Large Scale Spectral Clustering with Landmark-Based Representation

Xinlei Chen **Deng Cai**[*]
State Key Lab of CAD&CG, College of Computer Science,
Zhejiang University, China
endernewton@gmail.com, dengcai@cad.zju.edu.cn

Abstract

Spectral clustering is one of the most popular clustering approaches. Despite its good performance, it is limited in its applicability to large-scale problems due to its high computational complexity. Recently, many approaches have been proposed to accelerate the spectral clustering. Unfortunately, these methods usually sacrifice quite a lot information of the original data, thus result in a degradation of performance. In this paper, we propose a novel approach, called *Landmark-based Spectral Clustering* (LSC), for large scale clustering problems. Specifically, we select p ($\ll n$) representative data points as the landmarks and represent the original data points as the linear combinations of these landmarks. The spectral embedding of the data can then be efficiently computed with the landmark-based representation. The proposed algorithm scales linearly with the problem size. Extensive experiments show the effectiveness and efficiency of our approach comparing to the state-of-the-art methods.

Introduction

Clustering is one of the fundamental problems in data mining, pattern recognition and many other research fields. A series of methods have been proposed over the past decades (Jain, Murty, and Flynn 1999). Among them, *spectral clustering*, a class of methods which is based on eigen-decomposition of matrices, often yields more superior experimental performance comparing to other algorithms (Shi and Malik 2000). While many clustering algorithms are based on Euclidean geometry and consequently place limitations on the shape of the clusters, spectral clustering can adapt to a wider range of geometries and detect non-convex patterns and linearly non-separable clusters (Ng, Jordan, and Weiss 2001; Filippone et al. 2008).

Despite its good performance, spectral clustering is limited in its applicability to large-scale problems due to its high computational complexity. The general spectral clustering method needs to construct an adjacency matrix and calculate the eigen-decomposition of the corresponding Laplacian matrix (Chung 1997). Both of these two steps are computational expensive. For a data set consisting of n data points, the above two steps will have time complexities of $O(n^2)$ and $O(n^3)$, which is an unbearable burden for large-scale applications.

In recent years, much effort has been devoted for accelerating the spectral clustering algorithm. A natural option is finding the methods to reduce the computational cost of the eigen-decomposition of the graph Laplacian. (Fowlkes et al. 2004) adopted the classical Nyström method for efficiently computing an approximate solution of the eigenproblem. Another option is to perform a reduction in the data size beforehand. (Shinnou and Sasaki 2008) replaced the original data set with a relatively small number of data points, and the follow-up operations are performed on the adjacency matrix corresponding to the smaller set. Based on a similar idea, (Yan, Huang, and Jordan 2009) provided a general framework for fast approximate spectral clustering. (Sakai and Imiya 2009) used another variant which is based on random projection and sampling. (Chen et al. 2006; Liu et al. 2007) introduced a sequential reduction algorithm based on the observation that some data points converge to their true embedding quickly, so that an early stop strategy will speed up decomposition. However, their idea can only tackle binary clustering problems and should resort to a hierarchical scheme for multi-way clustering.

Inspired by the recent progress on sparse coding (Lee et al. 2006) and scalable semi-supervised learning (Liu, He, and Chang 2010), we propose a scalable spectral clustering method termed *Landmark-based Spectral Clustering* (LSC) in this paper. Specifically, LSC selects p ($\ll n$) representative data points as the landmarks and represent the remaining data points as the linear combinations of these landmarks. The spectral embedding of the data can then be efficiently computed with the landmark-based representation. The proposed algorithm scales linearly with the problem size. Extensive experiments show the effectiveness and efficiency of our approach comparing to the state-of-the-art methods.

The rest of the paper is organized as follows: in Section 2, we provide a brief review of several popular methods which are designed for speeding up the spectral clustering. Our Landmark-based Spectral Clustering method is introduced in Section 3. The experimental results are presented in Section 4. Finally, we provide the concluding remarks in Section 5.

[*]Corresponding author

Related Work

Given a set of data points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \in \mathbb{R}^m$, spectral clustering first constructs an undirected graph $\mathcal{G} = (V, E)$ represented by its adjacency matrix $W = (w_{ij})_{i,j=1}^n$, where $w_{ij} \geq 0$ denotes the similarity (affinity) between $\mathbf{x}_i$ and $\mathbf{x}_j$. The degree matrix D is a diagonal matrix whose entries are column (or row, since W is symmetric) sums of W, $D_{ii} = \sum_j W_{ji}$. Let $L = D - W$, which is called graph Laplacian (Chung 1997). Spectral clustering then use the top k eigenvectors of L (or, the normalized Laplacian $D^{-1/2}LD^{-1/2}$) corresponding to the k smallest eigenvalues[1] as the low dimensional (with dimensionality k) representations of the original data. Finally, the traditional k-means method (Hartigan and Wong 1979) is applied to obtain the clusters. Due to the high complexity of the graph construction ($O(n^2)$) and the eigen-decomposition ($O(n^3)$), it is not easy to apply spectral clustering on large-scale data sets.

A natural way to handle this scalability issue is using the sampling technique. The basic idea is using pre-processing to reduce the data size. (Yan, Huang, and Jordan 2009) proposed the *k-means-based approximate spectral clustering* (KASP) method. It firstly performs k-means on the data set with a large cluster number p. Then, the traditional spectral clustering is applied on the p cluster centers. The data point is assigned to the cluster as its nearest center.

(Shinnou and Sasaki 2008) adopted a slightly different way to reduce the data size. Their approach firstly applies k-means on the data set with a large cluster number p. It then removes those data points which are close to the centers (with pre-defined distance threshold). The centers are called committees in their algorithm. The traditional spectral clustering is applied on the remaining data points plus the cluster centers. Those removed data points are assigned to the cluster as their nearest centers. In the experiments, we named this approach *Committees-based Spectral Clustering* (CSC).

Another way to handle the scalability issue of spectral clustering is reducing the computational cost of the eigen-decomposition step. (Fowlkes et al. 2004) applied the Nyström method to accelerate the eigen-decomposition. Given an $n \times n$ matrix, Nyström method computes the eigenvectors of a $p \times p$ ($p \ll n$) sub-matrix (randomly sampled from the original matrix). The calculated eigenvectors are used to estimate an approximation of the eigenvectors of the original matrix.

All these approaches used the sampling technique. Some key data points are selected to represent the other data points. In reality, this idea is very effective. However, a lot of information of the detailed structure of the data is lost in the sampling step.

Landmark-based Spectral Clustering

In this section, we introduce our *Landmark-based Spectral Clustering* (LSC) for large scale spectral clustering. The basic idea of our approach is designing an efficient way for graph construction and Laplacian matrix eigen-decomposition. Specifically, we try to design the affinity matrix which has the property as follows:

$$W = \hat{Z}^T \hat{Z}, \tag{1}$$

where $\hat{Z} \in \mathbb{R}^{p \times n}$ and $p \ll n$. Thus, we can build the graph in $O(np)$ and compute eigenvectors of the graph Laplacian in $O(p^3 + p^2 n)$. Our approach is motivated from the recent progress on sparse coding (Lee et al. 2006) and scalable semi-supervised learning (Liu, He, and Chang 2010).

Landmark-based Sparse Coding

Sparse coding is a matrix factorization technique which tries to ”compress” the data by finding a set of *basis* vectors and the *representation* with respect to the basis for each data point. Let $X = [\mathbf{x}_1, \cdots, \mathbf{x}_n] \in \mathbb{R}^{m \times n}$ be the data matrix, matrix factorization can be mathematically defined as finding two matrices $U \in \mathbb{R}^{m \times p}$ and $Z \in \mathbb{R}^{p \times n}$ whose product can best approximate X:

$$X \approx UZ.$$

Each column of U can be regarded as a basis vector which captures the higher-level features in the data and each column of Z is the p-dimensional representation of the original inputs with respect to the new basis. A common way to measure the approximation is by Frobenius norm of a matrix $\|\cdot\|$. Thus, the matrix factorization can be defined as the optimization problem as follows:

$$\min_{U,Z} \|X - UZ\|^2 \tag{2}$$

Since each basis vector (column vector of U) can be regarded as a concept, a dense matrix Z indicates that each data point is a combination of *all* the concepts. This is contrary to our common knowledge since most of the data points only include several semantic concepts. Sparse Coding (SC) (Lee et al. 2006; Olshausen and Field 1997) is a recently popular matrix factorization method trying to solve this issue. Sparse coding adds the sparse constraint on Z, more specifically, on each column of A, in the optimization problem (2). In this way, SC can learn a sparse representation. SC has several advantages for data representation. First, it yields sparse representations such that each data point is represented as a linear combination of a small number of basis vectors. Thus, the data points can be interpreted in a more elegant way. Second, sparse representations naturally make for an indexing scheme that would allow quick retrieval. Third, the sparse representation can be over-complete, which offers a wide range of generating elements. Potentially, the wide range allows more flexibility in signal representation and more effectiveness at tasks like signal extraction and data compression (Olshausen and Field 1997).

[1] It is easy to check that the eigenvectors of $D^{-1/2}LD^{-1/2}$ corresponding to the smallest eigenvalues are the same as the eigenvectors of $D^{-1/2}WD^{-1/2}$ corresponding to the largest eigenvalues (Ng, Jordan, and Weiss 2001).

However, solving the optimization problem (2) with sparse constraint is very time consuming. Most of the existing approaches compute U and Z iteratively. Apparently, these approaches cannot be used for spectral clustering.

The basis vectors (column vectors of U) have the same dimensionality with the original data points. We can treat the basis vectors as the landmark points of the data set. The most efficient way to select landmark points from a data set is random sampling. Besides random selection, several methods were proposed for landmark points selection (Kumar, Mohri, and Talwalkar 2009; Boutsidis, Mahoney, and Drineas 2009). For instance, we can apply the k-means algorithm to first cluster all the data points and then use the cluster centers as the landmark points. But, many of these methods are computationally expensive and do not scale to large data sets. We therefore focus on the random selection method, although the comparison between random selection and k-means based landmark selection is presented in our empirical study.

Suppose we already have the landmark matrix U, we can solve the optimization problem (2) to compute the representation matrix Z. By fixing U, the optimization problem becomes a constraint (sparsity constraints) linear regression problem. There are many algorithms (Liu, He, and Chang 2010; Efron et al. 2004) which can solve this problem. However, these optimization approaches are still time consuming. In our approach, we simply use Nadaraya-Watson kernel regression (Härdle 1992) to compute the representation matrix Z.

For any data point $\mathbf{x}_i$, we find its approximation $\hat{\mathbf{x}}_i$ by

$$\hat{\mathbf{x}}_i = \sum_{j=1}^{p} z_{ji}\mathbf{u}_j \tag{3}$$

where $\mathbf{u}_j$ is j-th column vector of U and z_{ji} is ji-th element of Z. A natural assumption here is that z_{ji} should be larger if $\mathbf{x}_i$ is closer to $\mathbf{u}_j$. We can emphasize this assumption by setting the z_{ji} to zero as $\mathbf{u}_j$ is not among the r $(\leq p)$ nearest neighbors of $\mathbf{x}_i$. This restriction naturally leads to a sparse representation matrix Z. Let $U_{\langle i\rangle} \in \mathbb{R}^{m\times r}$ denote a sub-matrix of U composed of r nearest landmarks of $\mathbf{x}_i$. We compute z_{ji} as

$$z_{ji} = \frac{K_h(\mathbf{x}_i, \mathbf{u}_j)}{\sum_{j'\in U_{\langle i\rangle}} K_h(\mathbf{x}_i, \mathbf{u}_{j'})} \quad j \in U_{\langle i\rangle}. \tag{4}$$

where $K_h(\cdot)$ is a kernel function with a bandwidths h. The Gaussian kernel $K_h(\mathbf{x}_i, \mathbf{u}_j) = \exp(-\|\mathbf{x}_i - \mathbf{u}_j\|^2/2h^2)$ is one of the most commonly used.

Spectral Analysis on Landmark-based Graph

We have the landmark-based sparse representation $Z \in \mathbb{R}^{p\times n}$ now and we simply compute the graph matrix as

$$W = \hat{Z}^T\hat{Z}, \tag{5}$$

which can have a very efficient eigen-decomposition. In the algorithm, we choose $\hat{Z} = D^{-1/2}Z$ where D is the row-sum of Z. Note that in the previous section, each column of Z sums up to 1 and thus the degree matrix of W is I, i.e. the graph is automatically normalized.

Algorithm 1 Landmark-based Spectral Clustering

```
Input:
    $n$ data points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \in \mathbb{R}^m$;
    Cluster number $k$ ;
Output:
    $k$ clusters;
    Produce $p$ landmark points using $k$-means or random selection;
    Construct a sparse affinity matrix $Z \in \mathbb{R}^{p\times n}$ between data points and landmark points, with the affinity calculated according to Eq. (4);
    Compute the first $k$ eigenvectors of $ZZ^T$, denoted by $A = [\mathbf{a}_1, \cdots, \mathbf{a}_k]$;
    Compute $B = [\mathbf{b}_1, \cdots, \mathbf{b}_k]$ according to Eq. (7);
    Each row of $B$ is a data point and apply $k$-means to get the clusters.
```

Table 1: Time complexity of accelerating methods

Method	Pre-process	Construction	Decomposition
KASP	$O(tpnm)$	$O(p^2m)$	$O(p^3)$
CSC	$O(tpnm)$	$O(p^2m)$	$O(p^3)$
Nyström	/	$O(pnm)$	$O(p^3 + pn)$
LSC-R	/	$O(pnm)$	$O(p^3 + p^2n)$
LSC-K	$O(tpnm)$	$O(pnm)$	$O(p^3 + p^2n)$

* n: # of points; m: # of features; p: # of landmarks / centers / sampled points; t: # of iterations in k-means.

** The final clustering is $O(tnk^2)$ for each algorithm, with k denote the number of clusters.

Let the Singular Value Decomposition (SVD) of $\hat{Z}$ is as follows:

$$\hat{Z} = A\Sigma B^T, \tag{6}$$

where $\Sigma = diag(\sigma_1, \cdots, \sigma_p)$ and $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p \geq 0$ are the singular values of $\hat{Z}$, $A = [\mathbf{a}_1, \cdots, \mathbf{a}_p] \in \mathbb{R}^{p\times p}$ and $\mathbf{a}_i$'s are called left singular vectors, $B = [\mathbf{b}_1, \cdots, \mathbf{b}_p] \in \mathbb{R}^{n\times p}$ and $\mathbf{b}_i$'s are called right singular vectors.

It is easy to check that $B = [\mathbf{b}_1, \cdots, \mathbf{b}_p] \in \mathbb{R}^{n\times p}$ are the eigenvectors of matrix $W = \hat{Z}^T\hat{Z}$; $A = [\mathbf{a}_1, \cdots, \mathbf{a}_p] \in \mathbb{R}^{p\times p}$ are the eigenvectors of matrix $\hat{Z}\hat{Z}^T$; and σ_i^2 are the eigenvalues. Since the size of matrix $\hat{Z}\hat{Z}^T$ is $p \times p$, we can compute A within $O(p^3)$ time. B can then be computed as

$$B^T = \Sigma^{-1}A^T\hat{Z} \tag{7}$$

The overall time is $O(p^3 + p^2n)$, which is a significant reduction from $O(n^3)$ considering $p \ll n$.

Computational Complexity Analysis

Suppose we have n data points with dimensionality m and we use p landmarks, we need $O(pnm)$ to construct the graph and $O(p^3 + p^2n)$ to compute the eigenvectors. If we use k-means to select the landmarks, we need additional $O(tpnm)$ time, where t is the number of iterations in k-means. We summarize our algorithm in Algorithm 1 and the computational complexity in Table 1. For the sake of comparison,

Table 2: Data sets used in our experiments

Data set	# of instances	# of features	# of classes
MNIST	70000	784	10
LetterRec	20000	16	26
PenDigits	10992	16	10
Seismic	98528	50	3
Covtype	581012	54	7

Table 1 also lists several other popular accelerating spectral clustering methods. We use LSC-R to denote our method with random landmark selection and LSC-K to denote our method with *k*-means landmark selection.

Experiments

In this section, several experiments were conducted to demonstrate the effectiveness of the proposed Landmark-based Spectral Clustering (LSC).

Data Sets

We have conducted experiments on five real-world large data sets downloaded from the UCI machine learning repository[2] and the LibSVM data sets page[3]. An brief description of the data sets is listed below (see Table 2 for some important statistics):

MNIST A data set of handwritten digits from Yann LeCun's page [4]. Each image is represented as a 784 dimensional vector.

LetterRec A data set of 26 capital letters in the English alphabet. 16 character image features are selected.

PenDigits Also a handwritten digit data set of 250 samples from 44 writers, but it uses the sampled coordination information instead.

Seismic A data set initially built for the task of classifying the types of moving vehicles in a distributed, wireless sensor network (Duarte and Hu 2004).

Covtype A data set to predict forest cover type from cartographic variables.

Each data point is normalized to have the unit norm and no other preprocessing step is applied.

Evaluation Metric

The clustering result is evaluated by comparing the obtained label of each sample with the label provided by the data set. We use the accuracy (AC) (Cai et al. 2005) to measure the clustering performance. Given a data point $\mathbf{x}_i$, let r_i and s_i be the obtained cluster label and the label provided by the corpus, respectively. The AC is defined as follows:

$$\text{AC} = \frac{\sum_{i=1}^{N} \delta(s_i, map(r_i))}{N}$$

where N is the total number of samples and $\delta(x, y)$ is the delta function that equals 1 if $x = y$ and equals 0 otherwise, and map(r_i) is the permutation mapping function that maps each cluster label r_i to the equivalent label from the data corpus. The best mapping can be found by using the Kuhn-Munkres algorithm (Lovasz and Plummer 1986).

We also record the running time of each method. All the codes in the experiments are implemented in MATLAB R2010a and run on a Linux machine with 2.66 GHz CPU, 4GB main memory.

Compared Algorithms

To demonstrate the effectiveness and efficiency of our proposed *Landmark-based Spectral Clustering*, we compare it with three other state-of-the-art approaches described in Section 2. Following is a list of information concerning experimental settings of each method:

KASP *k*-means-based approximate spectral clustering method proposed in (Yan, Huang, and Jordan 2009). The authors have provided their R code on the website[5]. For fair comparison, we implement a multi-way partition version in MATLAB.

CSC Committees-based Spectral Clustering proposed in (Shinnou and Sasaki 2008).

Nyström There are several variants available for Nyström approximation based spectral clustering, and we choose the Matlab implementation with orthogonalization (Chen et al. 2010), which is available online[6].

To test the effectiveness of the accelerating scheme, we also report the results of the conventional spectral clustering. For our Landmark-based Spectral Clustering, we implemented two versions as follows:

LSC-R Short for Landmark-based Spectral Clustering using random sampling to select landmarks.

LSC-K Short for Landmark-based Spectral Clustering using *k*-means for landmark-selection.

There are two parameters in our LSC approach: the number of landmarks p and the number of nearest landmarks r for a single point.Throughout our experiments, we empirically set $r = 6$ and $p = 500$.

For fair comparison, we use the same clustering result for landmarks (centers) selection in KASP, CSC and LSC-K. We also use the same random selection for Nyström and LSC-R. For each landmark number p (or number of centers, number of selected samples), 20 tests are conducted and the average performance is reported.

Experimental Results

The performance of the five methods along with original spectral clustering on all the five data sets are reported in Table 3 and 4. These results reveal a number of interesting points as follows:

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

[3] http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/

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

[5] http://www.cs.berkeley.edu/~jordan/fasp.html

[6] http://alumni.cs.ucsb.edu/~wychen/

Table 3: Clustering time on the five data sets (s)

Data set	Original	KASP	CSC	Nyström	LSC-R	LSC-K
MNIST	*3654.90*	416.66	439.06	48.88	**35.95**	468.17
LetterRec	*195.63*	66.65	66.93	24.43	**9.63**	61.59
PenDigits	*60.48*	22.15	26.22	11.49	**3.11**	28.58
Seismic	*4328.35*	**16.64**	18.34	38.34	21.73	67.02
Covtype	*181006.17*	360.07	402.14	258.25	**134.71**	615.84

Table 4: Clustering accuracy on the five data sets (%)

Data set	Original	KASP	CSC	Nyström	LSC-R	LSC-K
MNIST	*72.46*	56.51	55.51	53.70	62.66	**67.04**
LetterRec	*31.04*	29.49	27.12	30.11	29.22	**30.33**
PenDigits	*76.55*	72.47	70.78	73.94	79.04	**79.27**
Seismic	*65.23*	63.70	66.76	66.92	67.60	**67.65**
Covtype	*44.24*	22.42	21.65	22.31	24.75	**25.50**

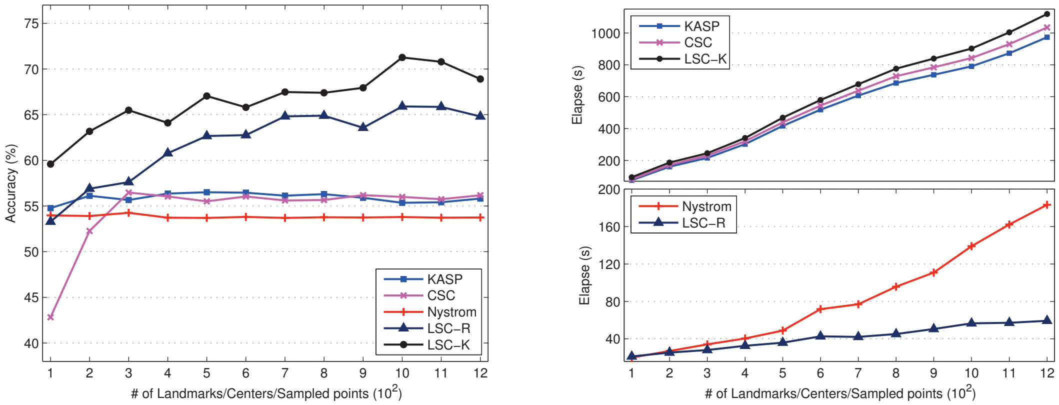

Figure 1: Clustering accuracy and running time VS. # of landmark points on MNIST data set

- Considering the accuracy, LSC-K outperforms all of its competitors on all the data sets. For example, LSC-K achieves a 11% performance gain on MNIST over the second best non-LSC method. It even beats the original spectral clustering algorithm on several data sets. The reason might be the effectiveness of the proposed landmark-based sparse representation. However, running time is its fatal weakness due to the k-means based landmarks selection.

- LSC-R demonstrates an elegant balance between running time and accuracy. It runs much faster than the other four methods while still achieves comparable accuracy with LSC-K. Particularly, on Covtype, it finishes in 135 seconds, which is almost 1500 times faster than the original spectral clustering. Comparing to LSC-K, LSC-R achieves a similar accuracy within 1/9 time on PenDigits. Overall, LSC-R is the best choice among the compared approaches.

- The running time difference between LSC-R and LSC-K shows how the initial k-means performs. It is not surprising that the k-means based landmark selection becomes very slow as either the sample number or the feature number gets large.

Parameters Selection

In order to further examine the behaviors of these methods, we choose the MNIST data set and conducted a thorough study.

All the algorithms have the same parameter: the number of landmarks p (or the number of centers in KASP and CSC, or the number of sampled points in Nyström). Figure 1 shows how the clustering accuracy and running time changes as p varying from 100 to 1200 on MNIST. It can be seen that LSC methods (both LSC-K and LSC-R) can achieve better clustering results as the number of landmarks increases.

Another essential parameter in LSC is the number of nearest landmarks r for a single data point in sparse representation learning. Figure 2 shows how the clustering accuracy and the running time of LSC varies with this parameter. As we can see, LSC is very robust with respect to r. It achieves consistent good performance with the r varying from 3 to 10.

Conclusion

In this paper, we have presented a novel large scale spectral clustering method, called *Landmark-based Spectral Clustering* (LSC). Given a data set with n data points, LSC selects

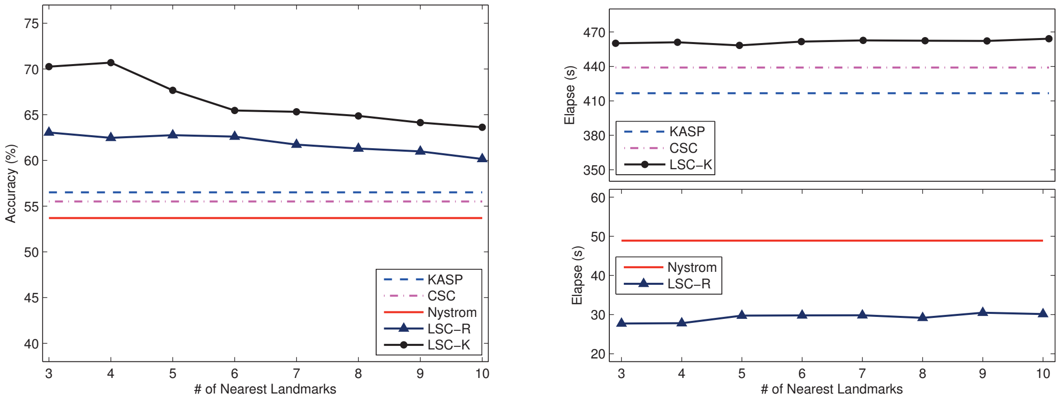

Figure 2: Clustering accuracy and running time VS. # of nearest landmarks on MNIST data set

p ($\ll n$) representative data points as the landmarks and represent the original data points as the linear sparse combinations of these landmarks. The spectral embedding of the data can then be efficiently computed with the landmark-based representation. As a result, LSC scales linearly with the problem size. Extensive experiments on clustering show the effectiveness and efficiency of our approach comparing to the state-of-the-art methods.

Acknowledgments

This work was supported in part by National Natural Science Foundation of China under Grants 60905001, National Basic Research Program of China (973 Program) under Grant 2011CB302206 and Fundamental Research Funds for the Central Universities.

References

Boutsidis, C.; Mahoney, M.; and Drineas, P. 2009. An improved approximation algorithm for the column subset selectionproblem. In *SODA'09*.

Cai, D.; He, X.; Han, J.; and Member, S. 2005. Document clustering using locality preserving indexing. *IEEE Transactions on Knowledge and Data Engineering*.

Chen, B.; Gao, B.; Liu, T.-Y.; Chen, Y.-F.; and Ma, W.-Y. 2006. Fast spectral clustering of data using sequential matrix compression. In *Proceedings of the 17th European Conference on Machine Learning (ECML'06)*.

Chen, W.-Y.; Song, Y.; Bai, H.; Lin, C.-J.; and Chang, E. Y. 2010. Parallel spectral clustering in distributed systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Chung, F. R. K. 1997. *Spectral Graph Theory*, volume 92 of *Regional Conference Series in Mathematics*. AMS.

Duarte, M. F., and Hu, Y. H. 2004. Vehicle classification in distributed sensor networks. *Journal of Parallel and Distributed Computing*.

Efron, B.; Hastie, T.; Johnstone, I.; and Tibshirani, R. 2004. Least angle regression. *Annals of Statistics* 32(2):407–499.

Filippone, M.; Camastra, F.; Masulli, F.; and Rovetta, S. 2008. A survey of kernel and spectral methods for clustering. *Pattern Recognition* 41:176–190.

Fowlkes, C.; Belongie, S.; Chung, F.; and Malik, J. 2004. Spectral grouping using the nyström method. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26:2004.

Härdle, W. 1992. *Applied nonparametric regression*. Cambridge University Press.

Hartigan, J. A., and Wong, M. A. 1979. A k-means clustering algorithm. *JSTOR: Applied Statistics* 28(1):100–108.

Jain, A. K.; Murty, M. N.; and Flynn, P. J. 1999. Data clustering: a review. *ACM Comput. Surv.*

Kumar, S.; Mohri, M.; and Talwalkar, A. 2009. On sampling-based approximate spectral decomposition. In *Proceedings of the 26th International Conference on Machine Learning (ICML'09)*.

Lee, H.; Battle, A.; Raina, R.; ; and Ng, A. Y. 2006. Efficient sparse coding algorithms. In *Advances in Neural Information Processing Systems 19*. MIT Press.

Liu, T.-Y.; Yang, H.-Y.; Zheng, X.; Qin, T.; and Ma, W.-Y. 2007. Fast large-scale spectral clustering by sequential shrinkage optimization. In *Proceedings of the 29th European Conference on IR research (ECIR'07)*.

Liu, W.; He, J.; and Chang, S.-F. 2010. Large graph construction for scalable semi-supervised learning. In *Proceedings of the 27th International Conference on Machine Learning (ICML'10)*.

Lovasz, L., and Plummer, M. 1986. *Matching Theory*. North Holland, Budapest: Akadémiai Kiadó.

Ng, A. Y.; Jordan, M. I.; and Weiss, Y. 2001. On spectral clustering: analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*. MIT Press.

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

Sakai, T., and Imiya, A. 2009. Fast spectral clustering with random projection and sampling. In *Proceedings of the 3rd International Conference on Machine Learning and Data Mining (MLDM'09)*.

Shi, J., and Malik, J. 2000. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22(8):888–905.

Shinnou, H., and Sasaki, M. 2008. Spectral clustering for a large data set by reducing the similarity matrix size. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*.

Yan, D.; Huang, L.; and Jordan, M. I. 2009. Fast approximate spectral clustering. In *Proceedings of the 15th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD'09)*.