Large Scale Sparse Clustering

Ruqi Zhang and **Zhiwu Lu***
Beijing Key Laboratory of Big Data Management and Analysis Methods
School of Information, Renmin University of China, Beijing 100872, China
zhiwu.lu@gmail.com

Abstract

Large-scale clustering has found wide applications in many fields and received much attention in recent years. However, most existing large-scale clustering methods can only achieve mediocre performance, because they are sensitive to the unavoidable presence of noise in the large-scale data. To address this challenging problem, we thus propose a large-scale sparse clustering (LSSC) algorithm. In this paper, we choose a two-step optimization strategy for large-scale sparse clustering: 1) k-means clustering over the large-scale data to obtain the initial clustering results; 2) clustering refinement over the initial results by developing a spare coding algorithm. To guarantee the scalability of the second step for large-scale data, we also utilize nonlinear approximation and dimension reduction techniques to speed up the sparse coding algorithm. Experimental results on both synthetic and real-world datasets demonstrate the promising performance of our LSSC algorithm.

1 Introduction

Clustering is an important data analysis tool in many areas of science and technology. Recently, large-scale clustering is drawing more and more attention because of its effectiveness and efficiency in coping with large-scale data. A number of clustering algorithms have been developed to efficiently group large-scale data [Li *et al.*, 2015; Chen and Cai, 2011; Zhang and Rudnicky, 2002; Wang *et al.*, 2011; Chitta *et al.*, 2011; Aggarwal *et al.*, 2003; Guha *et al.*, 2001]. These methods manage to reduce the computational complexity of clustering algorithm using different strategies. However, most existing large-scale clustering algorithms can only achieve mediocre performance, because they are sensitive to the unavoidable noise in the real-world data. Considering that the noise (i.e. an unstructured subset of data points) can disrupt the cluster structure of the data, detecting the accurate cluster structure becomes difficult in this case.

In the literature, some robust methods have been developed for clustering potentially noisy data. For example, using trimming to separate clusterable parts of the data from unstructured ones by fixing some noise-set size, several algorithms have been proposed [García-Escudero *et al.*, 2008; García-Escudero and Gordaliza, 1999; Cuesta-Albertos *et al.*, 1997]. However, these algorithms suffer from exponential computational complexity, and have to be compromised for efficient heuristic searches that have no performance guarantees. Moreover, transductive warping is also used in spectral clustering to reduce the influence of noise [Li *et al.*, 2007]. By applying data warping to reshape the noisy data, the block structure (destroyed by noise) of the affinity matrix can be recovered. However, it is very computationally expensive and is hard to apply to large-scale data, due to that it requires transductive warping of each data point.

To cope with large-scale noisy data, we thus propose a large-scale sparse clustering (LSSC) algorithm. In this paper, we choose a two-step optimization strategy for large-scale sparse clustering. In the first step, we adopt k-means clustering to obtain the initial clustering results. The resulting indicator matrix C ($C_{ij} = 1$ if the i-th data point is in the j-th cluster, and $C_{ij} = 0$ otherwise) is used as the inputs of the second step. In the second step, we formulate an L_1-optimization problem [Wright *et al.*, 2009] over the indicator matrix based on sparse coding [Lee *et al.*, 2006; Olshausen and Field, 1997] to rectify the initial clustering results. Although there exist previous methods [Elhamifar and Vidal, 2009; Ramirez *et al.*, 2010; Wang *et al.*, 2015] that also use sparse coding to improve the clustering robustness, they are very time consuming, especially for large-scale data. In contrast, our L_1-optimization problem defined on the indicator matrix can be solved very efficiently. Specifically, inspired by the superiority of spectral clustering [Ng *et al.*, 2002; Filippone *et al.*, 2008; Lu and Peng, 2013], we limit the solution of clustering to the space spanned by a small set of leading eigenvectors of the Laplacian matrix. Based on this dimension reduction technique, we thus significantly reduce the time complexity of our L_1-optimization problem.

However, finding the leading eigenvectors of the Laplacian matrix is still time consuming on large-scale data. We thus make use of the nonlinear approximation technique to speed up this step. Specifically, given a limited number of clustering centers (obtained by k-means), we represent each data point as the nonlinear approximation of these centers and then derive the Laplacian matrix as a symmetrical decomposition

*Corresponding author

form. Based on this special definition of the Laplacian matrix, we are able to find the leading eigenvectors in linear time complexity. In summary, by combining nonlinear approximation and dimension reduction together, we develop a large-scale sparse clustering (LSSC) algorithm of linear time and space complexity. Experimental results on both synthetic and real-world datasets demonstrate the promising performance of the proposed LSSC algorithm.

To emphasize our main contributions, we summarize the following distinct advantages of our LSSC algorithm:

- We have developed a novel large-scale sparse clustering algorithm which is shown to be very robust against the noise in large-scale data. In fact, the challenging problem of clustering over large-scale noisy data has been rarely studied in the literature.
- We have proposed a new two-step optimization strategy for clustering over large-scale noisy data. Since the two steps (i.e. initial clustering and clustering refinement) tend to learn the cluster structure from different aspects, this strategy can detect the cluster structure more accurately for large-scale noisy data.
- Our LSSC algorithm has a wide use in various clustering applications. Given that there is no single clustering algorithm suitable for all types of clustering applications, we can choose the suitable fundamental clustering algorithm for the first step of LSSC algorithm depending on the type of clustering applications.

The remainder of this paper is organized as follows. In Section 2, we develop our large-scale sparse clustering algorithm. The experimental results on synthetic and real-world datasets are presented in Sections 3 and 4, respectively. We give our conclusion in Section 5.

2 Large-Scale Sparse Clustering

In this section, we propose our large-scale sparse clustering (LSSC) algorithm. We first introduce our two-step optimization strategy and give the problem formulation. We further develop an efficient algorithm by combining the nonlinear approximation and dimension reduction techniques.

2.1 Problem Formulation

Let $X = [\mathbf{x}_1, \cdots, \mathbf{x}_n] \in R^{m \times n}$ be a dataset that has l clusters. In the first step of our algorithm, we apply k-means to obtain the initial clustering results. The resulting clustering indicator matrix is denoted as $C \in R^{n \times l}$, where $C_{ij} = 1$ if the i-th data point is in the j-th cluster and $C_{ij} = 0$ otherwise. The first step used in our algorithm has the following three advantages. Firstly, the initial clustering can give us rough cluster structures of the data. We can then define an optimization problem directly based upon C in the second step. The optimization problem defined on the indicator matrix C will help us to develop a very efficient noise-robust clustering algorithm. Secondly, the final results of our algorithm are the combination of the initial clustering and clustering refinement. Since both steps tend to learn the cluster structure from different aspects, our algorithm can thus obtain more superior performance on detecting the cluster structures. Thirdly, given that different clustering applications employ very different clustering algorithms, the suitable fundamental clustering method used for the first step can be determined depending on different situations.

In the second step of our algorithm, we perform clustering refinement over the initial results. Considering the superiority of spectral clustering, we formulate the clustering refinement problem based on the graph model. Specifically, we model the whole dataset X as a graph $\mathcal{G} = \{\mathcal{V}, W\}$ with its vertex set $\mathcal{V} = X$ and weight matrix $W = [w_{ij}]_{n \times n}$, where w_{ij} denotes the affinity relation of data points $\mathbf{x}_i$ and $\mathbf{x}_j$. In this paper, the weight of the edge between $\mathbf{x}_i$ and $\mathbf{x}_j$ is defined using the Gaussian kernel function:

$$w_{ij} = \exp(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2\sigma^2}) \tag{1}$$

where the variance σ is a free parameter that can be determined empirically. In fact, we can adopt various graph construction methods [Wang and Zhang, 2008; Cheng *et al.*, 2010] to eliminate the need to tune this parameter. Let $D \in R^{n \times n}$ be the degree matrix whose i-th diagonal element is $d_{ii} = \sum_{j=1}^{n} w_{ij}$ and I be an $n \times n$ identity matrix. The normalized Laplacian matrix of the graph $\mathcal{G}$ is given by

$$L = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}} \tag{2}$$

As a nonnegative definite matrix, L can be decomposed as

$$L = V \Sigma V^T \tag{3}$$

where V is an $n \times n$ orthogonal matrix with each column being an eigenvector of L, and Σ is an $n \times n$ diagonal matrix with its diagonal element Σ_{ii} being an eigenvalue of L (sorted as $0 \leq \Sigma_{11} \leq \cdots \leq \Sigma_{nn}$).

In spectral clustering, the k-means clustering is then applied over the first p smallest eigenvectors (i.e. the first p columns of V) to obtain the final results. However, the traditional spectral clustering algorithm tends to be significantly affected by the noise which would destroy the cluster structures of the data. To solve this problem and also keep the superiority of spectral clustering, we formulate the problem of clustering refinement as an L_1-optimization problem so that the robustness can be introduced into our algorithm. The problem formulation is described as follows.

We first denote L as a symmetrical decomposition form based on the above eigenvalue decomposition:

$$L = (\Sigma^{\frac{1}{2}} V^T)^T \Sigma^{\frac{1}{2}} V^T = B^T B \tag{4}$$

where $B = \Sigma^{\frac{1}{2}} V^T$. Since B is computed with all the eigenvectors of L, we can regard B as being explicitly defined based upon the manifold structure of the data. We further formulate the clustering refinement problem in the second step of our LSSC algorithm as follows:

$$\min_{Y} Q(Y) = \frac{1}{2}\|Y - C\|_{fro}^2 + \lambda \|BY\|_1 \tag{5}$$

where $\| \cdot \|_{fro}$ denotes the Frobenius norm of a matrix, λ is a positive regularization parameter, and Y denotes the optimal probabilistic clustering matrix. The first term of the objective function $Q(Y)$ denotes the reconstruction error, while the second term of $Q(Y)$ is closely related to the well-known Laplacian regularization [Lu and Peng, 2011; 2013] used for graph-based learning.

The above L_1-optimization problem formulation for clustering refitment has the following three advantages. Firstly, we transform the clustering refinement problem into an L_1-optimization problem so that the robustness can be introduced into our algorithm. Secondly, due to the first step of initial clustering, we can directly define the objective function on the indicator matrix C and thus perform noise reduction over C. Thirdly, by introducing Laplacian regularization, we can take the advantage of spectral clustering to reveal the cluster structure for large-scale clustering. More notably, we can readily limit the solution Y to the space spanned by the leading eigenvectors of the Laplacian matrix and thus transform the above L_1-optimization problem into a general sparse coding problem. This transformation will be described below.

2.2 The Proposed Algorithm

To keep the scalability of clustering refinement, we can reduce the dimension of Y dramatically by requiring it to take the form of $Y = V_p H_p$, where V_p is an $n \times p$ matrix whose columns are the p leading eigenvectors with smallest eigenvalues (i.e. the first p columns of V). In fact, such dimension reduction can ensure that Y is as smooth as possible, according to spectral theory. Hence, the objective function of clustering refinement can now be derived from Eq. (5) as:

$$\min_{H_p} Q(H_p) = \frac{1}{2}\|V_p H_p - C\|_{fro}^2 + \lambda \|\Sigma^{\frac{1}{2}} V^T V_p H_p\|_1 \quad (6)$$

To transform this L_1-optimization problem into a generalized sparse coding problem, we then decompose it into the following l independent subproblems:

$$\min_{H_{.i}} Q(H_{.i}) = \frac{1}{2}\|V_p H_{.i} - C_{.i}\|_2^2 + \lambda \|\Sigma^{\frac{1}{2}} V^T V_p H_{.i}\|_1 \quad (7)$$

where $1 \leq i \leq l$. Writing out the objective function, we have:

$$\begin{aligned} Q(H_{.i}) &= \frac{1}{2}\|V_p H_{.i} - C_{.i}\|_2^2 + \lambda \| \sum_{k=1}^{p} \Sigma^{\frac{1}{2}} (V^T V_{.k}) H_{ki} \|_1 \\ &= \frac{1}{2}\|V_p H_{.i} - C_{.i}\|_2^2 + \lambda \sum_{k=1}^{p} \Sigma_{kk}^{\frac{1}{2}} \|H_{ki}\|_1 \quad (8) \end{aligned}$$

The first term of $Q(H_{.i})$ denotes the reconstruction error, while the second term denotes the weighted L_1-norm sparsity regularization over the reconstruction coefficients.

Hence, our original L_1-optimization problem has been transformed into a generalized sparse coding problem $H_{.i}^* = \arg\min_{H_{.i}} Q(H_{.i})$, which can be solved efficiently by many standard algorithms. It should be noted that the formulation $Y = V_p H_{.i}$ used in Eq. (8) has two distinct advantages. Firstly, we can explain our clustering refinement in the framework of sparse coding. In fact, the second term of $Q(H_{.i})$ corresponds to both Laplacian regularization and sparsity regularization. We thus obtain novel noise-robust clustering by unifying these two types of regularization. Secondly, Since $Q(H_{.i})$ is minimized with respect to $H_{.i} \in R^p$ ($p \ll n$), we can readily develop fast sparse coding algorithms for our clustering refinement. Specifically, although many sparse coding algorithms scale polynomially with respect to p, they only have linear time complexity with respect to n. More importantly, we have eliminated the need to compute the full matrix B in Eq. (5), which is especially suitable for clustering on large-scale data. In fact, we only need to find the p leading eigenvectors of L.

However, it is still very time consuming to find the p leading eigenvectors of L on large-scale data. To keep the scalability of our algorithm, we thus exploit the following nonlinear approximation technique. Given k clustering centers $\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_k$ obtained by k-means clustering over the dataset X, we find the approximation $\hat{\mathbf{x}}_i$ of any data point $\mathbf{x}_i$ by Nadaraya-Watson kernel regression [Härdle, 1992]:

$$\hat{\mathbf{x}}_i = \sum_{j=1}^{k} z_{ij} \mathbf{u}_j \quad (9)$$

where $Z = [z_{ij}]_{n \times k}$ collects the regression coefficients. A natural assumption here is that z_{ij} should be larger if $\mathbf{x}_i$ is closer to $\mathbf{u}_j$. We can emphasize this assumption by setting $z_{ij} = 0$ as $\mathbf{u}_j$ is not among the $r(\leq k)$ nearest neighbors of $\mathbf{x}_i$. This restriction naturally leads to a sparse matrix Z. Let $U(i)$ denotes the indexes of r clustering centers that are nearest to $\mathbf{x}_i$. We compute $z_{ij} (j \in U(i))$ as:

$$z_{ij} = \frac{K_\sigma(\mathbf{x}_i, \mathbf{u}_j)}{\sum_{j' \in U(i)} K_\sigma(\mathbf{x}_i, \mathbf{u}_{j'})} \quad (10)$$

where $K_\sigma(.)$ is a kernel function with a bandwidth σ. Here, we adopt the Gaussian kernel $K_\sigma(\mathbf{x}_i, \mathbf{u}_j) = \exp(-\frac{\|\mathbf{x}_i - \mathbf{u}_j\|_2^2}{2\sigma^2})$ which is one of the most commonly used kernel functions. In this paper, the same parameter σ are selected for the Gaussian kernels used in both Eq. (1) and Eq. (10).

Let $Z = \hat{Z} D_z^{-\frac{1}{2}}$ and D_z be a $k \times k$ diagonal matrix whose i-th diagonal element is $d_{ii} = \sum_{j=1}^{n} z_{ji}$. The weight matrix $W \in R^{n \times n}$ of the graph $\mathcal{G}$ over the dataset X can now be computed as follows:

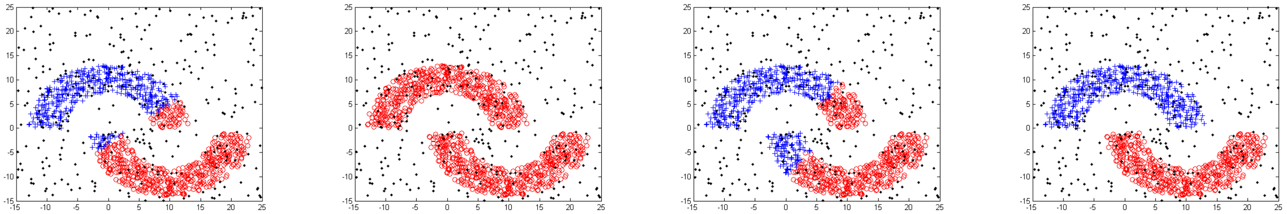
$$W = \hat{Z} \hat{Z}^T \quad (11)$$

Since each row of Z sums up to 1, the degree matrix of $\mathcal{G}$ is I and the normalized Laplacian matrix L is $I - W$. This means that finding the p smallest eigenvectors of L is equivalent to finding the p largest eigenvectors of W. Let the singular value decomposition (SVD) of $\hat{Z}$ be:

$$\hat{Z} = V_z \Sigma_z U_z^T \quad (12)$$

where $\Sigma_z = \text{diag}(\sigma_1, \cdots, \sigma_k)$ with σ_i being a singular value of $\hat{Z}$ (sorted as $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_k \geq 0$), V_z is an $n \times k$ matrix with each column being a left singular vector of $\hat{Z}$, and U_z is a $k \times k$ matrix with each column being a right singular vector of $\hat{Z}$. It is easy to check that each column of V_z is an eigenvector of $W = \hat{Z}\hat{Z}^T$, and each column of U_z is an eigenvector of $\hat{Z}^T\hat{Z}$ (the eigenvalues are $\sigma_1^2, \cdots, \sigma_k^2$ in both cases). Since $\hat{Z}^T\hat{Z} \in R^{k \times k}$, we can compute U_z within $O(k^3)$ time. V_z can then be computed as:

$$V_z = \hat{Z} U_z \Sigma_z^{-1} \quad (13)$$



(a) k-means (b) Nyström (c) LSC-K (d) LSSC (ours)

Figure 1: Clustering results on the synthetic dataset with 30% uniform noise

Hence, to find the p ($p < k$) smallest eigenvectors of $L = I - W$, we first find the p largest eigenvectors $U_p \in R^{k \times p}$ of $\hat{Z}^T \hat{Z}$ (the eigenvalues store in $\Sigma_p^2 = \text{diag}(\sigma_1^2, \cdots, \sigma_p^2)$) and then compute the p largest eigenvectors V_p of W as:

$$V_p = \hat{Z} U_p \Sigma_p^{-1} \tag{14}$$

which can then be used in Eq. (8). Since both finding V_p (including k-means) and solving min$Q(H_{.i})$ have a linear time and space complexity with respect to n ($p, k, r \ll n$), our algorithm is scalable to large-scale data.

Algorithm 1 Large-Scale Sparse Clustering (LSSC)

```
**Input:** the dataset $X$, the parameters: $l, \lambda, k, r, p$;
**Output:** the predicted labels;
1. Perform $k$-means clustering (with $l$ clusters) on the dataset $X$ to obtain the initial indicator matrix $C$;
2. Produce $k$ clustering centers ($k > l$) using $k$-means clustering on the dataset $X$;
3. Construct the weight matrix $W$ of the graph $\mathcal{G}$ over the dataset $X$ according to Eq. (11);
4. Compute the $p$ largest eigenvectors of $W$ denoted by $V_p$ according to Eq. (14);
5. Solve the problem $H_p^* = \arg\min_{H_p} Q(H_p)$ using the modified FISTA;
6. Compute the probabilistic clustering matrix $Y$ as: $Y^* = V_p H_p^*$;
7. Derive the predicted labels from $Y^* = [y_{ij}^*]_{n \times l}$. Each data point $\mathbf{x_i}$ is divided into cluster $\arg\max_j y_{ij}^*$.
```

The complete large-scale sparse clustering (LSSC) algorithm is outlined in Algorithm 1. Here, we adopt the Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) [Beck and Teboulle, 2009] to solve the generalized sparse coding problem $\min_{H_{.i}} Q(H_{.i})$, given that its implementation mainly involves lightweight operations such as vector operations and matrix-vector multiplications. To adjust FISTA for large-scale sparse clustering, we only need to modify the soft-thresholding function as:

$$\text{soft}(H_{ki}, \frac{\lambda \Sigma_{kk}^{\frac{1}{2}}}{\|V_p\|_s^2}) = \text{sign}(H_{ki}) \max\{|H_{ki}| - \frac{\lambda \Sigma_{kk}^{\frac{1}{2}}}{\|V_p\|_s^2}, 0\}$$

where $\|V_p\|_s$ represents the spectral norm of the matrix V_p. For large-scale problems, it is often computationally expensive to directly compute the Lipschitz constant $\|V_p\|_s^2$. In practice, it can be efficiently estimated by a backtracking line-search strategy [Beck and Teboulle, 2009].

3 Experiments on Synthetic Data

We first evaluate the effectiveness of the proposed LSSC algorithm on synthetic data.

3.1 Compared Algorithms

To demonstrate the noise-robustness of our LSSC algorithm, we compare it with two other state-of-the-art large-scale clustering methods. The details of these two large-scale clustering methods are given below.

- **Nyström** [Chen *et al.*, 2011]: a parallel spectral clustering algorithm developed based on Nyström approximation. The code is available online[1] and we choose the Matlab version with orthogonalization.
- **LSC-K** [Chen and Cai, 2011]: landmark-based spectral clustering using k-means for landmark-selection[2].

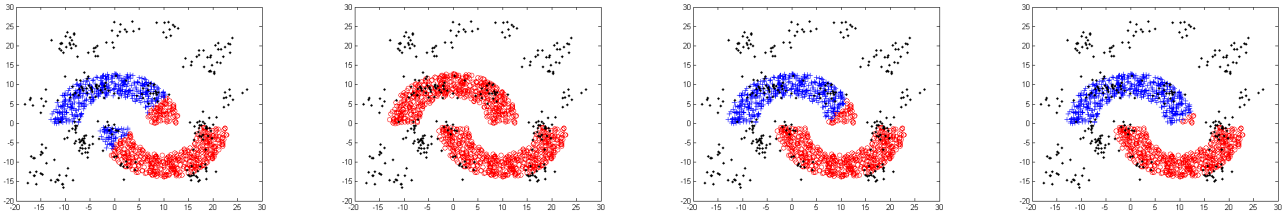
3.2 Clustering Results

We first conduct a group of experiments on a synthetic dataset to qualitatively evaluate the robustness of our LSSC algorithm. Specifically, we generate a two-moon dataset (see Figures 1 and 2), where each cluster comprises of 500 data points. On this noise-free dataset, all of the three large-scale clustering algorithms are shown to successfully detect the correct cluster structures. Furthermore, we add 30% uniform noise and Gaussian noise to the original dataset, and the clustering results for the obtained two noisy datasets are illustrated in Figures 1 and 2, respectively. To show the effectiveness of the second step of our LSSC algorithm, we also report the initial results of the first step (i.e. k-means in Figures 1 and 2) as a baseline. Here, it should be noted that the evaluation of the method used in this paper is what the workers in this field do routinely [Li *et al.*, 2007; Ben-David and Haghtalab, 2014].

From these two figures, we make the following observations: 1) our LSSC algorithm generally detects the accurate cluster structures in spite of the unstructured parts of the inputs; 2) LSC-K is affected by the noise and thus parts of the data points are divided into the wrong clusters; 3) Nyström completely fails in detecting the cluster structures of the noisy

[1] http://alumni.cs.ucsb.edu/wychen/

[2] http://www.cad.zju.edu.cn/home/dengcai/



(a) k-means (b) Nyström (c) LSC-K (d) LSSC (ours)

Figure 2: Clustering results on the synthetic dataset with 30% Gaussian noise

Table 1: Statistics of the two large-scale datasets.

Data Set	Samples	Dimensions	Classes
MNIST	70,000	784	10
Covtype	581,012	54	7

data. These qualitative results demonstrate the robustness of our LSSC algorithm against different types of noise. In addition, as compared to the results of the first step (i.e. k-means in Figures 1 and 2), the improvements achieved by our LSSC algorithm show that the second step of our LSSC algorithm does have the ability to rectify the wrong results induced by the initial k-means clustering.

4 Experiments on Real-World Data

We further evaluate our LSSC algorithm on two real-world datasets from the Yann LeCun's homepage[3] and the UCI repository[4]. Their statistical characteristics are listed in Table 1, and below is a brief description of each dataset:

- **MNIST**: a dataset of handwritten digits, and each digit is represented using 784 features.
- **Covtype**: a dataset to predict forest cover type from cartographic variables only, originally with 54 features.

4.1 Evaluation Metrics

The clustering results are evaluated by comparing the obtained clustering label of each data point with its ground-truth label. We use two standard metrics, accuracy (Accu) [Cai *et al.*, 2005] and purity [Ding *et al.*, 2006], to measure the clustering performance. Given a data point $\mathbf{x}_i$, let r_i and s_i be its obtained clustering label and ground-truth label, respectively. The Accu is defined as:

$$\text{Accu} = \frac{\sum_{i=1}^{n} \delta(s_i, \text{map}(r_i))}{n} \tag{15}$$

where $\text{map}(\cdot)$ denotes the best mapping between the obtained clustering labels and ground-truth labels of data points. Moreover, the purity is defined as:

$$\text{Purity} = \frac{1}{n} \sum_{k=1}^{l} \max_{1 \le j \le l} n_k^j \tag{16}$$

Table 2: Clustering results measured by Accu/Purity (%) for the compared methods.

Data set	Metrics	k-means	Nyström	LSC-K	LSSC (ours)
MNIST	Accu	55.20	53.55	61.56	**65.88**
	Purity	60.44	58.78	67.08	**71.98**
Covtype	Accu	23.05	21.60	22.13	**26.47**
	Purity	49.05	49.70	49.16	**49.76**

where n_k^j is the number of data points in the cluster k that belong to the ground-truth cluster j.

To evaluate the robustness of clustering algorithms, we also adopt δ-robust [Ackerman *et al.*, 2013; Ben-David and Haghtalab, 2014] as one of our measures in this paper. Note that other measures (e.g. accuracy) can not directly reflect the degree of the impact of noise on a clustering algorithm, but δ-robust can intuitively show it.

For a clustering C of the dataset X and data points $x, y \in X$, we write $x \sim_C y$ if x and y belong to the same cluster in C, and $x \nsim_C y$ otherwise.

Definition 1 *(Hamming distance). Given clusterings C and C' of the same dataset X, the Hamming distance between clusterings C and C' is*

$$\Delta(C, C') = |\{\{x, y\} \subset X | (x \sim_C y) \bigoplus (x \sim_{C'} y)\}| / \binom{|X|}{2}$$

where $\bigoplus$ denotes the logical XOR operation.

Given two datasets $X, Z \subseteq E$ with $X \subseteq Z$ and a clustering C of Z, we use $C|X$ to denote the restriction of C to X. If $C = \{C_1, \cdots, C_k\}$, we thus have $C|X = \{C_1 \cap X, \cdots, C_k \cap X\}$.

We further give the definition of δ-robust. Given a dataset X and a (typically large) subset Y, we use $O = X \setminus Y$ to denote a set of noisy data. We thus claim that Y is robust to the noisy dataset O relative to a clustering algorithm $\mathcal{A}$, if Y is clustered similarly with and without the data points in O.

Definition 2 *(δ-robust). Given datasets X, Y and O, where $X = Y \cup O$, Y is δ-robust to O with respect to a clustering algorithm $\mathcal{A}$, if*

$$\Delta(\mathcal{A}(X)|Y, \mathcal{A}(Y)) \le \delta, \tag{17}$$

where $\mathcal{A}(X)$ denotes a clustering of X obtained by the clustering algorithm $\mathcal{A}$.

Equivalently, $\Delta(C, C') = 1 - \text{RI}(C, C')$, where RI is the Rand index [Rand, 1971]. Δ satisfies the triangle inequality.

[3] http://yann.lecun.com/exdb/mnist/

[4] http://archive.ics.uci.edu/ml

Table 3: δ-robust for the compared methods with varying noise levels of uniform noise.

Data set	Noise	k-means	Nyström	LSC-K	LSSC (ours)
MNIST	15%	5.38	6.92	9.83	**3.31**
	30%	7.57	7.64	11.13	**5.17**
Covtype	15%	11.93	12.22	11.09	**6.27**
	30%	12.08	14.70	13.44	**8.05**

Table 4: δ-robust for the compared methods with varying noise levels of Gaussian noise.

Data set	Noise	k-means	Nyström	LSC-K	LSSC (ours)
MNIST	15%	6.43	7.29	7.20	**3.38**
	30%	8.65	10.35	12.36	**7.24**
Covtype	15%	10.24	13.10	15.72	**9.65**
	30%	12.19	15.35	16.76	**11.01**

4.2 Experimental Settings

In the experiments, we produce new noisy datasets by adding two types of noise (uniform noise and Gaussian noise) of different levels (i.e. 0%, 15%, and 30%) to the original datasets. We find that our LSSC algorithm is not sensitive to λ in our experiments, and thus fix this parameter at $\lambda = 0.01$ for all the datasets. By considering a tradeoff of running efficiency and effectiveness, we uniformly set $k = 1,000$ and empirically set $r = 4$, $p = 13$ for MNIST and $r = 2$, $p = 9$ for Covtype. For fair comparison, the same parameters are adopted for all the other related methods. In addition, all the methods are implemented in MATLAB R2014a and run on a 3.40 GHz, 32GB RAM Core 2 Duo PC.

4.3 Clustering Results

We first make comparison on the noise-free datasets. The resulting clustering accuracies and purities are shown in Table 2. To verify the effectiveness of the second step of our LSSC algorithm, we also report the initial results of the first step (i.e. k-means) as a baseline. We see that our LSSC algorithm yields the best performance in all cases, which shows that the two-step optimization strategy indeed helps to effectively detect the accurate cluster structures. As compared with the initial results, we observe that the clustering results can be significantly improved by the second step of our LSSC algorithm. This means that the L_1-optimization does suppress the negative effect of the complicated manifold structure hidden among the large-scale datasets.

We further evaluate the robustness of each clustering method by adding two types of noise (uniform noise and Gaussian noise) of different levels (15%, and 30%). The comparison results are reported in Tables 3 and 4, respectively. The immediate observation is that our LSSC algorithm outperforms all of its competitors in all cases according to δ-robust. The reason is that the L_1-optimization used in the second step can help to find a smooth and sparse solution and thus effectively suppress the negative effect of the noise. In particular, as compared to the most closely related method LSC-K, our LSSC algorithm is shown to achieve significant gains in clustering over large-scale noisy data.

Table 5: The clustering results in terms of both accuracy and running time over the MNIST dataset. The running time of k-means clustering to generate k centers (k = 1,000) for LSC-K and LSSC is 43.70 seconds.

Method	Accu (%)	Running time (sec.)
k-means	55.20	25.16
Nyström	53.55	23.30
LSC-K	61.56	43.70+5.85
LSSC (ours)	**65.88**	43.70+25.16+10.34

Table 6: The clustering results in terms of both accuracy and running time over the Covtype dataset. The running time of k-means clustering to generate k centers (k = 1,000) for LSC-K and LSSC is 68.95 seconds.

Method	Accu (%)	Running time (sec.)
k-means	23.05	44.31
Nyström	21.60	74.43
LSC-K	22.13	68.95+17.04
LSSC (ours)	**26.47**	68.95+44.31+16.68

Finally, the comparison results in terms of both accuracy and running time over the two noise-free datasets are shown in Tables 5 and 6. Although the running time of our LSSC algorithm is more than that of Nyström and LSC-K, it is shown to achieve obvious gains in terms of accuracy. By overall consideration, our LSSC algorithm is preferred in practice. Moreover, excluding the running time taken by k-means clustering to find clustering centers and the first step to obtain the initial clustering results, the L_1-optimization used in the second step of our LSSC algorithm itself is considered to run very efficiently over such large-scale datasets.

5 Conclusion

In this paper, we have investigated the challenging problem of clustering over large-scale noisy data. We have proposed a large-scale sparse clustering (LSSC) algorithm based on a two-step optimization strategy: 1) k-means clustering over the large-scale data to obtain the initial clustering results; 2) clustering refinement over the initial results by developing a spare coding algorithm. To guarantee the scalability of the second step for large-scale data, we have speeded up the sparse coding algorithm using the nonlinear approximation and dimension reduction techniques. Experimental results show the promising performance of our LSSC algorithm.

Acknowledgments

This work was partially supported by National Natural Science Foundation of China (61573363 and 61573026), 973 Program of China (2014CB340403 and 2015CB352502), the Fundamental Research Funds for the Central Universities and the Research Funds of Renmin University of China (15XNLQ01), and IBM Global SUR Award Program.

References

[Ackerman *et al.*, 2013] M. Ackerman, S. Ben-David, D. Loker, and S. Sabato. Clustering oligarchies. In *AISTATS*, pages 66–74, 2013.

[Aggarwal *et al.*, 2003] C. Aggarwal, J. Han, J. Wang, and P. Yu. A framework for clustering evolving data streams. In *VLDB*, pages 81–92, 2003.

[Beck and Teboulle, 2009] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[Ben-David and Haghtalab, 2014] S. Ben-David and N. Haghtalab. Clustering in the presence of background noise. In *ICML*, pages 280–288, 2014.

[Cai *et al.*, 2005] D. Cai, X. He, and J. Han. Document clustering using locality preserving indexing. *IEEE Trans. Knowledge and Data Engineering*, 17(12):1624–1637, 2005.

[Chen and Cai, 2011] X. Chen and D. Cai. Large scale spectral clustering with landmark-based representation. In *AAAI*, pages 313–318, 2011.

[Chen *et al.*, 2011] W. Chen, Y. Song, H. Bai, C. Lin, and E. Chang. Parallel spectral clustering in distributed systems. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 33(3):568–586, 2011.

[Cheng *et al.*, 2010] B. Cheng, J. Yang, S. Yan, Y. Fu, and T. Huang. Learning with l_1-graph for image analysis. *IEEE Trans. Image Processing*, 19(4):858–866, 2010.

[Chitta *et al.*, 2011] R. Chitta, R. Jin, T. Havens, and A. Jain. Approximate kernel k-means: Solution to large scale kernel clustering. In *SIGKDD*, pages 895–903, 2011.

[Cuesta-Albertos *et al.*, 1997] J. Cuesta-Albertos, A. Gordaliza, and C. Matrán. Trimmed k-means: An attempt to robustify quantizers. *The Annals of Statistics*, 25(2):553–576, 1997.

[Ding *et al.*, 2006] C. Ding, T. Li, W. Peng, and H. Park. Orthogonal nonnegative matrix t-factorizations for clustering. In *SIGKDD*, pages 126–135, 2006.

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

[Filippone *et al.*, 2008] M. Filippone, F. Camastra, F. Masulli, and S. Rovetta. A survey of kernel and spectral methods for clustering. *Pattern recognition*, 41(1):176–190, 2008.

[García-Escudero and Gordaliza, 1999] L. García-Escudero and A. Gordaliza. Robustness properties of k-means and trimmed k-means. *Journal of the American Statistical Association*, 94:956–969, 1999.

[García-Escudero *et al.*, 2008] L. García-Escudero, A. Gordaliza, C. Matrán, and A. Mayo-Iscar. A general trimming approach to robust cluster analysis. *The Annals of Statistics*, 36(3):1324–1345, 2008.

[Guha *et al.*, 2001] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. Cure: an efficient clustering algorithm for large databases. *Information Systems*, 26(1):35–58, 2001.

[Härdle, 1992] W. Härdle. Applied nonparametric regression. *Cambridge Books*, 1992.

[Lee *et al.*, 2006] H. Lee, A. Battle, R. Raina, and A. Ng. Efficient sparse coding algorithms. In *Advances in Neural Information Processing Systems*, pages 801–808, 2006.

[Li *et al.*, 2007] Z. Li, J. Liu, S. Chen, and X. Tang. Noise robust spectral clustering. In *ICCV*, pages 1–8, 2007.

[Li *et al.*, 2015] Y. Li, F. Nie, H. Huang, and J. Huang. Large-scale multi-view spectral clustering via bipartite graph. In *AAAI*, pages 2750–2756, 2015.

[Lu and Peng, 2011] Z. Lu and Y. Peng. Latent semantic learning by efficient sparse coding with hypergraph regularization. In *AAAI*, pages 411–416, 2011.

[Lu and Peng, 2013] Z. Lu and Y. Peng. Exhaustive and efficient constraint propagation: A graph-based learning approach and its applications. *International Journal of Computer Vision*, 103(3):306–325, 2013.

[Ng *et al.*, 2002] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems*, pages 849–856, 2002.

[Olshausen and Field, 1997] B. Olshausen and D. Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision Research*, 37(23):3311–3325, 1997.

[Ramirez *et al.*, 2010] I. Ramirez, P. Sprechmann, and G. Sapiro. Classification and clustering via dictionary learning with structured incoherence and shared features. In *CVPR*, pages 3501–3508, 2010.

[Rand, 1971] W. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66:846–850, 1971.

[Wang and Zhang, 2008] F. Wang and C. Zhang. Label propagation through linear neighborhoods. *IEEE Trans. Knowledge and Data Engineering*, 20(1):55–67, 2008.

[Wang *et al.*, 2011] H. Wang, F. Nie, H. Huang, and F. Makedon. Fast nonnegative matrix tri-factorization for large-scale data co-clustering. In *IJCAI*, pages 1553–1558, 2011.

[Wang *et al.*, 2015] Z. Wang, Y. Yang, S. Chang, J. Li, S. Fong, and T.S. Huang. A joint optimization framework of sparse coding and discriminative clustering. In *IJCAI*, pages 3932–3938, 2015.

[Wright *et al.*, 2009] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 31(2):210–227, 2009.

[Zhang and Rudnicky, 2002] R. Zhang and A. Rudnicky. A large scale clustering scheme for kernel k-means. In *ICPR*, pages 289–292, 2002.