepancy metrics on four typical synthetic datasets $\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, \mathcal{S}_4$, which are respectively shown in Fig. 1. Typically, they are generated from a mixture of four or three bivariate Gaussian distributions on the plane coordinate system (i.e., $d = 2$). Thus, a cluster or class takes the form of a Gaussian distribution. Particularly, all the Gaussian distributions are cap-shaped, that is, their covariance matrices have the form of $\sigma^2 I$, where σ is the standard variance. For the first three datasets, four Gaussian distributions, all with 300 sample points, are all located at $(-1, 0)$, $(1, 0)$, $(0, 1)$ and $(0, -1)$, respectively, and their standard variances σ keep the same, but vary with the dataset. Actually, σ takes the values of 0.2, 0.3, 0.4 for $\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3$, respectively. In this way, the degree of overlap among the actual clusters or Gaussian distributions in the dataset increases considerably from $\mathcal{S}_1$ to $\mathcal{S}_3$ and therefore the corresponding classification problem becomes more complicated. As for $\mathcal{S}_4$, we keep only three Gaussian distributions located at $(1, 0)$, $(0, 1)$ and $(0, -1)$, respectively. The standard variance of the Gaussian distribution at $(1, 0)$ is 0.3, while those of the other two Gaussian distributions are 0.2. In this case, three clusters have different numbers of sample points. In fact, the numbers of sample points for Gaussian distributions at $(1, 0)$, $(0, 1)$ and $(0, -1)$, are 400, 300, and 200, respectively. Therefore, $\mathcal{S}_4$ represents the asymmetric situation where the clusters do not take the same shape, also with different numbers of sample points.

We implement three new k'-means algorithms on each of the four datasets with $k = 8$. The penalty factor λ is selected by 0.2, 0.05, and 0.4 for the k'-means algorithms 1, 2, and 3, respectively. The seed-points are randomly initialized within the field of the sample data. It is found by the experiments that in any case, four or three seed-points can be finally located accurately at the centers of the actual clusters or Gaussian distributions, while the other four or five extra seed-points are eliminated automatically during the iterations. To demonstrate the stability of the classification performance on these datasets, we further implement each new k'-means algorithm for 100 times with different randomly selected initial values of the seed-points on each dataset and compute the average Classification Accuracy Rate (CAR) (with the standard variance), which are given in Table 1. For comparison, we also give the average CARs of the classical k-means algorithm ($k = k'$) and the original k'-means algorithm ($E = 0.65,\ k = 8$) on the four datasets in Table 1. Moreover, we also compute the average implementation times (seconds) of these algorithms over 100 trials on each of the four datasets, which are given in Table 2. It should be noted that all the experiments have been implemented on a notebook computer of Lenovo Thinkpad T420s in Matlab environment.

From the detailed numbers listed in Table 1, we can find that all the new k'-means algorithms can detect the number of actual clusters and lead to a rather high average CARs on each dataset. It is clear that the average CAR of each algorithm decreases slightly as the degree of overlap among the actual clusters or classes in the dataset becomes higher, that is, the structure of the dataset becomes more complicated. However, even when the actual clusters take different shapes and have different numbers of sample points like $\mathcal{S}_4$, the average CARs of these algorithms are still very high. The experimental results also demonstrate that these three k'-means algorithms lead to a similar average CAR on each dataset. That is, the three discrepancy metrics actually implement the same rewarding and penalizing competitive learning mechanism for the k'-means clustering. Moreover, the classification performances of these three k'-means algorithms can be even better than those of the original k'-means algorithm and classical k-means algorithm in some complicated cases. On detecting the number of actual clusters in a dataset, our new k'-means algorithms always converge correctly, but the original k'-means algorithm lead to a wrong result in a few times (1–4) over 100 experimental results on each of $\mathcal{S}_1$, $\mathcal{S}_2$, and $\mathcal{S}_3$. Thus, our new k'-means algorithms are more stable than the original k'-means algorithm on the cluster number detection. On the other hand, these k'-means algorithms not only detect the number of actual clusters in the dataset, but also lead to a higher CAR than the classical k-means algorithm. Oppositely, the RPCL algorithm detects the number of actual clusters in a dataset with a lower CAR than that of the classical k-means algorithm due to the cluster center deviation via the rival penalizing mechanism (Xu et al., 1993). In this sense, these k'-means algorithms can even be better than the RPCL algorithm.

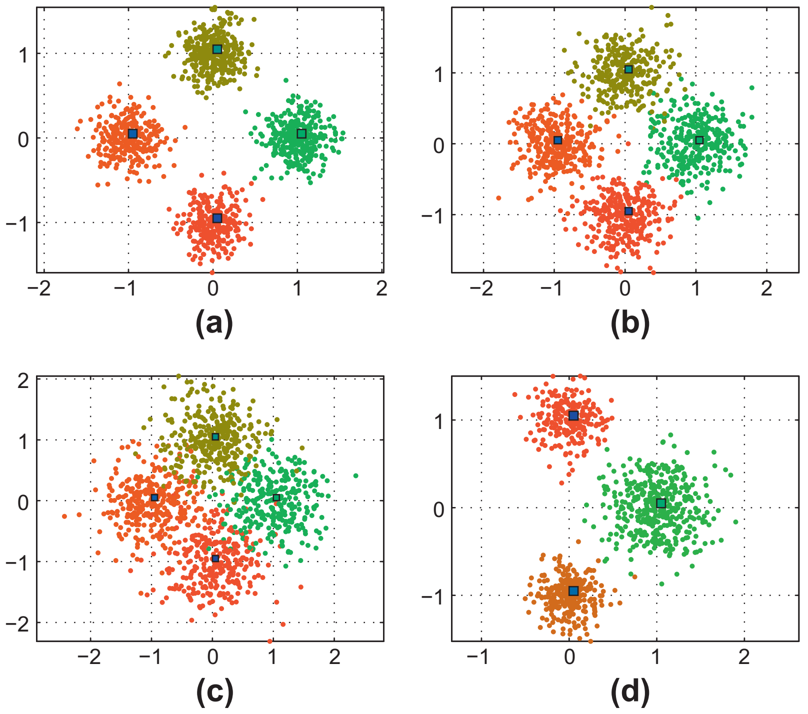

Fig. 1. The sketches of four typical synthetic datasets used in the experiments: (a) $\mathcal{S}_1$; (b) $\mathcal{S}_2$; (c) $\mathcal{S}_3$; (d) $\mathcal{S}_4$.

Table 1
The average CARs of the three new k'-means algorithms vs those of the original k'-means algorithm and classical k-means algorithm on the four synthetic datasets. For short, the new k'-means algorithm i is referred to as $k'mi$, the original k'-means algorithm is referred to as $ok'm$, and the classical k-means algorithm is referred to as km.

Algorithm	$\mathcal{S}_1$ (%)	$\mathcal{S}_2$ (%)	$\mathcal{S}_3$ (%)	$\mathcal{S}_4$ (%)
$k'm1$	99.92 ± 0.00	98.37 ± 0.05	92.12 ± 0.20	98.91 ± 0.09
$k'm2$	99.92 ± 0.00	98.46 ± 0.04	91.96 ± 0.13	97.92 ± 0.14
$k'm3$	99.92 ± 0.00	98.38 ± 0.05	92.04 ± 0.29	98.56 ± 0.00
$ok'm$	99.92 ± 0.00	98.42 ± 0.00	92.06 ± 0.23	98.56 ± 0.00
km	99.92 ± 0.00	98.42 ± 0.00	91.94 ± 0.11	98.33 ± 0.09

Table 2
The average implementation times (seconds) of the three new k'-means algorithms vs those of the original k'-means algorithm and classical k-means algorithm on the four synthetic datasets. The algorithms are shortly denoted as above.

Algorithm	$\mathcal{S}_1$	$\mathcal{S}_2$	$\mathcal{S}_3$	$\mathcal{S}_4$
$k'm1$	0.00697	0.01223	0.02280	0.00848
$k'm2$	0.00644	0.01024	0.01542	0.00701
$k'm3$	0.00668	0.01158	0.01844	0.00844
$ok'm$	0.00804	0.01308	0.02456	0.00926
km	0.00201	0.00213	0.00297	0.00164

On the other hand, from the detailed numbers listed in Table 2, we can further find that the implementation times of the three new k'-means algorithms are obviously less than that of the original k'-means algorithm on each of the four datasets. So, we can consider that these new k'-means algorithms converge more quickly than the original one. Among the three new k'-means algorithms, the second one provides the fastest convergence, while the first one provides the slowest convergence. Although the differences between their implementation times (or convergence speeds) are not relatively large on each of these dataset, it can be observed from the other experiment on a large dataset that they can be relatively large. Specifically, the implementation time of the second k'-means algorithm is just about one half of that of the first k'-means algorithm, and one third of that of the original k'-means algorithm. On the other hand, since the k'-means algorithms involve more computation to implement the rewarding and penalizing competitive learning mechanism, they converge much more slowly than the classical k-means algorithm. Moreover, we can also find that the implementation time of the k'-means algorithm depends on the structure or complexity of the dataset, but that of the classical k-means algorithm does not.

For comparison, we also implement DBSCAN on these four datasets for clustering analysis. In fact, DBSCAN has been considered as one of the best clustering methods. Theoretically, DBSCAN detects the cluster by the stable sample density and its expansion. In such a manner, DBSCAN is able to detect actual clusters in a dataset only when they are well separated. It is found by the experiments that DBSCAN can detect the four actual clusters in $\mathcal{S}_1$. However, DBSCAN cannot detect specific actual clusters and only recognize all the dataset as one cluster in each case of $\mathcal{S}_2$, $\mathcal{S}_3$ and $\mathcal{S}_4$, where the overlap between the actual clusters is obviously high in certain degree. Even in the case of $\mathcal{S}_1$, the average CAR of DBSCAN is just 97%, which is slightly lower than those of the new k'-means algorithms. Moreover, its average implementation time is 0.01647 (s), which is over two times of that of each new k'-means algorithm.

3.1.2. Comparison with classical cluster number selection criteria

Furthermore, we try to compare these new k'-means algorithms with two classical clustering criteria AIC and MML on detecting the number of actual clusters in a dataset. Here, we use the same group of datasets used in (Oliver et al., 1996). Actually, each dataset consists of 100 samples generated from a mixture of three 2-dimensional Gaussian distributions with the same covariance matrix. The mean vectors of the three Gaussian distributions are $(0,0)$, $(2,2\sqrt{3})$, and $(4,0)$, respectively, while the covariance matrix also takes the form $\sigma^2 I$. By varying σ with the values of 0.67, 1, 1.2 and 1.33, respectively, we can get four datasets with differently complicated structures. To check the performances of the three k'-means algorithms on detecting the correct number of clusters in each dataset with the specific value of σ, we implement the three k'-means algorithms with $k = 5$ for 100 times from different randomly selected initial values of the seed-points and then compute the numbers of different detecting results for the cluster number. We summarize these numbers with those of the MML and AIC based clustering methods given in (Oliver et al., 1996) in Table 3. It can be seen clearly from Table 3 that these new k'-means algorithms always lead to a better result than the two classical cluster number selection criteria based methods.

3.1.3. Further discussions

We finally discuss the (unsupervised) classification performances of these three new k'-means algorithms on the general datasets. Experimentally, it can be easily found that the correct cluster number selection of a k'-means algorithm strongly depends on the overlap among the actual clusters in a dataset. Particularly, as long as the overlap among the actual clusters is low enough, the k'-means algorithm can lead to the correct cluster number selection. Moreover, the shapes of the actual clusters are also important. When the actual clusters take a shape of sphere or similar form, it is easy for a k'-means algorithm to detect the cluster number. Oppositely, when they are very flat or bend, it may be difficult for a k'-means algorithm to detect the cluster number. Since the actual clusters generally take a shape of sphere like a Gaussian distribution, these k'-means algorithms can be effectively applied in practice.

The penalty factor λ also plays an important role on the implementation of these k'-means algorithms. As the three discrepancy metrics take the different forms, λ may have different feasible values for three new k'-means algorithms on detecting the number of actual clusters in a dataset. By the experiments on the four typical synthetic datasets, we have found that the feasible intervals of λ for the k'-means algorithms 1, 2, and 3 are $[0.10, 0.30]$, $[0.02, 0.10]$, and $[0.16, 0.60]$, respectively. It is clear that the feasible intervals of λ for these three k'-means algorithms are quite different, but each of them is quite large for its selection. As for the setting of k, it is required to be larger than k'. But when it is too larger than k', the k'-means algorithms may lead to a wrong result.

Since the proposed k'-means algorithms are new versions of k-means algorithm, it can be easily verified that they also have the computation complexity $O(Nkdt)$, where d is the dimensionality of the sample points or inputs, and t is the number of iterations. Thus, their implementation times are linear with the sample size N, which is demonstrated well by our simulation experiments with different sample sizes. Moreover, they are also linear with the dimensionality of the sample points so that the k'-means algorithms can be implemented well on the high-dimensional datasets.

In a summary, these new k'-means algorithms can be implemented effectively for cluster number selection and classification as long as the actual clusters are separated in a certain degree and taken a shape of sphere or similar form. It is demonstrated by the experiments that they are even better than DBSCAN as well as MML and AIC based clustering methods on detecting the actual clusters in a complicated dataset. Moreover, they converge more quickly than the original k'-means algorithm.

Table 3
The cluster numbers detected by the new k'-means algorithms as well as the MML and AIC based clustering methods on the four datasets.

Algorithm	k	$\sigma = 0.67$	$\sigma = 1$	$\sigma = 1.2$	$\sigma = 1.33$
k'-means Algorithm 1					
	1	0	0	0	0
	2	0	0	0	0
	3 (True)	100	95	82	78
	4	0	5	16	18
	5	0	0	2	4
k'-means Algorithm 2					
	1	0	0	0	0
	2	0	0	0	0
	3 (True)	100	96	75	68
	4	0	4	22	28
	5	0	0	3	4
k'-means Algorithm 3					
	1	0	0	0	0
	2	0	0	0	0
	3 (True)	100	100	84	78
	4	0	0	16	14
	5	0	0	0	8
MML					
	1	0	0	16	41
	2	0	21	44	41
	3 (True)	96	72	39	17
	4	4	7	1	1
	5	0	0	0	0
AIC					
	1	0	0	6	20
	2	0	8	29	39
	3 (True)	73	65	48	27
	4	13	19	9	5
	5	14	8	8	9

3.2. On the real-world datasets

In this subsection, we continue to test these new k'-means algorithms on five typical real-world datasets from <http://mlearn.ics.uci.edu/databases/>. We firstly consider the wine dataset. Actually, the wine dataset contains 178 sample points of three types of wine. Each sample point is 13-dimensional and the numbers of sample points in the three classes are 48, 71, and 59, respectively. We implement these new k'-means algorithms on the wine data with $k = 6$. The experimental results show that the three classes of wine can be always detected. Moreover, the classification accuracy rates of the k'-means algorithms 1, 2, and 3 are 98.31% (there are 3 errors), 97.75% (4 errors), and 97.26% (5 errors), respectively. These are almost as high as those of the original k'-means algorithm (4 errors) given in (Zalik, 2008) and the method of linear mixing kernels (4 errors) given in (Roberts et al., 2000). As for the classical k-means algorithm with $k = 3$, there are at least 9 errors in the resulted classification. Therefore, our new k'-means algorithms perform very well on the unsupervised classification of the wine data. But DBSCAN cannot detect the actual clusters or classes in this as well as the following real-world datasets.

We further consider the protein localization sites dataset which mainly contains three classes called Cp, Im, and Pp, respectively. Each sample point is 7-dimensional and the numbers of sample points in these three classes are 143, 77, and 52, respectively. We here consider the dataset of these three classes and implement the new and original k'-means algorithms with $k = 8$. The first and second k'-means algorithms can detect the three actual clusters or classes of protein localization sites (i.e., $k' = 3$) with the CARs being 93.75% (there are 17 errors) and 93.34% (18 errors), respectively. However, the third and original k'-means algorithms cannot detect the three actual clusters in any case. So, in this particular case, the k'-means algorithms 1 and 2 are superior to the third and original k'-means algorithm on determining the number of the actual clusters, which may be caused by the different penalizing mechanisms of two kinds of the second terms in the discrepancy metrics used in the k'-means algorithms.

Furthermore, we consider the Wisconsin Breast Cancer Dataset which contains 699 9-dimensional sample points belong to the two classes called benign and malignant, respectively. In fact, there are 458 sample points in the benign class, and 241 sample points in

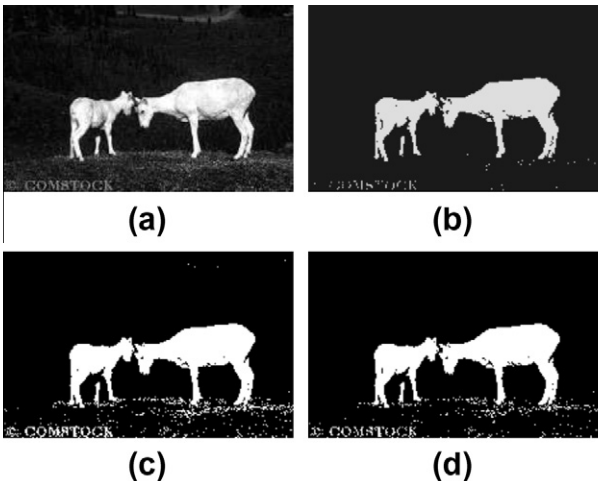

Fig. 2. The unsupervised segmentation results on the color image of two goats. (a) The original image. (b)–(d) The segmentation results of the k'-means Algorithm 1, 2, and 3, respectively.

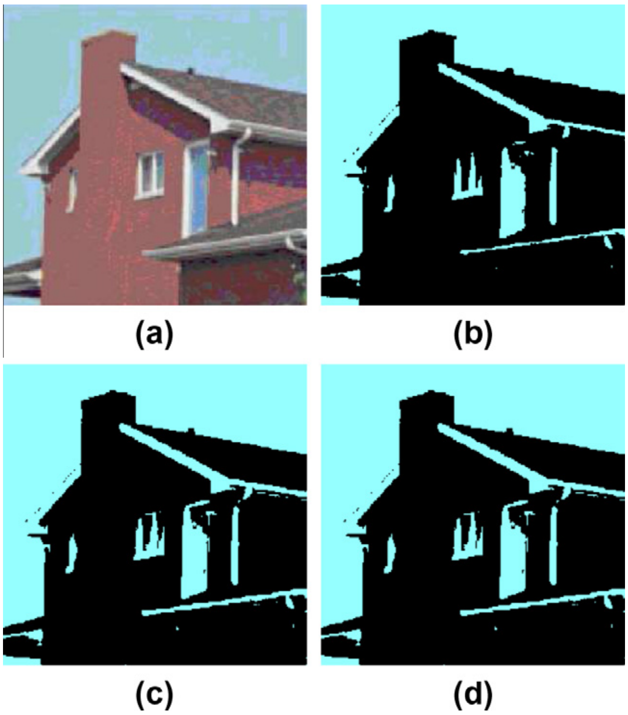

Fig. 3. The unsupervised segmentation results on the color image of one house. (a) The original image. (b)–(d) The segmentation results of the k'-means Algorithm 1, 2, and 3, respectively.

the malignant class. Actually, there are 16 missing values in some sample points and we just set them as zeros. On this dataset, we implement these new k'-means algorithms with $k = 8$. All the three k'-means algorithms can detect the two actual clusters or classes of breast cancers (i.e., $k' = 2$). Moreover, the CARs of the k'-means algorithms 1, 2, and 3 are 94.85% (36 errors), 96.57% (24 errors), and 96.14% (27 errors), respectively. Although the original k'-means algorithm can detect the two actual classes, its CAR is only 91.27 % (61 errors). So, our new k'-means algorithms are much better than the original k'-means algorithm on the classification on this real-world dataset.

For the higher dimensional real-world data, we turn to the Wisconsin Diagnostic Breast Cancer (WDBC) dataset. It consists of 569 33-dimensional sample points also belong to the two class: benign and malignant. In this case, there are 357 sample points in the benign class, and 212 sample points in the malignant class. On this WDBC dataset, we implement these new k'-means algorithms with $k = 10$. Again, the three k'-means algorithms can detect the two actual classes of breast cancers. The classification accuracy rates of the k'-means algorithms 1, 2, and 3 are 90.33% (there are 55 errors), 90.69% (53 errors), and 90.33% (55 errors), respectively. But on this dataset, the original k'-means algorithm (Zalik, 2008) cannot converge to any reasonable result. Therefore, these new k'-means algorithms are considerably better than the original k'-means algorithm on the classification of those high dimensional real-world data.

We finally consider another higher dimensional real-world dataset called Landsat satellite dataset. For simplicity, we only consider the first and second classes of the original dataset. So, the Landsat satellite dataset we consider here consists of 1551 37-dimensional sample points belong to two classes called red soil and cotton crop, respectively. Particularly, there are 1072 sample points in the red soil class, and 479 sample points in the cotton crop class. On this Landsat satellite dataset, we implement these new k'-means algorithms with $k = 10$. In this situation, the three k'-means algorithms can detect the two actual classes of objects: red soil and cotton crop. Moreover, the classification accuracy rates of the k'-means algorithms 1, 2, and 3 are 92.39% (118 errors), 96.71% (51 errors), and 96.58% (53 errors), respectively. However, the classification accuracy rate of the original k'-means algorithm (Zalik, 2008) is only 91.36 % (134 errors). So, these new k'-means algorithms are much better than the original k'-means algorithm on the classification of those high dimensional real-world data.

Based on the above experimental results on the real-world datasets, it can be seen that our new k'-means algorithms generally lead to a good clustering result for both cluster number detection and classification performance. Moreover, as the dimensionality of data becomes higher, they perform much better than the original k'-means algorithm.

3.3. Application to unsupervised color image segmentation

In this subsection, for practical usage and test, we apply our new k'-means algorithms to unsupervised color image segmentation. In fact, image segmentation is a fundamental problem in image processing and can be treated as a clustering problem. In computer vision, it is usual that the number of objects in an image is not pre-known. Thus, the image segmentation is generally in an unsupervised mode to automatically determine the number of objects and background in the image, which is still a rather difficult task in image processing. However, these k'-means algorithms really provide a new tool for unsupervised image segmentation. Here, we try to apply these algorithms to unsupervised color image segmentation on two typical color images called two goats and one house shown in Fig. 2(a) and Fig. 3(a), respectively. In these two color images, each pixel is a 3-dimensional datum, corresponding to its RGB coordinates, and these pixel data are normalized with certain linear transformation at first. We then implement the new k'-means algorithms on the normalized data of the image pixels for clustering or unsupervised classification with $k = 6$. As a result, 2 clusters are remained after the convergence and the segmentation results of the two color images by the three new k'-means algorithms are shown in Figs. 2 and 3, respectively. Compared with the original images, our segmentation results are quite good. For the image of two goats, we can observe that the two objects are finally detected and matched well, which means that the segmentation result coincides with the actual objects. For the image of the house, the house and the sky are separated clearly.

Therefore, our new k'-means algorithms can be successfully applied to unsupervised color image segmentation.

4. Conclusions

We have investigated the data discrepancy metric for the k'-means algorithm from different points of view and constructed three different discrepancy metrics according to which the new k'-means algorithms are established. These new k'-means algorithms keep a simple learning rule, but have a rewarding and penalizing mechanism being similar to that of the rival penalized competitive learning (RPCL) algorithm. It is demonstrated by the experiments on both the synthetic and real-world datasets that these new k'-means algorithms can detect the number of actual clusters in a dataset with a classification accuracy rate as high as or better than those of the original k'-means algorithm those of the original k'-means algorithm, DBSCAN, as well as MML and AIC based clustering methods. Moreover, they converge more quickly than the original k'-means algorithm. Finally, they are successfully applied to unsupervised color iamge segmentation.

References

Akaike, H., 1974. A new look at the statistical model identification. IEEE Trans. Autom. Control AC-19, 716–723.

Bradley, P.S., Fayyad, U.M., 1998. Refining initial points for *k*-means clustering. In: Proc. 5th Internat. Conf. on Machine Learning (ICML'98). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 91–99.

Cheung, Y.M., 2003. k*-means: a new generalized *k*-means clustering algorithm. Pattern Recognition Lett. 24 (15), 2883–2893.

Ester, M., Kriegel H.P., Sander J., Xu, X., 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proc. 2nd Internat. Conf. on Knowledge Discovery and Data Mining (KDD'96), Portland, OR, USA, pp. 226–231.

Fang, C., Ma, J. 2009. A novel k'-means algorithm for clustering analysis. In: Proc. 2nd Internat. Conf. on Biomedical Engineering and Informatics (BMEI, 2009), 17–19, October, 2009, Tianjin, China.

Kanungo, T., Mount, D., Netanyahu, N., et al., 2002. An efficient *k*-means clustering algorithm: analysis and implementation. IEEE Trans. Pattern Anal. Machine Intell. 24 (7), 881–892.

Li, L., Ma, J., 2008. A BYY scale-incremental EM algorithm for Gaussian mixture learning. Appl. Math. Comput. 205, 832–840.

Likas, A., Vlassis, N., Verbeek, J.J., 2003. The global *k*-means clustering algorithm. Pattern Recognition 36 (2), 451–461.

Ma, J., Cao, B., 2006. The mahalanobis distance based rival penalized competitive learning algorithm. Lect. Notes Comput. Sci. 3971, 442–447.

Ma, J., Wang, T., Xu, L., 2004. A gradient BYY harmony learning rule on Gaussian mixture with automated model selection. Neurocomputing 56, 481–487.

Ma, J., Wang, T., 2006. A cost-function approach to rival penalized competitive learning (RPCL). IEEE Trans. Systems Man Cybernet. 36 (4), 722–737.

Ma, J., Liu, J., 2007. The BYY annealing learning algorithm for Gaussian mixture with automated model selection. Pattern Recognition 40 (7), 2029–2037.

MacQueen, J.B. 1967. Some methods of classification and analysis of multivariate observations. In: Proc. 5th Berkeley Symposium on Mathemtical Statistics and Probability, pp. 281–297.

Oliver, J., Baxter, R., Wallace, C., 1996. Unsupervised learning using mml. In: Machine Learning: Proc. 13th Internat. Conf. (ICML 96). Morgan Kaufmann Publishers, pp. 364–372.

Roberts, S.J., Everson, R.M., Rezek, I., 2000. Maximum certainty data partitioning. Pattern Recognition 33 (5), 833–839.

Schwarz, G., 1978. Estimating the dimension of a model. Ann. Statist. 6 (2), 461–464.

Wallace, C.S., Dowe, D.L., 1999. Minimum message length and Kolmogorov complexity. Comput. J. 42, 270–283.

Xu, L., Krzyzak, A., Oja, E., 1993. Rival penalized competitive learning for clustering analysis, RBF net, and curve detection. IEEE Trans. Neural Networks 4 (4), 636–649.

Zalik, K.R., 2008. An efficient k'-means clustering algorithm. Pattern Recognition Lett. 29 (9), 1385–1391.

Zhang, Y., Liu, Z., 2002. Self-splitting competitive learning: a new on-line clustering paradigm. IEEE Trans. Neural Networks 13 (2), 369–380.