A Single-Pass Algorithm for Efficiently Recovering Sparse Cluster Centers of High-dimensional Data

Jinfeng Yi JINFENGY@US.IBM.COM
IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598, USA

Lijun Zhang ZHANGLJ@LAMDA.NJU.EDU.CN
National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China

Jun Wang WANGJUN@US.IBM.COM
IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598, USA

Rong Jin RONGJIN@CSE.MSU.EDU
Anil K. Jain JAIN@CSE.MSU.EDU
Department of Computer Science and Engineering, Michigan State University, East Lansing, MI 48824 USA

Abstract

Learning a statistical model for high-dimensional data is an important topic in machine learning. Although this problem has been well studied in the supervised setting, little is known about its unsupervised counterpart. In this work, we focus on the problem of clustering high-dimensional data with sparse centers. In particular, we address the following open question in unsupervised learning: "is it possible to reliably cluster high-dimensional data when the number of samples is smaller than the data dimensionality?" We develop an efficient clustering algorithm that is able to estimate sparse cluster centers with a single pass over the data. Our theoretical analysis shows that the proposed algorithm is able to accurately recover cluster centers with only $O(s \log d)$ number of samples (data points), provided all the cluster centers are s-sparse vectors in a d dimensional space. Experimental results verify both the effectiveness and efficiency of the proposed clustering algorithm compared to the state-of-the-art algorithms on several benchmark datasets.

Proceedings of the 31^{st} International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32.

1. Introduction

Data clustering, also known as unsupervised learning, is an important task in machine learning. Since clustering is closely related to density estimation, analyzing the number of samples required for accurately recovering the underlying distributions, referred to the problem of *sample complexity*, is an important but challenging open problem (Srebro, 2007).

Although numerous algorithms have been developed for data clustering, only a few of them address the challenge of clustering high-dimensional data. It is well known that the number of samples needed for accurate density estimation is at least exponential in the dimensionality (Tsybakov, 2008). Even in the case when data points are sampled from a finite mixture of Gaussian distributions, the sample complexity is still polynomial in dimensionality (Dasgupta, 1999; Arora & Kannan, 2001; Vempala & Wang, 2004; Kannan et al., 2005; Srebro et al., 2006; Chaudhuri et al., 2009). Given these theoretical results, we aim to examine the following question in this study:

Is it possible to achieve accurate clustering results when the data dimensionality is larger than the number of samples to be clustered?

In (Azizyan et al., 2013), the authors studied a special case of this problem where data points were sampled from a mixture of two isotropic Gaussians. The authors showed that when the cluster centers are d dimensional s-sparse vectors (i.e. there are no more than s non-zero entries), the sample complexity can be reduced to $O(s^2 \log d)$. In this work, we examine this question in a more general setting

where there are K clusters, $K > 2$. We show that when data points are sampled from a mixture of $K \geq 2$ spherical Gaussians with s-sparse centers, only $O(s \log d)$ samples are needed to reliably estimate the cluster centers. Our results indicate that it is indeed possible to reliably cluster high-dimensional data even when the number of samples is smaller than the dimensionality as long as the cluster centers are sparse.

To support our theoretical claim, we present an efficient clustering algorithm that only needs to go through all the data points once to obtain an accurate estimation of cluster centers. This is in contrast to many clustering algorithms, such as k-means (Lloyd, 1982) and mixture models (Lindsey, 1996), which require going through the data set multiple times before the final centers can be determined. To take advantage of sparse cluster centers, similar to sparse learning, an ℓ_1 regularizer is introduced to obtain a better estimates of cluster centers. Our empirical study with multiple high-dimensional data sets shows that the proposed algorithm, despite its simplicity, yields better results compared to state-of-the-art clustering algorithms for high-dimensional data.

We finally comment on the applications of the proposed clustering technique. Although most clustering studies assume a sufficiently large number of data points, there are many scenarios where the number of data points to be clustered is significantly smaller than the dimensionality, and at the same time, the cluster centers are likely to be sparse. One example is document clustering, where the dimensionality, i.e., the number of distinct terms, can be significantly larger than the number of documents to be clustered (Ertöz et al., 2003; Cormack & Lynam, 2005; Keerthi & DeCoste, 2005). At the same time, the term frequencies usually follow a power-law distribution, thus many terms in a given cluster will only occur in one or two documents, giving them very low weight in the cluster center (Sculley, 2010). Another example is the clustering of genes based on their sequence information (De Smet et al., 2002; Wang & Yang, 2005). In this application, each gene is represented by a histogram vector of motifs, where the number of unique motifs derived from the data can be sometimes larger than the number of genes to be clustered and usually only a small number of motifs are biologically functional.

Besides the application to clustering high-dimensional data, the theoretical results presented in this work have profound impact on the practice of data clustering. It implies that when all the cluster centers are s-sparse, it is possible to only utilize $O(s \log d)$ data points for accurately estimating the cluster centers, suggesting a simple approach for efficiently clustering billions of data points (i.e. estimate the cluster centers using the randomly sampled $O(s \log d)$ data points and then apply the estimated center to find the appropriate cluster members for all the data points).

The remainder of the paper is organized as follows. Section 2 reviews the related work on high-dimensional clustering, Gaussian mixture model and theoretical studies of the k-means algorithm. In Section 3, we introduce the proposed framework for clustering high-dimensional data. Theoretical analysis of the proposed algorithm is presented in Section 4. We summarize the results of our empirical studies in Section 5. Section 6 concludes with the future work.

2. Related work

In this section, we review the existing work on clustering high-dimensional data, as well as theoretical analysis for Gaussian mixture models and k-means algorithm, the two of the most popular clustering algorithms.

Clustering high-dimensional data One common approach to high-dimensional clustering is to first perform dimensionality reduction using algorithms such as Principal Component Analysis (PCA) and kernel PCA (Mika et al., 1998) as a pre-processing step, followed by a standard clustering method in the lower dimensional space. However, Johnstone (2007) showed that when $n < d$, PCA fails because it is unable to distinguish signal from noise. Another limitation of these dimensionality reduction approaches is that they assume the data points from different clusters share the same subspace, which may not hold in many real-world applications. Subspace clustering (Agrawal et al., 1998) addresses this limitation by trying to identify a low-dimensional subspace for each cluster that captures most of the data variance in the cluster. The main shortcoming of subspace clustering is that it has to make a strong assumption about data and is usually computationally expensive. Since the number of candidate subspaces is exponential in the dimensionality, a naive implementation of subspace clustering algorithm will be computationally infeasible for high dimensional data. Several clustering algorithms have been proposed to improve the computational efficiency by exploring the property of sparsity. Pan & Shen (2007) applied penalized mixture models to conduct variable selection and clustering simultaneously. Witten & Tibshirani (2010) developed a lasso-type penalty to perform feature selection in both k-means and hierarchical clustering. Sun et al. (2012) explored adaptive group lasso for data clustering. However, none of these approaches provide a theoretical guarantee on how the sparsity assumption can be used to enhance clustering performance.

Gaussian mixture model Assuming that data points are sampled from a mixture of Gaussian distributions, a Gaussian mixture model (GMM) can be used for clustering. The most popular approach for learning a GMM is the EM algo-

rithm (Figueiredo & Jain, 2002), which is not guaranteed to find the global optimal. Over the past decade, many studies have examined the learnability of GMM. Among them, pairwise methods (Dasgupta, 1999; Dasgupta & Schulman, 2000; Arora & Kannan, 2001) were first proposed to estimate the mixture distribution. However, these methods require a larger distance between centers with increasing dimensionality. To address this problem, spectral methods (Vempala & Wang, 2004; Achlioptas & McSherry, 2005; Kannan et al., 2005; Brubaker & Vempala, 2008; Hsu & Kakade, 2013) were introduced to estimate a mixture of Gaussians with a mean separation that is independent of data dimensionality. In (Belkin & Sinha, 2010; Kalai et al., 2012), the authors use the method of moments to estimate the Gaussian mixtures without requiring a large separation between Gaussian components. However, a major limitation of these studies is their high sample complexity, i.e., the number of samples required for accurately recovering the underlying mixture components is at least polynomial in dimensionality. Although this issue was addressed in (Azizyan et al., 2013) under the assumption of sparse cluster centers, the result is restricted to only two clusters, significantly limiting its potential applications.

In addition to Gaussian mixture models, several studies have addressed general mixture models. Dasgupta et al. (2005) studied the question of learning mixtures of distributions with heavy-tails. In (Achlioptas & McSherry, 2005; Kannan et al., 2005), the authors showed that their results can also be applied to mixtures of log-concave distributions. Moreover, Chaudhuri (2007); Blum et al. (2009) examined the sample complexity problem associated with learning mixtures of binary product distributions.

k-means The proposed algorithm is related to some of the theoretical studies on the k-means algorithm. Chaudhuri et al. (2009) presented a modified k-means algorithm that is able to accurately recover the cluster centers for two clusters when the number of samples is at least linear in the dimensionality. Balcan et al. (2009a;b; 2013) showed it is possible to efficiently find a solution that is close to the true data partition if all c-approximations to the global optimal solution of k-means only differ from the true partition on at most ϵ fraction of data points. It is however unclear when the assumption will hold in real-world applications.

3. High-dimensional Clustering as Iteratively Refined Estimation

The proposed clustering algorithm for high-dimensional data needs only one pass over the data points to be clustered to estimate the cluster centers. Let $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ be the set of n data points to be clustered into K clusters, where each $\mathbf{x}_i \in \mathbb{R}^d$ is a vector of d dimensions. To accurately estimate the cluster centers, we develop an iteratively refined estimation procedure: it divides $\mathcal{D}$ into a sequence of disjoint subsets whose sizes increase exponentially over the sequence; the cluster centers computed from the first subset will be used as the initial solution, and then be refined by using the data points from the second subset, and so on. Below, we first present our procedure for iteratively refined estimation of sparse cluster centers, and then discuss its theoretical properties in the next section.

The proposed algorithm is an iterative procedure. Without loss of generality, we assume $n = T(2^m - 1)$ for some integers T and m. The proposed algorithm first randomly divides the collection of n data points into m subsets, denoted by $\mathcal{S}^1, \ldots, \mathcal{S}^m$, with $|\mathcal{S}^i| = T2^{i-1}$. The initial guess for cluster centers is denoted by $\widehat{\mathbf{c}}_1^1, \ldots, \widehat{\mathbf{c}}_K^1$.

Given the initial cluster centers, we iteratively update them. At each iteration t, we use the data points in $\mathcal{S}^t$, and identify, for each data point $\mathbf{x}_i^t \in \mathcal{S}^t$, its closest cluster $\widehat{k}_i^t$ by $\widehat{k}_i^t = \arg\max_{j \in [K]} [\widehat{\mathbf{c}}_j^t]^\top \mathbf{x}_i^t$.

One major difference between the proposed algorithm and the k-means algorithm is that we use a different subset of data points at each iteration, which is the key to ensure that the final cluster center estimates will be close to the optimal cluster centers. While (Chaudhuri et al., 2009) also used different subsets of data at each iteration, they keep the size of subsets unchanged for all the iterations. Consequently, they are only able to deal with two clusters. In contrast, we increase the size of subset by a constant factor $p > 1$, and are able to address the K clusters problem, $K \geq 2$.

Given computed cluster memberships, our next step is to update the cluster centers. To take advantage of the sparsity of cluster centers, we introduce an ℓ_1 regularizer in estimating the new cluster centers. More specifically, given the estimated cluster centers at iteration t, $\{\widehat{\mathbf{c}}_k^t\}_{k=1}^K$, we denote by $\widehat{\mathcal{S}}_k^t$ the subset of data points in $\mathcal{S}^t$ that are assigned to $\widehat{\mathbf{c}}_k^t$. Instead of computing the new center as the average of data points in $\widehat{\mathcal{S}}_k^t$, we estimate the new cluster center $\widehat{\mathbf{c}}_k^{t+1}$ at iteration $t+1$ by solving the following optimization problem

$$\min_{\mathbf{c} \in \mathbb{R}^d} \lambda^t \|\mathbf{c}\|_1 + \frac{1}{|\widehat{\mathcal{S}}_k^t|} \sum_{\mathbf{x}_i^t \in \widehat{\mathcal{S}}_k^t} \|\mathbf{c} - \mathbf{x}_i^t\|^2, \tag{1}$$

where $\lambda^t > 0$ is the regularization parameter at iteration t. The optimal solution of the k-th cluster center in (1) is given by

$$\bar{\mathbf{x}}_k^t - \text{sign}(\bar{\mathbf{x}}_k^t) \min(|\bar{\mathbf{x}}_k^t|, \frac{\lambda^t}{2}), \tag{2}$$

where $\bar{\mathbf{x}}_k^t = \frac{1}{|\widehat{\mathcal{S}}_k^t|} \sum_{\mathbf{x}_i^t \in \widehat{\mathcal{S}}_k^t} \mathbf{x}_i^t$ is the average of data points in S_k^t.

Algorithm 1 shows the detailed steps of this approach. Note

Algorithm 1 Clustering as Iteratively Refined Estimation of *Sparse* Cluster Centers

Input: The number of clusters K, size T, and λ^1

Randomly divide the collection of n data points into $\mathcal{S}^1, \ldots, \mathcal{S}^m$, with $|\mathcal{S}^i| = T2^{i-1}$.
Compute the initial cluster centers $\widehat{\mathbf{c}}_1^1, \ldots, \widehat{\mathbf{c}}_K^1$
for $t = 1, \ldots, m$ **do**
  //Find closest cluster for each data point in $\mathcal{S}^t$
  for $i = 1, \ldots, |\mathcal{S}^t|$ **do**
    Find the closet cluster $\widehat{k}_i^t$ for $\mathbf{x}_i^t \in \mathcal{S}^t$, i.e.,

$$\widehat{k}_i^t = \arg\max_{j \in [K]} [\widehat{\mathbf{c}}_j^t]^\top \mathbf{x}_i^t$$

  end for
  //Update cluster centers using data points in $\mathcal{S}^t$
  for $k = 1, \ldots, K$ **do**
    Set $\widehat{\mathcal{S}}_k^t = \{\mathbf{x}_i^t \in \mathcal{S}^t : \widehat{k}_i^t = k\}$ includes all the data points assigned to cluster $\widehat{\mathbf{c}}_k^t$
    Update the k-th cluster center

$$\widehat{\mathbf{c}}_k^{t+1} = \arg\min_{\mathbf{c} \in \mathbb{R}^d} \lambda^t \|\mathbf{c}\|_1 + \frac{1}{|\widehat{\mathcal{S}}_k^t|} \sum_{\mathbf{x}_i^t \in \widehat{\mathcal{S}}_k^t} \|\mathbf{c} - \mathbf{x}_i^t\|^2$$

  end for
  $\lambda^{t+1} = \lambda^t / \sqrt{2}$.
end for

Return The cluster centers $\widehat{\mathbf{c}}_1^{m+1}, \ldots, \widehat{\mathbf{c}}_K^{m+1}$

that in Algorithm 1, the regularization coefficient λ^t is reduced by a constant factor $\sqrt{2}$ at each iteration. This is similar to other algorithms for sparse recovery (Wright et al., 2009; Hale et al., 2008). We note that although the regularization parameter is reduced over iterations, the estimated cluster centers will be ℓ_1 sparse (instead of ℓ_0 sparsity), as shown in (Jin et al., 2013). This is because at the beginning of the iterations, we would expect the estimated centers to be far away from the true cluster centers, and therefore only the entries with large magnitude are kept. With more iterations, we expect the estimated cluster centers to be close to the true ones, and we thus need to reduce the threshold (i.e., the regularization parameter). Since in each iteration, λ^t is reduced by $\sqrt{2}$ and the sample size $|\mathcal{S}^t|$ is increased by a factor of 2, λ^t is proportional to $1/(\sqrt{|\mathcal{S}^t|})$, which is consistent with the classical theory on Lasso (Tibshirani, 1996). Finally, we choose to use ℓ_1 regularizer instead of ℓ_1 constraint because of its computational efficiency.

4. Main Theoretical Result

In this analysis, we discuss the sample complexity of recovering sparse centers in Algorithm 1. We will focus on the GMM model to make our analysis comparable to previous work. We however emphasize that according to our empirical study, the proposed algorithm also works well for a wide range of domains even when the assumption of normality does not hold. We will investigate in the future the theoretical property of the proposed algorithm for mixture model beyond GMM.

We assume that the data points in $\mathcal{D}$ are generated by a mixture of K Gaussians with centers $\mathbf{c}_1, \ldots, \mathbf{c}_K$ and mixing weights $\mu_1, \mu_2, \cdots, \mu_K$. Similar to most studies on clustering, we assume that the number of clusters K is known apriori. Following (Chaudhuri et al., 2009), we assume that each cluster center is a unit vector, i.e., $\|\mathbf{c}_i\| = 1, \forall i \in [K]$. We denoted by ρ the maximum overlap between different clusters, i.e.,

$$\rho = \max_{i \neq j} \mathbf{c}_i^\top \mathbf{c}_j.$$

It is easy to verify that the minimum distance between any two centers is $\sqrt{2(1-\rho)}$. We note that the introduction of ρ is closely related to the minimum separation requirement used in the theoretical analysis of learning a GMM (Hug-gins, 2011). Similar to GMM, we assume that each data point $\mathbf{x}_i$ is generated by the addition of a selected cluster center $\mathbf{c}_{k_i}$ and an independent Gaussian random noise $\mathbf{g}_i \sim \mathcal{N}(0, \sigma^2 I)$, i.e., $\mathbf{x}_i = \mathbf{c}_{k_i} + \mathbf{g}_i$.

For simplicity, we first assume that the initial centers are not too far away from the true centers. More specifically, the maximum distance between the initial centers and the true centers, defined as

$$\Delta^1 = \max_{1 \leq i \leq K} \|\widehat{\mathbf{c}}_i^1 - \mathbf{c}_i\|,$$

should be smaller than

$$\Delta_{\max} := \frac{1-\rho}{2} - \sigma\sqrt{5 \ln (3K)}.$$

Note that in order to ensure a sufficiently large value for $\Delta_{\max}$, σ has to be reasonably small, indicating that the proposed algorithm will not work appropriately with large noise. Similar restriction on clustering can be found in (Vempala & Wang, 2004; Brubaker & Vempala, 2008; Azizyan et al., 2013).

For clustering with sparse centers, we denote by Δ_* the solution to the following nonlinear equation

$$\frac{\Delta_*}{6\sqrt{2s}} = c_1 \exp\left(-\frac{(1 - 2\Delta_* - \rho)^2}{8(1+\Delta_*)^2\sigma^2}\right)(\eta_0 + \sigma\sqrt{\ln n}), \quad (3)$$

where c_1 is some universal constant. The definitions of $\Delta_{\max}$ and Δ_* arise from our theoretical analysis; see the detailed steps in the supplementary material. Following the analysis of compressive sensing (Donoho, 2006), we define the incoherence measure η_0 for all the cluster centers $\{\mathbf{c}_k\}_{k=1}^K$ as

$$\eta_0 = \max_{1 \leq i \leq K} \|\mathbf{c}_k\|_\infty,$$

where $\|\mathbf{c}_k\|_\infty$ measures the largest element in $\mathbf{c}_k$. Finally, we define

$$\mu_0 = \min_{1\le i\le K} \mu_i,$$

as the smallest mixing weight of K Gaussian mixtures. Now, the performance of our algorithm can be characterized by the following theorem.

Theorem 1. *Let $\epsilon \le 1/(6m)$ be a parameter that controls the success probability. Assume*

$$\Delta_* \le \Delta^1 \le \Delta_{\max}, \tag{4}$$

$$\frac{\Delta^1}{2\sqrt{2s}} \le \lambda^1 \le c\frac{\Delta^1}{2\sqrt{2s}}, \tag{5}$$

$$T \ge \max\left(\frac{18}{\mu_0}\ln\frac{2K}{\epsilon}, \frac{3c_2\eta_0}{\lambda^1}, \left(\frac{6c_3\sigma}{\lambda^1}\right)^2(\ln n + \ln d)\right) \tag{6}$$

where c, c_2 and c_3 are some universal constants that are defined in the supplementary material. Then, with a probability at least $1 - 6m\epsilon$, we have

$$\Delta^{m+1} = \max_{1\le i\le K} \|\widehat{\mathbf{c}}_i^{m+1} - \mathbf{c}_i\| \le \max\left(\Delta_*, \frac{c\Delta^1}{\sqrt{2^m}}\right).$$

Based on Theorem 1, we have the following corollary regarding the convergence rate of the proposed algorithm.

Corollary 1. *The convergence rate for Δ, the maximum difference between the optimal cluster centers and the estimated ones, is $O(\sqrt{(s\log d)/n})$ before reaching the optimal difference Δ_*.*

The proof and detailed analysis are shown in the supplementary file.

Remark First, the $O(\sqrt{(s\log d)/n})$ convergence rate implies that the sample complexity for accurately recovering s-sparse cluster centers is $O(s\log d)$, which is significantly lower than the dimensionality d. Similar results on sample complexity have also been obtained in sparse supervised learning (e.g. Lasso regression (Tibshirani, 1996; Zhao & Yu, 2006) and compressive sensing (Donoho, 2006)). Compared to the sample complexity $O(d)$ for accurately recovering cluster centers (Chaudhuri et al., 2007; 2009), $O(s\log d)$ is a significant improvement for high-dimensional data and sparse cluster centers. Compared to the minimax sample complexity $O(s^2\log d)$ developed in (Azizyan et al., 2013), our sample complexity has a lower dependence on s. However, we note that our sample complexity is developed for recovering cluster centers, while the sample complexity developed in (Azizyan et al., 2013) is for unsupervised classification error.

Second, Δ_* is defined as the best recovery accuracy that can be achieved by the proposed algorithm. To bound Δ_*, using the condition $\Delta_* \le (1-\rho)/2$, we have

$$\Delta_* \le 6\sqrt{2}c_1\exp\left(\frac{-(1-2\Delta_*-\rho)^2}{2(3-\rho)^2\sigma^2}\right)(\sqrt{s}\eta_0 + \sqrt{s}\sigma\sqrt{\ln n})$$

In the case when $\sigma\sqrt{d} = \Omega(1)$, namely the length of the random vector $\mathbf{g}_i$ is on the same order as the cluster center, we have $\Delta_* \le O(\exp(-O(d))$, which is a small value for high-dimensional data. We note that the residual error Δ_* cannot be removed even with increasing number of samples. This is mostly due to the greedy nature of our algorithm, i.e. each data point is assigned to the closest cluster, even when it is separated by a similar distance from all the clusters. In contrast, for GMM, it is possible to recover the cluster centers with arbitrary accuracy provided sufficiently large number of samples.

5. Experiments

In this section, we first conduct experiments with simulated data to verify that the initial centers computed by hierarchical clustering algorithm are not too far away from the true centers. We then compare the proposed clustering algorithm to several clustering algorithms that are used for high-dimensional data on several benchmark datasets.

According to our analysis, we need to ensure that the initial centers are not too far away from the true centers (i.e., $\Delta^1 \le \Delta_{\max}$). To satisfy this condition, we find the initial cluster centers by applying a hierarchical clustering algorithm (Murtagh, 1984) to a small subset of data, a practice commonly used in many clustering algorithms such as k-means (Jain et al., 1999). In more detail, we first randomly sample $[5K\log n]$ instances, and run the hierarchical clustering algorithm against the sampled data instances. Since the sample size used by the hierarchical clustering is relatively small, its running time is miniscule when compared to clustering the entire dataset.

Experimental Results We first conduct experiments with simulated data to verify that the initial centers determined by hierarchical clustering algorithm are not too far away from the true centers, i.e., they satisfy $\Delta^1 \le \Delta_{\max}$. To this end, for a fixed dimensionality d, we create 2 d-dimensional binary sparse cluster centers $\mathbf{c}_1$ and $\mathbf{c}_2$. In the representation of each cluster center, we randomly select $1,000$ entries from the d dimensional vector, and set their values to be 1; the remaining entries are set to 0. We further normalize the centers onto the sphere of a unit ball. In addition, for each cluster center $\mathbf{c}_i$, we generate $10,000$ data points by adding Gaussian noise (sampled from $\mathcal{N}(0,\sigma^2 I)$ with $\sigma = 0.002$) to the cluster center $\mathbf{c}_i$. A hierarchical clustering algorithm is applied to the randomly sampled data points to compute the initial cluster centers. We vary dimensionality d in range {20,000, 50,000, 100,000,

Table 1. A comparison of Δ^1 and $\Delta_{\max}$ with different dimensionality d. The number of samples n is set to $10,000$.

d	20K	50K	100K	200K	400K
Δ^1	0.097	0.161	0.236	0.338	0.459
$\Delta_{\max}$	0.474	0.484	0.488	0.492	0.493

200,000, 400,000}. Table 1 shows the magnitudes of Δ^1 and $\Delta_{\max}$ with different values of d. We observe that in all the cases, $\Delta^1 < \Delta_{\max}$ holds, verifying that the hierarchical clustering algorithm is a promising way to select the initial cluster centers.

We now report experimental results on four real-world high-dimensional benchmark datasets where the dimensionalities are significantly larger than the numbers of samples. Below, we briefly describe each of the testbeds.

- ***Yale*** database[1] contains 165 grayscale face images of 15 people. For each people, 11 images were taken with one per different facial expressions or configurations. Each image has size 64×64, leading to a $4,096$-dimensional vector.
- ***Reuters-21578*** dataset[2] contains $21,578$ documents in 135 categories. After discarding the documents belonging to multiple categories, $8,293$ documents are left and they belong to 65 imbalanced clusters.
- ***TDT2*** dataset contains the top 30 categories, in terms of size, of TDT2 corpus[3]. These 30 categories are comprised of $9,394$ documents in a $36,771$ dimensional space.
- ***TREC 05*** dataset (Cormack & Lynam, 2005) contains $823,470$ binary variables describing the presence of word tokens in $92,189$ email messages. All the emails belong to one of the two classes: spam or non-spam.

Table 2 summarizes the statistics of all the testbeds used in our study.

We compare the proposed clustering algorithm to the following five clustering approaches that are applicable to high-dimensional data. They are (a) **KM**, the k-means algorithm (Lloyd, 1982), (b) **PCAKM**, that first applies PCA to project data points to a low-dimensional space, with 95% of variance kept, before applying the k-means algorithm, (c) **LDA**, the latent Dirichlet allocation (LDA) algorithm (Blei et al., 2003), (d) **LSC**, the large scale spectral clustering with landmark-based representation (Chen & Cai, 2011), and (e) **RegKM** (Sun et al., 2012), a group lasso regularized k-means algorithm. Some of the state-of-the-art subspace clustering algorithms (e.g. sparse subspace clustering (SCC) (Elhamifar & Vidal, 2009) and generalized principal component analysis (GPCA) (Vidal et al., 2005)) were not included in our comparative study because of their very high computational cost for high-dimensional data. We refer to the proposed clustering algorithm as High-dimensional Clustering with Sparse Centers, or **HDSC** for short. The proposed clustering algorithm as well as all the baseline algorithms are implemented in Matlab and all the experiments are performed on a Xeon 2.40 GHz processor with 16 GB memory. Each experiment is repeated five times, and the clustering performance as well as the running time averaged over five trials are reported. We mark the results as N/A if an algorithm did not output the results within 5 hours.

Table 2. Description of Datasets

Name	#Instances	#Features	#Clusters
Yale	165	4,096	15
Reuters-21578	8,293	18,933	65
TDT2	9,394	36,771	30
TREC 05	92,189	823,470	2

Table 3 summarizes the performance of the proposed clustering algorithm HDSC and the baseline algorithms. We first observe that the proposed clustering algorithm HDSC is significantly more efficient than all the baseline clustering algorithms on all the four datasets. For example, HDSC clusters the *TDT2* database in less than 1 second. As a comparison, all of the five baseline algorithms take more than 40 seconds to partition this dataset. Although we note that the running time of spectral clustering can be significantly reduced by applying randomized methods (Halko et al., 2011), it usually works only in the case of approximate low rank matrices. Unfortunately, this assumption does not hold for document data, whose eigenspectrums usually have a long tail. In addition, compared to the standard k-means algorithm, the proposed HDSC algorithm is at least 50 times more efficient, due to the fact that HDSC only needs one pass over data points for cluster center estimation. In addition to efficiency, we also observe that HDSC outperforms the baseline methods in three out of four benchmark datasets. We note that although PCAKM improves the performance of standard k-means by dimensionality reduction, it still takes the longest running time and yields worse results than the proposed algorithm and LSC. This result indicates that blind dimensionality reduction may not be optimal for clustering high-dimensional data with sparse cluster centers. Overall, the empirical result-

[1] http://vision.ucsd.edu/content/yale-face-database

[2] http://www.daviddlewis.com/resources/testcollections/reuters21578/

[3] http://projects.ldc.upenn.edu/TDT2/data-release.html

Table 3. Average performance of the proposed algorithm (HDSC) and the baseline algorithms (KM (Lloyd, 1982), PCAKM, LDA (Blei et al., 2003), LSC (Chen & Cai, 2011), and RegKM (Sun et al., 2012)) on four benchmark datasets

Datasets		HDSC	KM	PCAKM	LDA	LSC	RegKM
Yale	NMI	**0.51**	0.49	0.50	0.49	**0.51**	0.49
	CPU time (s)	**0.06**	5.8	7.4	6.7	2.9	3.0
Reuters-21578	NMI	**0.43**	0.40	0.41	0.41	0.42	0.40
	CPU time (s)	**0.5**	29	79	66	21	32
TDT2	NMI	0.66	0.62	0.65	0.66	**0.68**	0.65
	CPU time (s)	**0.9**	52	221	115	47	101
TREC 05	NMI	**0.22**	0.16	N/A	0.21	0.20	0.19
	CPU time (s)	**7.8**	412	N/A	932	614	671

s demonstrate both the efficiency and effectiveness of the proposed algorithm for high-dimensional data clustering.

6. Conclusions

In this paper, we propose a framework for efficient clustering of high dimensional data with sparse cluster centers. The key idea is to cast the high-dimensional data clustering problem into the problem of recovering the optimal cluster centers, and iteratively estimating the cluster centers using disjoint subsets with exponentially increasing number of data points. This is a key step to ensure that the estimated cluster centers will be close to the optimal cluster centers. To satisfy the assumption that the true cluster centers are sparse with no more than s non-zero elements, we introduce an ℓ_1 regularizer when updating the cluster centers. We show that with a high probability, the proposed clustering algorithm can accurately recover the optimal cluster centers by only going through $O(s \log d)$ data instances. This logarithmic dependence on data dimensionality is a significant improvement over state of the art, making the proposed clustering algorithm both efficient and effective for high-dimensional clustering problem. Our empirical studies with several real-world high-dimensional datasets show the promising performance of the proposed clustering algorithm. We plan to further improve the algorithm for streaming data where all the data points can not be stored in the memory.

Acknowledgement:

This work was supported in part by the National Science Foundation (IIS-1251031) and the Office of Naval Research (N00014-11-1-0100 and N00014-12-1-0431).

A. Proof of Theorem 1

Our analysis is based on induction. Let $\widehat{\mathbf{c}}_1, \ldots, \widehat{\mathbf{c}}_K$ be the K cluster centers obtained at the t-th iteration. Define Δ^t to be the maximum difference between the estimated cluster centers and the true ones, i.e.,

$$\Delta^t = \max_{i \in [K]} \|\widehat{\mathbf{c}}_i - \mathbf{c}_i\|.$$

We denote by $\mathcal{C}_k$ the support of $\mathbf{c}_k$. Let $s = \max |\mathcal{C}_k|$ be the maximal number of non-zero entries in all the sparse centers.

We develop the following lemma to guarantee the correctness of the induction step.

Lemma 1. *Let Δ^t be the maximum difference between the optimal cluster centers and the ones estimated at iteration t, and $\epsilon \in (0, 1)$ be the failure probability. Assume*

$$\Delta^t \leq \Delta_{\max}, \tag{7}$$

$$|\mathcal{S}^t| \geq \frac{18}{\mu_0} \ln \frac{2K}{\epsilon}, \tag{8}$$

$$\lambda^t \geq c_1 \exp\left(-\frac{(1 - 2\Delta^t - \rho)^2}{8(1 + \Delta^t)^2 \sigma^2}\right)(\eta_0 + \sigma\sqrt{\ln |\mathcal{S}^t|}) + \frac{c_2 \eta_0}{|\mathcal{S}^t|} + c_3 \sigma \frac{\sqrt{\ln |\mathcal{S}^t|} + \sqrt{\ln d}}{\sqrt{|\mathcal{S}^t|}}, \tag{9}$$

for some universal constants c_1, c_2 and c_3. Then with a probability $1 - 6\epsilon$, we have

$$\Delta^{t+1} \leq 2\sqrt{s}\lambda^t.$$

The detailed proof of this Lemma can be found in the supplementary materials.

To prove Theorem 1, we need to show that if $\{\Delta^1, \ldots, \Delta^m\} \geq \Delta_*$, (9) is always true. Consequently, we can apply Lemma 1 in each epoch, leading to

$$\Delta^{m+1} \leq 2\sqrt{s}\lambda^m = 2\sqrt{s}\frac{\sqrt{2}\lambda_1}{\sqrt{2^m}} \overset{(5)}{=} \frac{c\Delta^1}{\sqrt{2^m}}.$$

We show that (9) is true by induction. Given the definition

of Δ_* and $\Delta_1 \geq \Delta_*$, it naturally follows that

$$\begin{aligned} & c_1 \exp\left(-\frac{(1-2\Delta^1-\rho)^2}{8(1+\Delta^1)^2\sigma^2}\right)(\eta_0+\sigma\sqrt{\ln n}) \\ \overset{(3)}{\leq} & \frac{\Delta^1}{6\sqrt{2s}} \overset{(5)}{\leq} \frac{\lambda^1}{3}. \end{aligned} \tag{10}$$

From the definition of T in (6), we have

$$\frac{c_2\eta_0}{T} + c_3\sigma\frac{\sqrt{\ln T}+\sqrt{\ln d}}{\sqrt{T}} \leq \frac{2\lambda^1}{3}. \tag{11}$$

Following (10) and (11), we know (9) holds for $t=1$.

Assume that with a probability at least $1-6(t-1)\epsilon$,

$$\begin{aligned} \lambda^t \geq & c_1 \exp\left(-\frac{(1-2\Delta^t-\rho)^2}{8(1+\Delta^t)^2\sigma^2}\right)(\eta_0+\sigma\sqrt{\ln n}) \\ & + \frac{c_2\eta_0}{|\mathcal{S}^t|} + c_3\sigma\frac{\sqrt{\ln|\mathcal{S}^t|}+\sqrt{\ln d}}{\sqrt{|\mathcal{S}^t|}}, \end{aligned}$$

holds for some $t \geq 1$. Then, based on Lemma 1, with a probability at least $1-6t\epsilon$, we have

$$\Delta^{t+1} \leq 2\sqrt{s}\lambda^t. \tag{12}$$

Given the definition of Δ_*, we know that for any $\Delta^{t+1} \geq \Delta_*$, we have

$$\begin{aligned} & c_1 \exp\left(-\frac{(1-2\Delta^{t+1}-\rho)^2}{8(1+\Delta^{t+1})^2\sigma^2}\right)(\eta_0+\sigma\sqrt{\ln n}) \\ \leq & \frac{\Delta^{t+1}}{6\sqrt{2s}} \overset{(12)}{\leq} \frac{\lambda^t}{3\sqrt{2}} = \frac{\lambda^{t+1}}{3}. \end{aligned} \tag{13}$$

Similar to (11), we have

$$\begin{aligned} & \frac{c_2\eta_0}{|\mathcal{S}^{t+1}|} + c_3\sigma\frac{\sqrt{\ln|\mathcal{S}^{t+1}|}+\sqrt{\ln d}}{\sqrt{|\mathcal{S}^{t+1}|}} \\ \leq & \frac{1}{\sqrt{2^t}}\left(\frac{c_2\eta_0}{T} + c_3\sigma\frac{\sqrt{\ln n}+\sqrt{\ln d}}{\sqrt{T}}\right) \overset{(6)}{\leq} \frac{2\lambda^{t+1}}{3}. \end{aligned} \tag{14}$$

Combining (13) and (14), we have, with a probability at least $1-6t\epsilon$, (9) holds for $t+1$.

References

Achlioptas, D. and McSherry, F. On spectral learning of mixtures of distributions. In *COLT*, pp. 458–469, 2005.

Agrawal, R., Gehrke, J., Gunopulos, D., and Raghavan, P. Automatic subspace clustering of high dimensional data for data mining applications. In *SIGMOD*, pp. 94–105, 1998.

Arora, Sanjeev and Kannan, Ravi. Learning mixtures of arbitrary gaussians. In *STOC*, pp. 247–257, 2001.

Azizyan, M., Singh, A., and Wasserman, L. A. Minimax theory for high-dimensional gaussian mixtures with sparse mean separation. In *NIPS*, pp. 2139–2147, 2013.

Balcan, M., Blum, A., and Gupta, A. Finding low error clusterings. In *COLT*, 2009a.

Balcan, M., Blum, A., and Gupta, A. Approximate clustering without the approximation. In *SODA*, pp. 1068–1077, 2009b.

Balcan, M., Blum, A., and Gupta, A. Clustering under approximation stability. *Journal of the ACM*, 60(2):8:1–8:34, 2013.

Belkin, M. and Sinha, K. Polynomial learning of distribution families. In *FOCS*, pp. 103–112, 2010.

Blei, David M., Ng, Andrew Y., and Jordan, Michael I. Latent dirichlet allocation. *JMLR*, 3:993–1022, 2003.

Blum, A., Coja-Oghlan, A., Frieze, A., and Zhou, S. Separating populations with wide data: a spectral analysis. *Electronic Journal of Statistics*, 3:73–113, 2009.

Brubaker, S. C. and Vempala, S. Isotropic pca and affine-invariant clustering. In *FOCS*, pp. 551–560, 2008.

Chaudhuri, K. Learning mixtures of distributions. *PhD dissertation, EECS department, UC Berkeley*, 2007.

Chaudhuri, K., Halperin, E., Rao, S., and Zhou, S. A rigorous analysis of population stratification with limited data. In *SODA*, pp. 1046–1055, 2007.

Chaudhuri, K., Dasgupta, S., and Vattani, A. Learning mixtures of gaussians using the k-means algorithm. *CoRR*, abs/0912.0086, 2009.

Chen, Xinlei and Cai, Deng. Large scale spectral clustering with landmark-based representation. In *AAAI*, 2011.

Cormack, Gordon V. and Lynam, Thomas R. Spam corpus creation for TREC. In *CEAS*, 2005.

Dasgupta, A., Hopcroft, J. E., Kleinberg, J. M., and Sandler, M. On learning mixtures of heavy-tailed distributions. In *FOCS*, pp. 491–500, 2005.

Dasgupta, Sanjoy. Learning mixtures of gaussians. In *FOCS*, pp. 634–644, 1999.

Dasgupta, Sanjoy and Schulman, Leonard J. A two-round variant of em for gaussian mixtures. In *UAI*, pp. 152–159, 2000.

De Smet, Frank, Mathys, Janick, Marchal, Kathleen, Thijs, Gert, De Moor, Bart, and Moreau, Yves. Adaptive quality-based clustering of gene expression profiles. *Bioinformatics*, 18(5):735–746, 2002.

Donoho, D.L. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.

Elhamifar, E. and Vidal, R. Sparse subspace clustering. In *CVPR*, pp. 2790–2797, 2009.

Ertöz, Levent, Steinbach, Michael, and Kumar, Vipin. Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data. In *SDM*. SIAM, 2003.

Figueiredo, M. and Jain, A. K. Unsupervised learning of finite mixture models. *PAMI*, 24(3):381–396, 2002.

Hale, E. T., Wotao, Y., and Zhang, Y. Fixed-point continuation for L1-minimization: methodology and convergence. *SIAM J. on Optimization*, 19(3):1107–1130, 2008.

Halko, N., Martinsson, P., and Tropp, J. A. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288, 2011.

Hsu, Daniel and Kakade, Sham M. Learning mixtures of spherical gaussians: moment methods and spectral decompositions. In *ITCS*, pp. 11–20, 2013.

Huggins, J. Provably learning mixtures of gaussians and more. Technical report, Columbia University, 2011.

Jain, A. K., Murty, M. N., and Flynn, P. J. Data clustering: A review. *ACM Comput. Surv.*, 31(3):264–323, 1999.

Jin, R., Yang, T., and Zhu, S. A new analysis of compressive sensing by stochastic proximal gradient descent. *CoRR*, abs/1304.4680, 2013.

Johnstone, Iain M. High dimensional statistical inference and random matrices. In *Proceedings of the international congress of mathematicians (ICM)*, pp. 307–333, 2007.

Kalai, A. T., Moitra, A., and Valiant, G. Disentangling gaussians. *Commun. ACM*, 55(2):113–120, 2012.

Kannan, R., Salmasian, H., and Vempala, S. The spectral method for general mixture models. In *COLT*, pp. 444–457, 2005.

Keerthi, S. Sathiya and DeCoste, Dennis. A modified finite Newton method for fast solution of large scale linear svms. *JMLR*, 6:341–361, 2005.

Lindsey, B. G. *Mixture Models: Theory, Geometry and Applications*. IMS, 1996.

Lloyd, Stuart P. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–136, 1982.

Mika, S., Schölkopf, B., Smola, A. J., Müller, K., Scholz, M., and Rätsch, G. Kernel pca and de-noising in feature spaces. In *NIPS*, pp. 536–542, 1998.

Murtagh, F. Complexities of hierarchic clustering algorithms: state of the art. *Computational Statistics Quarterly*, 1(2):101–113, 1984.

Pan, W. and Shen, X. Penalized model-based clustering with application to variable selection. *JMLR*, 8:1145–1164, 2007.

Sculley, D. Web-scale k-means clustering. In *WWW*, pp. 1177–1178, 2010.

Srebro, N., Shakhnarovich, G., and Roweis, S. T. An investigation of computational and informational limits in gaussian mixture clustering. In *ICML*, pp. 865–872, 2006.

Srebro, Nathan. Are there local maxima in the infinite-sample likelihood of gaussian mixture estimation? In *COLT*, pp. 628–629, 2007.

Sun, W., Wang, J., and Fang, Y. Regularized k-means clustering of high-dimensional data and its asymptotic consistency. *Electronic Journal of Statistics*, 6:148–167, 2012.

Tibshirani, Robert. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.

Tsybakov, Alexandre B. *Introduction to Nonparametric Estimation*. Springer, 2008.

Vempala, S. and Wang, G. A spectral algorithm for learning mixture models. *J. Comput. Syst. Sci.*, 68(4):841–860, 2004.

Vidal, René, Ma, Yi, and Sastry, Shankar. Generalized principal component analysis (gpca). *PAMI*, 27(12): 1945–1959, 2005.

Wang, Wei and Yang, Jiong. Mining high-dimensional data. In *Data Mining and Knowledge Discovery Handbook*, pp. 793–799. Springer, 2005.

Witten, Daniela M and Tibshirani, Robert. A framework for feature selection in clustering. *Journal of the American Statistical Association*, 105(490), 2010.

Wright, S. J., Nowak, R. D., and Figueiredo, M. A. T. Sparse reconstruction by separable approximation. *IEEE Transactions on Signal Processing*, 57(7):2479–2493, 2009.

Zhao, Peng and Yu, Bin. On model selection consistency of lasso. *JMLR*, 7:2541–2563, 2006.