Pattern Recognition Letters 54 (2015) 50–55

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

PCA-guided search for *K*-means ☆

Qin Xu [a,*], Chris Ding [b], Jinpei Liu [c], Bin Luo [a]

[a] School of Computer Science and Technology, Anhui University, Hefei, Anhui 230601, China
[b] Department of Computer Science and Engineering, University of Texas, Arlington, Texas 76019, USA
[c] School of Business, Anhui University, Hefei, Anhui 230601, China

ARTICLE INFO

Article history:
Received 27 June 2014
Available online 23 December 2014

Keywords:
K-means
Principal component analysis
Cluster centroid initialization
Clustering

ABSTRACT

K-means is undoubtedly the most widely used partitional clustering algorithm. Unfortunately, due to the non-convexity of the model formulations, expectation-maximization (EM) type algorithms converge to different local optima with different initializations. Recent discoveries have identified that the global solution of *K*-means cluster centroids lies in the principal component analysis (PCA) subspace. Based on this insight, we propose PCA-guided effective search for *K*-means. Because the PCA subspace is much smaller than the original space, searching in the PCA subspace is both more effective and efficient. Extensive experiments on four real world data sets and systematic comparison with previous algorithms demonstrate that our proposed method outperforms the rest as it makes the *K*-means more effective.

1. Introduction

Data clustering (or just clustering), also known as unsupervised classification, is a method of creating groups of objects, or clusters, in such a way that all objects within a single cluster are very similar while objects in different clusters are quite distinct [1]. This approach is widely used for several different types of applications, including exploratory pattern analysis, data mining, decision-making, machine-learning, vector quantization, and compression situations [2].

The *K*-means algorithm is one of the most widely used clustering algorithms, and is simple and easy to understand. It can be efficiently implemented on parallel and distributed computers to solve large scale practical problems. *K*-means clustering models ball-shaped clusters using simple squared distances as distortion [3]. However, the Lloyd algorithm for *K*-means clustering (which is also an EM-style algorithm) converges quickly to a local solution from an initial starting set of cluster centroids. Experimentally, the *K*-means algorithm can be easily trapped in a local minima near the initialization. This is because the formulation of *K*-means is non-convex and there are exponentially many local solutions in high dimensional data.

Generally, there are two directions that can be taken to achieve better solutions. The first direction is to devise methods that can allow the system to escape the local minima. This is the well-known global optimization problem with regard to non-convex problems. Recent advances include simulated annealing, genetic algorithms, and swarm intelligence. These are known as metaheuristics. To the best knowledge of the authors, no computationally practical algorithm that guarantees global minima has been developed yet. Aloise et al. [4] provide some examples of recent work. The second direction is to design better initialization algorithms. Historically, this direction has been studied by numerous authors [5,6]. We give a detailed review in Section 5 with details of six approaches that we compare in a number of experiments.

In previous research, searching for better initialization has been performed in the original high dimensional space (in this paper, we call it full space). Recent work [7,8] provides a theoretical analysis of the close relationship between *K*-means clustering and principal component analysis (PCA). This work reveals that the global solution to *K*-means clustering lies in the PCA subspace. These new theoretical conclusions open up an entirely new research direction.

There are many possible ways to explore this new research direction. In this paper, we propose one method that follows this new direction: We perform more effective search in the PCA subspace and explore more solution spaces by randomly searching centroid space. We call this method PCA-guided search for *K*-means clustering. We utilize the fact that *K*-means clustering is more efficient in low-dimensional space because its computational complexity is dominated by the *M*-step, where the distance between each data point to all K centroids is computed: $O(nKp)$, where n, K, p are, respectively, the number of data points, the number of clusters, and the dimension of the data. Thus our approach utilizes this advantage because we work in the PCA subspace with a small dimensionality.

☆ This paper has been recommended for acceptance by S. Todorovic.

* Corresponding author. Tel.: +086 13083062520; fax: +010 0551 63861131.
E-mail addresses: xuqin2013@aliyun.com (Q. Xu), chqding@cse.uta.edu (C. Ding), liujinpei2012@163.com (J. Liu), luobinahu@gmail.com (B. Luo).

http://dx.doi.org/10.1016/j.patrec.2014.11.017
0167-8655/

Searching within a greater solution space is achieved by repeated randomly initialized K-means clustering. We perform extensive experiments on several data sets and compare the results with a number of existing algorithms. Experimental results demonstrate that our method can lead to good results with a much lower CPU time than with the standard K-means method.

The rest of this paper is organized as follows: Section 2 refines the theoretical analysis that PCA provides continuous solutions to the discrete cluster membership indicators for K-means clustering. Section 3 describes the PCA-guided search for the K-means algorithm, and Section 4 presents a detailed discussion concerning random initialization method. Section 5 reviews some representative methods including KR, KKZ, HAC, K-means++ and PCA-part algorithms that we compare in a number of experiments. Section 6 shows experiments on four real-world data sets. Section 7 concludes the paper.

2. Solution of K-means clustering lies in PCA-subspace

Recent research discoveries have indicated that there is a close relationship between K-means clustering and PCA [7,8]. More precisely, the solution to K-means clustering is specified by cluster indicators (as refined in Theorem 1). Moreover, the continuous solution of the cluster indicators is given by the principal components of PCA in the form of Theorem 2. Therefore, the global solution to K-means clustering lies in the PCA subspace. Based on this, performing clustering in the PCA subspace is highly desirable because the solutions are close to the global solution.

2.1. PCA provides continuous solution to K-means clustering

For the PCA of input data $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n) \in \Re^{p\times n}$, where p denotes the dimension of data, we compute the principal eigenvectors $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_k)$ of the covariance matrix $\mathrm{cov}(\mathbf{X}) = (1/n)\sum_i (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T$ with principal eigenvalues $(\lambda_1, \ldots, \lambda_k)$. The new coordinates in the PCA subspace are given by $\mathbf{Y} = (\mathbf{y}_1, \ldots, \mathbf{y}_n)$, $\mathbf{y}_i = \mathbf{U}^T\mathbf{x}_i$. The *whitened* coordinates are defined as

$$\tilde{\mathbf{y}}_i = \left(\frac{\mathbf{u}_1}{\sigma_1}, \ldots, \frac{\mathbf{u}_k}{\sigma_k}\right)^T \mathbf{x}_i = (\mathbf{U}\boldsymbol{\Sigma}^{-1})^T\mathbf{x}_i \tag{1}$$

where $\sigma_k = \sqrt{\lambda_k}$, $\boldsymbol{\Sigma} = \mathrm{diag}(\sigma_1, \ldots, \sigma_k)$. The difference between $\mathbf{Y}$ and whitened $\tilde{\mathbf{Y}} = (\tilde{\mathbf{y}}_1, \ldots, \tilde{\mathbf{y}}_n)$ is that the covariance of $\mathbf{Y}$ is $\mathrm{cov}(\mathbf{Y}) = \boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_k)$, while the covariance of $\tilde{\mathbf{Y}}$ is $\mathrm{cov}(\tilde{\mathbf{Y}}) = \mathbf{I}$. Recent works [7,8] show a direct relationship between PCA and K-means clustering. The K-means clustering minimizes the objective function

$$J_{K\text{-means}} = \sum_{j=1}^{K}\sum_{\mathbf{x}_i\in C_j} \|\mathbf{x}_i - \boldsymbol{\mu}_j\|^2. \tag{2}$$

where $\boldsymbol{\mu}_j$ is the centroid of cluster C_j. Throughout this paper, $\|\cdot\|$ denotes the Euclidean norm. For the sake of consistency with PCA, we use the Euclidean distance given in Eq. (2). Relevant background research is provided in Ref. [9].

The solution of K-means clustering is specified by the cluster membership indicator matrix $\mathbf{H} = (\mathbf{h}_1, \ldots, \mathbf{h}_\ell, \ldots, \mathbf{h}_K)$ where

$$\mathbf{h}_\ell = (0\cdots 0, \overbrace{1\cdots 1}^{n_l}, 0\cdots 0)^T/n_\ell^{1/2} \tag{3}$$

i.e., $\mathbf{H}_{i\ell} = 1/n_\ell^{1/2}$ if x_i is clustered into cluster ℓ; $\mathbf{H}_{i\ell} = 0$, otherwise. The final outcome of K-means clustering is to obtain $\mathbf{H}$. The discrete solution of $\mathbf{H}$ cannot be analyzed. But the continuous relaxed solution of $\mathbf{H}$ (i.e. the indicator matrix elements relax from discrete values into continuous values in $(-1, 1)$) can be analyzed.

Theorem 1. *The optimal solution of the relaxed cluster indicators* $\mathbf{H}$ *for* K*-means clustering on data* $\mathbf{X}$ *is given by the whitened projection coordinates* $\tilde{\mathbf{Y}}^T\mathbf{R}^T$*, where* $\mathbf{R}$ *is a* K*-by-*K *unknown rotation matrix satisfying* $\mathbf{R}\mathbf{R}^T = \mathbf{R}^T\mathbf{R} = \mathbf{I}$.

We note that $\mathbf{R}$ does not change the results of K-means clustering of Eq. (2). In all the following analysis, $\mathbf{R}$ always cancels out exactly.

Proof. Using the cluster indicators of Eq. (3), the cluster centroid can be expressed as

$$\boldsymbol{\mu}_j = \frac{1}{n_j}\sum_{\mathbf{x}_i\in C_j}\mathbf{x}_i = \frac{1}{n_j^{1/2}}\mathbf{X}\mathbf{h}_j. \tag{4}$$

As Eq. (2) can be written by

$$J_{k\text{-means}} = \sum_{j=1}^{K}\sum_{\mathbf{x}_i\in C_j}(\|\mathbf{x}_i\|^2 - 2\boldsymbol{\mu}_j^T\mathbf{x}_i + \|\boldsymbol{\mu}_j\|^2) \tag{5}$$

$$= \sum_{i=1}^{n}\|\mathbf{x}_i\|^2 - \sum_{j=1}^{K}\boldsymbol{\mu}_j^T\sum_{\mathbf{x}_i\in C_j}(2\mathbf{x}_i - \boldsymbol{\mu}_j). \tag{6}$$

Substituting Eq. (4) into Eq. (6) we obtain

$$J_{k\text{-means}} = \sum_{i=1}^{n}\|\mathbf{x}_i\|^2 - \sum_{j=1}^{K} n_j\boldsymbol{\mu}_j^T\boldsymbol{\mu}_j \tag{7}$$

$$= \sum_{i=1}^{n}\|\mathbf{x}_i\|^2 - \sum_{j=1}^{K}\mathbf{h}_j^T\mathbf{X}^T\mathbf{X}\mathbf{h}_j. \tag{8}$$

The first term is a constant which is independent of the clustering solution. Thus minimizing $J_{k\text{-means}}$ becomes maximizing

$$\sum_{j=1}^{K}\mathbf{h}_j^T\mathbf{X}^T\mathbf{X}\mathbf{h}_j = \mathrm{Tr}\,\mathbf{H}^T\mathbf{X}^T\mathbf{X}\mathbf{H} = \mathrm{Tr}\,\tilde{\mathbf{H}}^T\mathbf{X}^T\mathbf{X}\tilde{\mathbf{H}}, \tag{9}$$

where $\tilde{\mathbf{H}} = \mathbf{H}\mathbf{R}$ and $\mathbf{R}$ is a rotation matrix satisfying $\mathbf{R}\mathbf{R}^T = \mathbf{I}$. Any solution of K-means clustering can be specified by the cluster membership indicator matrix $\mathbf{H}$. It can be easily verified that $\mathbf{H}$ and $\tilde{\mathbf{H}}$ satisfy $\mathbf{I} = \mathbf{H}^T\mathbf{H} = \tilde{\mathbf{H}}^T\tilde{\mathbf{H}}$.

Thus the K-means clustering becomes the optimization

$$\max_{\tilde{\mathbf{H}}}\ \mathrm{Tr}\,\tilde{\mathbf{H}}^T\mathbf{X}^T\mathbf{X}\tilde{\mathbf{H}},\ \text{s.t.}\ \tilde{\mathbf{H}}^T\tilde{\mathbf{H}} = \mathbf{I} \tag{10}$$

Note that although $\mathbf{H}$ contains only $\{0, 1\}$ (with proper column normalization as in Eq. (3)), the rotation matrix $\mathbf{R}$ contains negative elements. Thus $\tilde{\mathbf{H}}$ may contain negative elements. The above optimization is a discrete optimization and is a well-known standard NP-hard problem.

The spectral relaxation (approximation) of the above discrete optimization can be obtained by relaxing elements of $\mathbf{H}$ to be continuous real numbers in the range $(0, 1)$[7]. With this relaxation of $\mathbf{H}$, the maximization of Eq. (10) has the optimal solution for $\tilde{\mathbf{H}}$ given by the K eigenvectors $\mathbf{V} = (\mathbf{v}_1, \ldots, \mathbf{v}_K)$ of the Gram matrix $\mathbf{X}^T\mathbf{X}$ associated with the K largest eigenvalues $(\xi_1, \ldots, \xi_K)$, i.e., the optimal solution is given by

$$\tilde{\mathbf{H}}^* = \mathbf{V},\quad \text{or}\quad \mathbf{H}^* = \mathbf{V}\mathbf{R}^T. \tag{11}$$

Now, through the singular value decomposition

$$\mathbf{X} = \sum_{i=1}^{r}\sigma_i\mathbf{u}_i\mathbf{v}_i^T = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T, \tag{12}$$

where r is the rank of $\mathbf{X}$. We can thus verify that

$$\tilde{\mathbf{Y}} = (\mathbf{U}\boldsymbol{\Sigma}^{-1})^T\mathbf{X} = \mathbf{V}^T = \tilde{\mathbf{H}}^{*T} = \mathbf{R}^T\mathbf{H}^{*T}$$

and $\sigma_k = \sqrt{\lambda_k} = \sqrt{\xi_k}$. This completes the proof. One consequence of this theorem is that the optimal solution of K-means clustering lies in the PCA-subspace, as explained below. □

2.2. Global K-means solution lies in PCA-subspace

If we assume that the optimal solution for K-means clustering is obtained, the optimal solution is specified by the obtained optimal cluster centroids $(\mathbf{c}_1, \ldots, \mathbf{c}_K)$. These cluster centroids span a subspace, which we call *cluster centroid subspace*. By definition, the global optimal solution of K-means lies in this centroid subspace. On the other hand, the PCA-subspace is spanned by the principal directions of PCA $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_k)$. We then have the following results:

Theorem 2. *Cluster centroid subspace is identical to PCA-subspace.*

Proof. Let $\mathbf{C} = (\mathbf{c}_1, \ldots, \mathbf{c}_K)$. Because cluster centroids are in general not mutually orthogonal, we construct an orthonormal basis as $\widetilde{\mathbf{C}} = \mathbf{C}(\mathbf{C}^T\mathbf{C})^{-1/2}$. Clearly, $\widetilde{\mathbf{C}}^T\widetilde{\mathbf{C}} = I$. Therefore the projection operator is $\widetilde{\mathbf{C}}\widetilde{\mathbf{C}}^T$: for any data point $\mathbf{x}_i$, $\widetilde{\mathbf{C}}\widetilde{\mathbf{C}}^T$ projects it into the cluster centroid space, $\widetilde{\mathbf{C}}\widetilde{\mathbf{C}}^T\mathbf{x}_i = \widetilde{\mathbf{C}}(\widetilde{\mathbf{C}}^T\mathbf{x}_i)$. Now, we prove

$$\widetilde{\mathbf{C}}\widetilde{\mathbf{C}}^T = \mathbf{U}\mathbf{U}^T. \tag{13}$$

We first note that in a similar manner to Eq. (4), we have $\mathbf{c}_j = n_j^{-1/2}\mathbf{X}\mathbf{h}_j$. Thus

$$\mathbf{C} = (\mathbf{c}_1, \ldots, \mathbf{c}_K) = \mathbf{X}\left(\frac{\mathbf{h}_1}{n_1^{1/2}}, \ldots, \frac{\mathbf{h}_K}{n_k^{1/2}}\right) = \mathbf{X}\mathbf{H}\mathbf{N}^{-1/2},$$

where $\mathbf{N} = \mathrm{diag}(n_1, \ldots, n_K)$. Thus we have

$$\begin{aligned}\widetilde{\mathbf{C}}\widetilde{\mathbf{C}}^T &= \mathbf{C}(\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T \\ &= \mathbf{X}\mathbf{H}\mathbf{N}^{-1/2}(\mathbf{N}^{-1/2}\mathbf{H}^T\mathbf{X}^T\mathbf{X}\mathbf{H}\mathbf{N}^{-1/2})^{-1}(\mathbf{X}\mathbf{H}\mathbf{N}^{-1/2})^T \\ &= \mathbf{X}\mathbf{H}(\mathbf{H}^T\mathbf{X}^T\mathbf{X}\mathbf{H})^{-1}(\mathbf{X}\mathbf{H})^T \\ &= \mathbf{X}\tilde{\mathbf{H}}(\tilde{\mathbf{H}}^T\mathbf{X}^T\mathbf{X}\tilde{\mathbf{H}})^{-1}(\mathbf{X}\tilde{\mathbf{H}})^T. \end{aligned} \tag{14}$$

According to Theorem 1, the solution of K-means clustering is $(\mathbf{h}_1, \ldots, \mathbf{h}_K)\mathbf{R} = \mathbf{V}$ or $\tilde{\mathbf{H}} = \mathbf{V}$. Thus we have

$$\widetilde{\mathbf{C}}\widetilde{\mathbf{C}}^T = \mathbf{X}\boldsymbol{V}(\boldsymbol{V}^T\mathbf{X}^T\mathbf{X}\boldsymbol{V})^{-1}(\mathbf{X}\boldsymbol{V})^T. \tag{15}$$

From Eq. (12), by right-multiplying $\boldsymbol{v}_i$, we obtain a crucial property of SVD: $\mathbf{X}\boldsymbol{v}_i = \sigma_k\mathbf{u}_i$, i.e., $\mathbf{X}\boldsymbol{V} = \mathbf{U}\boldsymbol{\Sigma}$. Substituting this into Eq. (13) we arrive at

$$\widetilde{\mathbf{C}}\widetilde{\mathbf{C}}^T = \mathbf{U}\boldsymbol{\Sigma}(\boldsymbol{\Sigma}^T\mathbf{U}^T\mathbf{U}\boldsymbol{\Sigma})^{-1}(\mathbf{U}\boldsymbol{\Sigma})^T = \mathbf{U}\mathbf{U}^T \tag{16}$$

This completes the proof. □

Theorems 1 and 2 explain that the PCA dimensionality reduction is beneficial for K-means clustering: They assert that the optimal solution lies in k-dimensional PCA subspace (within the accuracy of spectral relaxation). The key point is that the k-dimensional PCA subspace is much smaller than the original p-dimensional data space (*full* space). In other words, we search for the optimal solution in a much *narrower* space because we know the correct solution lies within this narrower space. This is more effective than searching for the optimal solution in the much larger original p-dimensional data space. Although we cannot guarantee finding the true global solution, we usually obtain *better* solutions (measured in the objective function value), in comparison to the solutions obtained by searching within the full data space.

3. PCA-guided search for K-means

The algorithm for K-means clustering with PCA-guided search is as follows:

(1) Perform standard K-means clustering in the PCA subspace.
(2) Use the cluster membership obtained in Step 1 to construct initial cluster centroids in the full space.[1] Perform standard K-means clustering in the full space.

[1] These centroids are usually not local optimal solutions in full space, although they are local optimal solutions in PCA subspace.

There are a number of points that should be observed here. First, Step 2 is used to let the solution obtained in the PCA subspace relax into a local optimal solution in the full space. Thus in this algorithm, the majority of iterations of the K-means algorithm are spent in Step 1 (within the PCA subspace). The number of iterations in Step 2 is usually relatively small, because the initially defined cluster centroids are already close to a local minima.

The second observation is with regard to computation speed. In the PCA subspace, the dimensionality is very low and thus the computation of distances is faster than that in the full space. Because most iterations are performed in Step 1 (in the PCA subspace), therefore the K-means clustering with PCA-guided search algorithm is *faster* than the standard K-means algorithm. This point is evident in all runs on all data sets (see the experiments in Section 6).

The third point to be made is that the obtained solution is with good quality with respect to (w.r.t.) the original p-dimensional data space. This is the motivation behind Step 2. All objective function values measured in this paper are w.r.t. the full space, unless explicitly stated otherwise.

In summary, the PCA-guided search algorithm is both *effective* in obtaining better solutions (because it searches a narrower subspace where the global solution lies) and *fast* (because it runs effectively in a lower-dimensional space).

4. Effective search of centroid space

An important focus of this study is to search the centroid space for K-means algorithms. We provide a discussion here because extensive experiments (see Section 6) show that random search of centroid space is an effective method. The K-means algorithm is iterative, in that it starts from an initial guess of the solution and iteratively improves the solution until a *local* optimal solution is reached. Because the objective functions of K-means clustering are non-convex, there can be many different local optimal solutions. Starting from different initializations, it often converges to different local optimal solutions. Our task is to find a good quality local optimal solution.

We use random search in this paper. The simplest random search method is random initialization [5]. There are two approaches. (R1) randomly partitions data points into K groups [10], and (R2) randomly selects K data points as K class centroids. There are studies [11] which suggest method R1 gives better results than method R2.

Here we provide a theoretical analysis to show that the R2 method is better. First, method R2 is easier to implement. More importantly, from the point of view of searching within the wider centroid space, method R2 searches a larger space than method R1. This is because in method R1, the K class centroids $(\mathbf{c}_1, \ldots, \mathbf{c}_K)$ are much closer to each other. We provide analysis to support this statement below.

The standard partition of R1 is by data points. We first randomly select n_1 data points and compute their mean $\boldsymbol{\mu}_1$. From the remaining data points, we randomly select n_2 points and compute their mean $\boldsymbol{\mu}_2$. This is repeated $K-1$ times. The remaining data points give $\boldsymbol{\mu}_K$.

Suppose data X follow a fixed and spatially finite distribution (the variance is finite). Then $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_K$ are close to each other. This is because for the randomly selected n_1 points from the data distribution, the sample mean $\boldsymbol{\mu}_1$ is generally very close to the population mean $\boldsymbol{\mu}$ (computed from the entire population, i.e. the entire data). As n_1 increases, $\boldsymbol{\mu}_1$ is closer to μ. For the same reason, $\boldsymbol{\mu}_2$ is close to $\boldsymbol{\mu}$, and thus is close to $\boldsymbol{\mu}_1$. Therefore, $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$ are close to each other. In summary, the centroids produced by the R1 method are restricted to a narrow space near the population mean.

Based on the above analysis, R1 is not considered to be a good method. In this paper, our focus is on searching broader centroid space, thus we use method R2 for random search.

In addition to random initialization, there are many algorithms that can potentially be considered for use. We provide a comprehensive discussion in Section 5.

5. Related work on *K*-means initialization

There is a significant body of literature on the initialization for *K*-means [6]. In this section, we review some representative work. Among these methods we will compare six of them in the experiment section.

Random initialization [5,10] is an early approach that has been discussed in more depth previously in Section 4.

Kaufman and Rousseeuw [12] selected the initial centroids one after another. The first centroid is the most centrally located in the set of objects. The remaining centroids are then chosen in a certain way in which each chosen centroid is expected to be far away from the previously selected centroids but still have many data points close to it. In this paper, we call this the KR algorithm.

Katsavounidis et al. [13] proposed that the first centroid should be the one which has the maximum norm. Then for each nonselected data point (candidate point), set the minimal distance between it and the centroid set as the distance between them, then choose the candidate point which has the maximal distance between it and the centroids as the next centroid. Repeat this until K centroids have been chosen. In this paper, we name this the KKZ algorithm after the three authors.

Fayyad et al. [14] proposed a refined algorithm that builds a set of small random sub-samples of the data, then clusters data in each sub-sample by k-means. All centroids of all sub-samples are then clustered together by k-means using the k-centroids of each sub-sample as initial centers. The centers of the final clusters that give the minimum clustering error are then used as the initial centers for clustering the original set of data using the k-means algorithm.

Several studies [15–17] used the results of hierarchical agglomerative clustering (HAC) as the initialization of centroids. HAC is a "bottom up" approach that begins from a large number of small clusters (potentially up to N clusters and each of them may include exactly one object). This is followed by a series of merge operations to reduce the number of clusters to one (i.e. all objects belong to the same cluster). Each merge incorporates the two closest clusters. One essential ingredient of the algorithm is the inter cluster distance $d(k, l)$. In our experiments we adopt the method as described in Ref. [15].

In *K*-means++, Arthur and Vassilvitskii [18] also proposed a method which selects the K centroids greedily. The first centroid is selected randomly from the data set. Then the next point $\mathbf{x}'$ is chosen as the centroid with probability $\frac{d(\mathbf{x}')^2}{\sum_{\mathbf{x}\in\mathbf{X}} d(\mathbf{x})^2}$, where $d(\mathbf{x})$ denotes the shortest distance from the data point $\mathbf{x}$ to the closest centroid. This is repeated until K centroids are identified.

Recursive partitioning of data in a "top-down" fashion has also been used for initialization. Su and Dy [19] proposed to recursively bi-partition a current cluster into two by computing the first principal component and use the sign of each element to split the cluster, similar to the indicator h_1 of Theorem 1. In each iteration, they select the cluster with the maximum distortion to split. This process is repeated until K clusters are obtained. The obtained clusters are used for *K*-means initialization. We call this the PCA-part method. This algorithm is similar to PDDP, proposed by Boley [20]. For more discussion on selecting clusters to split and objective function analysis, see Ref. [21].

6. Experiments and results

To evaluate the performance of the various initialization methods, experiments based on image data clustering are conducted on four real-world data sets. All experiments are implemented in Matlab and executed on a dual-core 3.0 GHz Pentium CPU with 2 GB of memory and no effort made to optimize algorithm speed.

6.1. Data sets description

The four data sets consist of a facial image data set, a handwritten digits image data set, a handwritten alphabet image data set, and a data set of images of objects rotated through 360°.

The AT&T Face Data Set consists of ten different images of 40 distinct people. The size of each image is 92×112 pixels, with 256 grey levels per pixel. In our experiment, we use pixels as features. Each image was resized to be 23×28 and converted into a vector of a dimension 644 by ordering the rows of the matrix one after the other.

The MNIST Handwritten Digits Data Set is composed of 8-bit grayscale images of "0" through "9" with approximately 6000 training examples of each class (digit). We select the first 50 images from each class as our experimental data and convert them into vectors of dimension 784.

The Binary Alphabet Data Set contains 26 handwritten alphabets "A"–"Z" with 39 examples of each class (alphabet). Each image is a 20×16 binary image. We convert them into vectors of dimension 320.

The Coil20 Data Set is a database of 1440 grayscale images of 20 objects (72 images per object). The objects have a wide variety of complex geometric and reflectance characteristics. The database has two sets of images. We use the second, which contains 1440 size normalized images of 20 objects, and select the first 50 images per cluster to constitute the experimental data. Each image's size is 32×32. We convert them into vectors of dimension 1024.

6.2. Results on K-means

We perform *K*-means clustering using the various methods discussed previously to initialize centroids. As the goal of *K*-means is to minimize the within-class sum of square errors (WCSS), we compute the cost function value (distortion) of Eq. (2) as the comparison criterion. In the PCA-guided search method, the dimensionality of the PCA subspace is defined as the number of clusters. In this case, as there are 40 distinct people in the AT&T data set, the dimension is reduced to 40. For similar reasons, the MNIST data set is reduced to a dimension of 10, the BinAlpha data set to dimensionality of 26, and the Coil20 data set is reduced to a dimensionality of 20. The methods that we compare in this paper are: random search, *K*-means++, PCA-guided search, KKZ, KR, HAC, and PCA-part.

The results of the *K*-means algorithm are given in Figs. 1–4. Here, for the three probabilistic methods (random-search, *K*-means++ and PCA-guided-search), we perform 1000 trials of each algorithm. For the

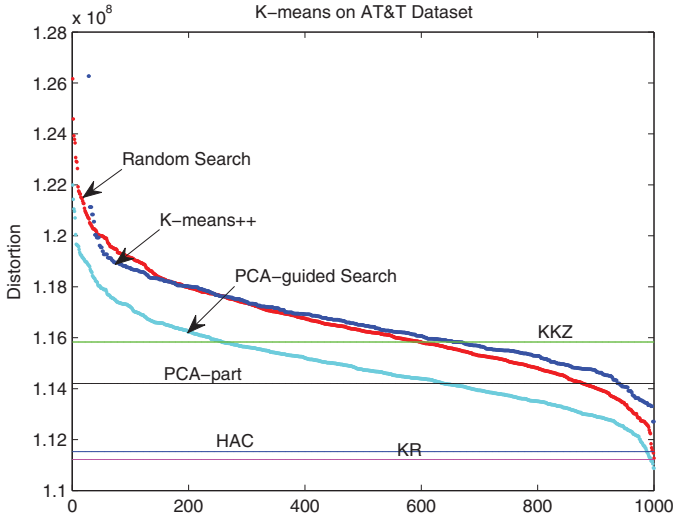

Fig. 1. Results for *K*-means on the AT&T data set.

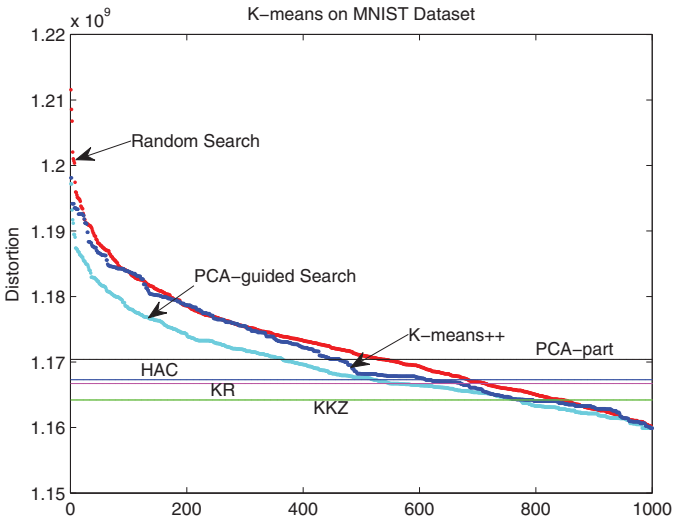

Fig. 2. Results for *K*-means on the MNIST data set.

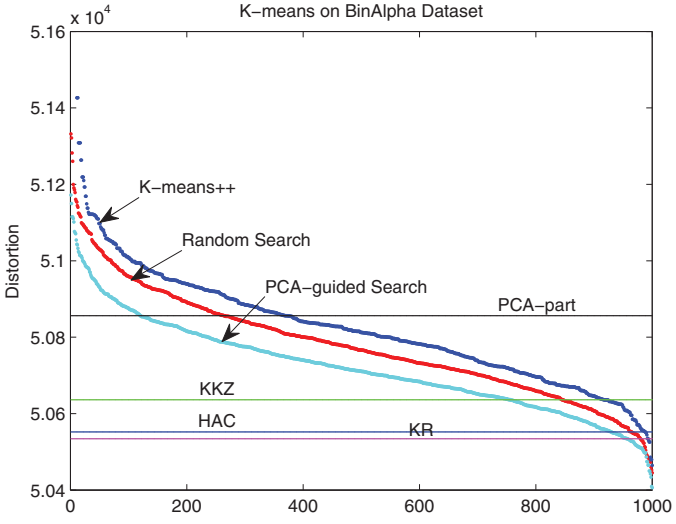

Fig. 3. Results for *K*-means on the BinAlpha data set.

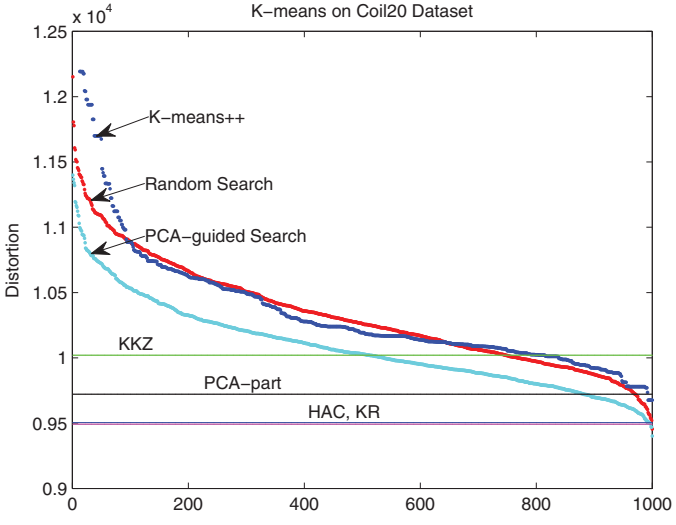

Fig. 4. Results for *K*-means on the Coil20 data set.

Table 1
Results for *K*-means on the AT&T Face Dataset.

Method	Time (s)	Runs	Lowest distortion
Random search	1672.46	1000	1.1161e+08
K-means++	2590.76	1000	1.1271e+08
PCA-guided search	1051.44	1000	**1.1086e + 08**
KKZ	2.24	1	1.1583e+08
KR	373.75	1	1.1122e+08
HAC	532.21	1	1.1153e+08
PCA-part	2.12	1	1.1420e+08

Table 2
Results for *K*-means on the MNIST Dataset.

Method	Time (s)	Runs	Lowest distortion
Random search	2196.35	1000	**1.1599e + 09**
K-means++	2700.09	1000	**1.1599e + 09**
PCA-guided search	1003.62	1000	**1.1599e + 09**
KKZ	3.14	1	1.1642e+09
KR	54.15	1	1.1667e+09
HAC	1237.94	1	1.1673e+09
PCA-part	1.23	1	1.1704e+09

Table 3
Results for *K*-means on the Binary Alphabet Dataset.

Method	Time (s)	Runs	Lowest distortion
Random search	6950.98	1000	5.0445e+04
K-means++	7456.65	1000	5.0465e+04
PCA-guided search	4618.33	1000	**5.0406e + 04**
KKZ	7.83	1	5.0636e+04
KR	805.89	1	5.0534e+04
HAC	7932.24	1	5.0552e+04
PCA-part	1.08	1	5.0856e+04

Table 4
Results for *K*-means on the Coil20 Dataset.

Method	Time (s)	Runs	Lowest distortion
Random search	2156.59	1000	9454.7
K-means++	2949.93	1000	9676.3
PCA-guided search	1121.96	1000	**9401.3**
KKZ	3.37	1	10019.9
KR	205.53	1	9492.2
HAC	1388.54	1	9505.2
PCA-part	2.46	1	9721.3

determinate methods (KKZ, KR, HAC and PCA-part) only one run is necessary. For the convenience of comparison, the results of random-search, *K*-means++ and PCA-guided-search are plotted in descending order of distortion values. The obtained distortion of KKZ, KR, HAC and PCA-part are plotted as four lines respectively. The execution time and the lowest distortion of the compared seven methods are shown in Tables 1–4 for the four data sets. Boldface indicates the initialization method produced the minimal *K*-means WCSS for each data set.

From Figs. 1–4 and Tables 1–4, we can see that the obtained distortion results of *K*-means++ initialization could be fairly poor (high). Random search always outperforms *K*-means++ both in obtained distortion values and CPU time. Furthermore, PCA-guided search always outperforms random search and produces the best results (lowest distortion), and at a lower CPU time. The determinate methods (KKZ, KR, HAC and PCA-part) cannot obtain a low distortion that is comparable to the PCA-guided search method. From Fig. 1 and Table 1, it can be seen that the KR method is more effective in comparison to the other determined methods. Among the 1000 trials of the PCA-guided search methods, there are 4 trials providing better results than the KR method. In other words, KR is equivalent to $1000/4 = 250$ trials of PCA-guided search, which takes $1051.44/1000 \times 250 = 262.8$ s (see Table 1). However, the KR method takes 373.75 s. The ratio of the

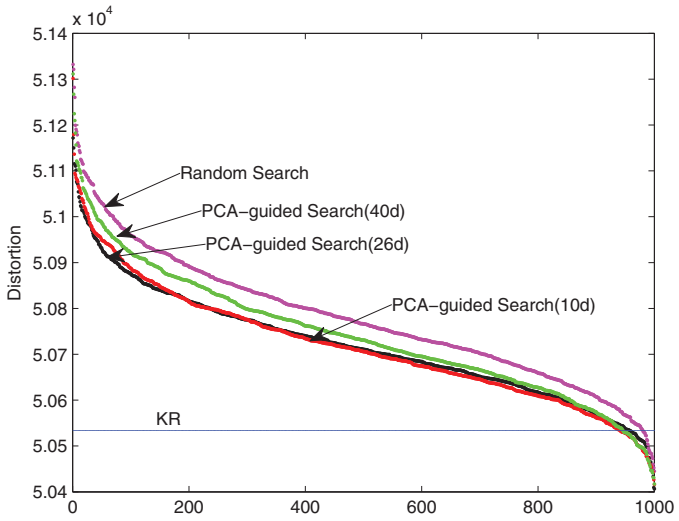

Fig. 5. Variation of *K*-means results with PCA subspace dimension on the BinAlpha data set.

PCA-guided search approach to the KR method is 373.75/262.8 = 1.42. As this ratio is greater than 1, this shows that the PCA-guided method is more efficient than the KR method. For the same reason, the KR method is not more efficient than the PCA-guided method for the other 3 data sets. Moreover, the efficiency of the KR method depends on the number of clusters. When the number of clusters increases, the time complexity also increases dramatically.

It should be noted that the PCA-subspace computation here is performed only once. The CPU times for these are: 0.215 s for the AT&T data set, 0.262 s for the MNIST data set, 0.287 s for the BinAlpha data set, and 0.443 s for the Coil20 data set. This is a small overhead in comparison to the PCA-guided search time listed in Tables 1–4.

To see the effect of varying the PCA subspace dimension, we reduce the BinAlpha data set to 10-d, 26-d and 40-d dimensions respectively and plot the results of the PCA-guided search method, as shown in Fig. 5. For comparison, we also plot the results of the KR and random search methods, which always outperform the other methods. From Fig. 5, it can be seen that the results of 10-d, 26-d and 40-d are very similar. Thus the PCA-guided search method is not sensitive to the dimension of the PCA subspace. Therefore, in general, the dimension of the data set is reduced to the number of clusters for implementation.

In summary, for *K*-means clustering, if one can afford to use additional CPU time to search for a better solution, the PCA-guided search method presented in this paper has been clearly identified to be the optimal choice.

7. Summary

In this paper we seek to identify a more effective initialization for *K*-means clustering, known as the PCA-guided search method. This is based on the theoretical analysis that the global solution to *K*-means lies in the PCA-subspace. Through extensive experiments and comparisons to six other methods, it was confirmed that PCA-guided search produces the best results for all the data sets. To expand on these positive results, it should be noted that as the data size increases, an implementation of an online *K*-means algorithm becomes increasingly important for application to real world problems. For *K*-means clustering, this is easy to implement by keeping the *K* centroids in memory and updating them as each new data instances are received and projected into the PCA subspace. The subspace can occasionally be updated using an efficient algorithm [22]. Other additional future work involves utilizing more advanced subspace methods for *K*-means clustering.

Acknowledgments

We thank the anonymous referees for their constructive comments which have helped improve the paper. We would also like to thank Andrew Abel of the University of Stirling for his suggestions and revisions to improve the language of the paper. The research of Q. Xu and B. Luo is supported by the National Natural Science Foundation of China (no. 61073116, no. 61211130309), and the "211 Project" of Anhui University. The research of C. Ding is partially supported by US NSF CCF-0917274 and NSF DMS-0915228. The research of J. P. Liu is supported by Humanity and Social Science Youth foundation of Ministry of Education (13YJC630092), and Humanities and Social Science Research Project of Department of Education of Anhui Province (no. SK2013B041), and Anhui Provincial Natural Science Foundation.

References

[1] G. Gan, C. Ma, J. Wu, Data Clustering: Theory, Algorithms, and Applications, ASA-SIAM Series on Statistics and Applied Probability, SIAM, Society for Industrial and Applied Mathematics, 2007.
[2] P. Berkhin, Survey of Clustering Data Mining Techniques, third ed., Macmillan, New York, NY, 2007.
[3] A.K. Jain, Data clustering: 50 years beyond k-means, Pattern Recognit. Lett. 31 (2010) 651–666.
[4] D. Aloise, P. Hansen, L. LibertiAn, Improved column generation algorithm for minimum sum-of-squares clustering, Math. Program. 131 (2012) 195–220.
[5] E. Forgy, Cluster analysis of multivariate data: efficiency vs. interpretability of classifications, Biometrics 21 (1965) 768–769.
[6] M.E. Celebi, H. Kingravi, P.A. Vela, A comparative study of efficient initialization methods for the k-means clustering algorithm, Expert Syst. Appl. 40 (2013) 200–210.
[7] H. Zha, X. He, C. Ding, H. Simon, M. Gu, Spectral relaxation for k-means clustering, in: Advances in Neural Information Processing Systems, 2001, pp. 200–210.
[8] C. Ding, X. He, K-means clustering via principal component analysis, in: Proceedings of 21st International Conference on Machine Learning, 2004, pp. 225–232.
[9] C. Ding, D. Zhou, X. He, H. Zha, R1-pca: rotational invariant l1-norm principal component analysis for robust subspace factorization, in: Proceedings of the 23rd International Conference on Machine Learning, 2006, pp. 281–288.
[10] M.R. Anderberg, Cluster Analysis for Applications, New York, NY, 1973.
[11] J.M. Peña, J.A. Lozano, P. Larrañaga, An empirical comparison of four initialization methods for the k-means algorithm, Pattern Recognit. Lett. 20 (1999) 1027–1040.
[12] L. Kaufman, P.J. Rousseeuw, Finding Groups in Data. An Introduction to Cluster Analysis, Wiley, New York, 1990.
[13] I. Katsavounidis, C.C. Jay Kuo, Z. Zhang, A new initialization technique for generalized lloyd iteration, IEEE Singal Process. Lett. 1 (1994) 144–146.
[14] U.M. Fayyad, C.A. Reina, P.S. Bradley, Initialization of iterative refinement clustering algorithms, in: Proceedings of 3rd International Conference on Knowledge Discovery and Data Mining, 1998, pp. 194–198.
[15] C. Fraley, Algorithms for model-based gaussian hierarchical clustering, SIAM J. Sci. Comput. 20 (1998) 270–281.
[16] M. Meilă, D. Heckerman, An experimental comparison of several clustering and initialization methods, in: Proceedings of 4th Conference on Uncertainty in Artificial Intelligence, 1998, pp. 386–395.
[17] S.J. Redmond, C. Heneghan, A method for initialising the k-means clustering algorithm using kd-trees, Pattern Recognit. Lett. 29 (2007) 965–973.
[18] D. Arthur, S. Vassilvitskii, k-means++: the advantages of careful seeding, in: SODA'07: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035.
[19] T. Su, J.G. Dy, In search of deterministic methods for initializing k-means and gaussian mixture clustering, Intell. Data Anal. 11 (2007) 319–338.
[20] D. Boley, Principal direction divisive partitioning, Data Mining Knowl. Discov. 2 (1998) 325–344.
[21] C. Ding, X. He, Cluster merge and split in hierarchical clustering, in: Proceedings of 2nd IEEE ICDM, 2002, pp. 139–146.
[22] H. Zha, H.D. Simon, On updating problems in latent semantic indexing, SIAM J. Sci. Comput. 21 (1999) 782–791.