Scalable Normalized Cut with Improved Spectral Rotation

Xiaojun Chen[1], Feiping Nie[2]*, Joshua Zhexue Huang[1], Min Yang[3]
[1]College of Computer Science and Software, Shenzhen University, Shenzhen 518060, P.R. China
[2]School of Computer Science and Center for OPTical IMagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University, Xi′an 710072, P. R. China
[3]Tencent AI Lab, Shenzhen, P.R. China
xjchen@szu.edu.cn, feipingnie@gmail.com, zx.huang@szu.edu.cn, min.yang1129@gmail.com

Abstract

Many spectral clustering algorithms have been proposed and successfully applied to many high-dimensional applications. However, there are still two problems that need to be solved: 1) existing methods for obtaining the final clustering assignments may deviate from the true discrete solution, and 2) most of these methods usually have very high computational complexity. In this paper, we propose a Scalable Normalized Cut method for clustering of large scale data. In the new method, an efficient method is used to construct a small representation matrix and then clustering is performed on the representation matrix. In the clustering process, an improved spectral rotation method is proposed to obtain the solution of the final clustering assignments. A series of experimental were conducted on 14 benchmark data sets and the experimental results show the superior performance of the new method.

1 Introduction

Clustering is a hot topic in machine learning and data mining. Over the past decades, many clustering algorithms have been proposed for cluster analysis of high-dimensional data, such as spectral clustering [Von Luxburg, 2007], subspace clustering [Kriegel *et al.*, 2009; Chen *et al.*, 2012], multi-view clustering [Cai *et al.*, 2011; Chen *et al.*, 2013], etc. Among them, spectral clustering is a popular method because it is easy to implement and often shows good clustering performance due to the use of manifold information. Various spectral clustering algorithms have been proposed, such as Ratio Cut [Hagen and Kahng, 1992], k-way Ratio Cut [Samaria and Harter, 1995], Normalized Cut [Ng *et al.*, 2002], Spectral Embedded Clustering [Nie *et al.*, 2011] and MinMax Cut [Nie *et al.*, 2010]. They have been successfully applied to many high-dimensional applications, such as image segmentation [Shi and Malik, 2000; Yu and Shi, 2003], clustering gene expression data [de Souto *et al.*, 2008] and power network decomposing [Sánchez-García *et al.*, 2014].

*Corresponding Author.

Spectral clustering methods usually transform the data into a weighted, undirected graph based on pairwise similarities. To obtain the final discrete clustering assignments, they often perform eigendecomposition first, and then the final clustering assignments can be obtained from eigenvectors by k-means or spectral rotation [Yu and Shi, 2003]. According to the analysis in [Huang *et al.*, 2013], spectral rotation can obtain better clustering result than k-means. However, spectral rotation involves a two-stage process in which an approximate continuous cluster assignment matrix is first computed, and the final discrete solution is a nearby discrete solution obtained from the approximate continuous cluster assignment matrix. A disadvantage of this two-stage process is that the final clustering structures may deviate from the true discrete solution.

Moreover, since both graph construction as well as spectral analysis are time consuming, spectral clustering usually has a time complexity of $O(n^3)$ where n is the number of samples. In recent years, much effort has been devoted for accelerating the spectral clustering. There are mainly two ways to handle the scalability issue of spectral clustering. One way is to reduce the computational cost of the eigendecomposition step [Fowlkes *et al.*, 2010; Li *et al.*, 2010], and another way is to sample the original data and perform clustering on the reduced data [Yan *et al.*, 2009; Shinnou and Sasaki, 2008]. However, these methods are based on sampling, and a lot of information of the data will be lost in the sampling step. Recently, Cai et al. proposed a landmarks-based spectral clustering (LSC) method [Cai and Chen, 2015]. Given a data set with n samples, LSC generates $m \ll n$ representative data points to compute a representation matrix and the eigendecomposition can be performed on the low-size representation matrix. The final discrete clustering result is obtained from eigenvectors by k-means. The overall time of LSC is $O(ndmt + nm^2)$ where t is the number of iterations of k-means for anchor generation, which is significant reduction from $O(n^3)$ considering $m \ll n$. However, how to effectively construct a representation matrix and how to effectively obtain the clustering assignments are still two problems that need to be solved.

In this paper, we propose a Scalable Normalized Cut method (SNC) for large scale data. Given a data set with n samples, we first use k-means to find $m \ll n$ representative data points, a new method to construct a low-size $n \times m$

representation matrix on which the eigendecomposition can be performed. We propose an Improved Spectral Rotation method to obtain the final clustering assignments. SNC has the same computational complexity as LSC for large scale data. The main contributions of our work include:

1. We propose an Improved Spectral Rotation (ISR) to obtain the solution of the final clustering assignments.
2. We propose an efficient method to construct a small representation matrix, which can be used to computed an affinity matrix. We futher prove that the resulting affinity matrix is symmetric and doubly stochastic.
3. Comprehensive experiments on 14 benchmark data sets show the efficiency and effectiveness of the proposed method.

The rest of this paper is organized as follows. Notations and preliminaries are given in Section 2. We review the background and related work in Section 3. The Improved Spectral Rotation (ISR) is given in Section 4 and the Scalable Normalized Cut (SNC) is given in Section 5. We present experimental results and analysis in Section 6. Conclusions and future work are given in Section 7.

2 Notations and Definitions

We summarize the notations and the definition of norms used in this paper. Matrices are written as boldface uppercase letters. Vectors are written as boldface lowercase letters. For matrix $\mathbf{M} = (m_{ij})$, its i-th row is denoted as $\mathbf{m}^i$, and its j-th column is denoted by $\mathbf{m}_j$. The Frobenius norm of the matrix $\mathbf{M} \in \mathcal{R}^{n\times m}$ is defined as $\|\mathbf{M}\|_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{m} m_{ij}^2}$.

3 Background and Related Work

In this section, we introduce the anchor-based similarity matrix construction and spectral rotation.

3.1 Anchor-based Similarity Matrix Construction

To handle the scalability issue of spectral clustering, Liu et al. proposed an anchor-based strategy [Liu *et al.*, 2010], which is also called landmarks-based method [Cai and Chen, 2015]. Given a data set $\mathbf{X} \in \mathcal{R}^{d\times n}$ with n objects $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$, anchor-based strategy first seeks m anchors, where $m \ll n$, and then construct the affinity matrix by calculating the distance between anchors and original samples. There are mainly two methods for anchor generation, i.e., random selection and k-means generation. Since clustering centers have a stronger representation power than random selected data, it is preferred to use k-means for anchor generation [Liu *et al.*, 2010; Cai and Chen, 2015].

After we have m anchors $\mathbf{W} \in \mathcal{R}^{d\times m}$, the next step is to obtain a representation matrix $\mathbf{B}$ such that $\mathbf{X} \approx \mathbf{WB}$. With $\mathbf{B}$, we can obtain an affinity matrix $\mathbf{A}$ as [Liu *et al.*, 2010]

$$\mathbf{A} = \mathbf{B}\Delta^{-1}\mathbf{B}^T \tag{1}$$

where $\Delta \in \mathcal{R}^{m\times m}$ is a diagonal matrix and the j-th entry is defined as $\Delta_{jj} = \sum_{i=1}^{n} b_{ij}$. The most important property of this similarity matrix is that it can be represented as $\mathbf{A} = \mathbf{PP}^T$ where $\mathbf{P} \in \mathcal{R}^{n\times m} = \mathbf{B}\Delta^{-\frac{1}{2}}$.

3.2 Spectral Rotation

In this subsection, we introduce the spectral rotation which is used in Multiclass Spectral Clustering (MSC) [Yu and Shi, 2003]. Given an affinity matrix $\mathbf{A}$, we can compute the corresponding degree matrix $\mathbf{D}_A$, which is a diagonal matrix with the i-th diagonal element as $d_{ii} = \sum_{j=1}^{n} a_{ij}$. The objective function of MSC is

$$\max_{\mathbf{Y}\in Ind, \mathbf{Z}=\mathbf{Y}(\mathbf{Y}^T\mathbf{D}_A\mathbf{Y})^{-\frac{1}{2}}} Tr(\mathbf{Z}^T\mathbf{AZ}) \tag{2}$$

where $\mathbf{Z} \in \mathcal{R}^{n\times c}$ is the scaled partition matrix. It is hard to directly solve problem (2). A well known way is to relax the matrix $\mathbf{Z}$ from the discrete values to the continuous ones, and form the new problem

$$\max_{\mathbf{Z}^T\mathbf{D}_A\mathbf{Z}=\mathbf{I}_c} Tr(\mathbf{Z}^T\mathbf{AZ}) \tag{3}$$

According to Proposition 1 in [Yu and Shi, 2003], the optimal solution of Z is $\{\mathbf{Z}^*\mathbf{R} : \mathbf{R}^T\mathbf{R} = \mathbf{I}_c\}$ where $\mathbf{Z}^* \in \mathcal{R}^{n\times c}$ is the c column vectors of the eigenvectors of $\mathbf{D}_A^{-1}\mathbf{A}$ which correspond to the c biggest eigenvalues.

To obtain the discrete solution $\mathbf{Y}$, we first compute an approximate $\mathbf{Y}^*$ as

$$\mathbf{Y}^* = Diag(\mathbf{Z}^*(\mathbf{Z}^*)^T)^{-\frac{1}{2}}\mathbf{Z}^* \tag{4}$$

Then we can learn suitable $\mathbf{R}$ and $\mathbf{Y}$ such that $\mathbf{Y}^*\mathbf{R}$ is closest to $\mathbf{Y}$ by solving the following problem

$$\min_{\mathbf{Y}\in\mathcal{B}^{n\times c}, \mathbf{R}\in\mathcal{R}^{c\times c}, \mathbf{Y1}_c=\mathbf{1}_n, \mathbf{R}^T\mathbf{R}=\mathbf{I}_c} \|\mathbf{Y} - \mathbf{Y}^*\mathbf{R}\|_F^2 \tag{5}$$

4 Improved Spectral Rotation

In MSC, approximate Y^* is first computed and then a suitable $\mathbf{R}$ is learnt for the final cluster indicator matrix $\mathbf{Y}$. However, the final clustering results may deviate from the true discrete solution since Y^* is an approximate solution. In this paper, we propose a new spectral rotation method to obtain better discrete solution of $\mathbf{Y}$. We first rewrite problem (2) as follows

$$\max_{\mathbf{Y}\in Ind, \mathbf{F}=\mathbf{D}_A^{\frac{1}{2}}\mathbf{Y}(\mathbf{Y}^T\mathbf{D}_A\mathbf{Y})^{-\frac{1}{2}}} Tr(\mathbf{F}^T\mathbf{D}_A^{-\frac{1}{2}}\mathbf{AD}_A^{-\frac{1}{2}}\mathbf{F}) \tag{6}$$

where $\mathbf{F} \in \mathcal{R}^{n\times c}$ is the cluster indicator matrix. We can relax $\mathbf{F}$ to continuous matrix and form the new problem

$$\max_{\mathbf{F}^T\mathbf{F}=\mathbf{I}} Tr(\mathbf{F}^T\mathbf{D}_A^{-\frac{1}{2}}\mathbf{AD}_A^{-\frac{1}{2}}\mathbf{F}) \tag{7}$$

It can be verified that the optimal solution of $\mathbf{F}$ is $\{\mathbf{F}^*\mathbf{R} : \mathbf{R}^T\mathbf{R} = \mathbf{I}_c\}$ where $\mathbf{F}^* \in \mathcal{R}^{n\times c}$ is the c column vectors of the eigenvectors of $\mathbf{D}_A^{-\frac{1}{2}}\mathbf{AD}_A^{-\frac{1}{2}}$ which correspond to the c biggest eigenvalues.

With $\mathbf{F}^*$, the next step is to obtain the discrete solution of $\mathbf{Y}$. In this paper, we propose to directly obtain the discrete solution $\mathbf{Y}$ by solving the following problem

$$\min_{\mathbf{Y}\in\mathcal{B}^{n\times c}, \mathbf{R}\in\mathcal{R}^{c\times c}} \left\|\mathbf{D}_A^{\frac{1}{2}}\mathbf{Y}(\mathbf{Y}^T\mathbf{D}_A\mathbf{Y})^{-\frac{1}{2}} - \mathbf{F}^*\mathbf{R}\right\|_F^2 \tag{8}$$
$$s.t.\ \mathbf{Y1}_c = \mathbf{1}_n, \mathbf{R}^T\mathbf{R} = \mathbf{I}_c$$

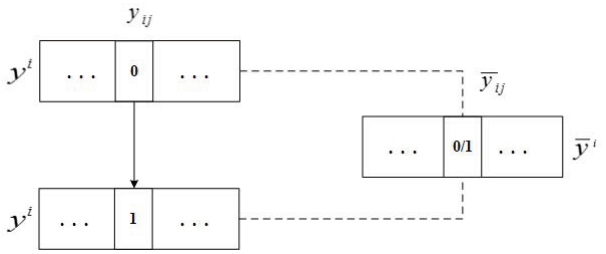


Figure 1: Illustration of computing the increment s_{ij}.

Note that $\left\|\mathbf{D}_A^{\frac{1}{2}}\mathbf{Y}(\mathbf{Y}^T\mathbf{D}_A\mathbf{Y})^{-\frac{1}{2}} - \mathbf{F}^*\mathbf{R}\right\|_F^2 = 2n - 2Tr((\mathbf{Y}^T\mathbf{D}_A\mathbf{Y})^{-\frac{1}{2}}\mathbf{Y}^T\mathbf{D}_A^{\frac{1}{2}}\mathbf{F}^*\mathbf{R})$, problem (8) can be rewritten as

$$\max_{\mathbf{Y}\in\mathcal{B}^{n\times c},\mathbf{R}\in\mathcal{R}^{c\times c},\mathbf{Y}\mathbf{1}_c=\mathbf{1}_n,\mathbf{R}^T\mathbf{R}=\mathbf{I}_c} Tr((\mathbf{Y}^T\mathbf{D}_A\mathbf{Y})^{-\frac{1}{2}}\mathbf{Y}^T\mathbf{D}_A^{\frac{1}{2}}\mathbf{F}^*\mathbf{R}) \tag{9}$$

We can apply the alternative optimization approach to solve problem (9).

4.1 Update $\mathbf{R}$ with $\mathbf{Y}$ fixed

If $\mathbf{Y}$ is fixed, denote $(\mathbf{Y}^T\mathbf{D}_A\mathbf{Y})^{-\frac{1}{2}}\mathbf{Y}^T\mathbf{D}_A^{\frac{1}{2}}$ as $\mathbf{M} \in \mathcal{R}^{c\times n}$. Suppose the SVD of $\mathbf{M}\mathbf{F}^*$ is $\mathbf{M}\mathbf{F}^* = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, then we have

$$Tr(\mathbf{M}\mathbf{F}^*\mathbf{R}) = Tr(\mathbf{R}\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T) = Tr(\mathbf{\Sigma}\mathbf{E}) = \sum_{i=1}^{c}\lambda_{ii}e_{ii} \tag{10}$$

where $\mathbf{E} = \mathbf{V}^T\mathbf{R}\mathbf{U}$, λ_{ii} and e_{ii} are the $(i,i)-th$ element of matrix $\mathbf{\Sigma}$ and $\mathbf{E}$ respectively.

Since $\mathbf{E}^T\mathbf{E} = \mathbf{U}^T\mathbf{R}\mathbf{V}\mathbf{V}^T\mathbf{R}^T\mathbf{U} = \mathbf{I}_c$, i.e, $\sum_{j=1}^{c} e_{ji}^2 = 1$, we know $e_{ii} \leq 1 (1 \leq i \leq c)$. On the other hand, $\lambda_{ii} \geq 0$ since λ_{ii} is singular value. Therefore, $Tr(\mathbf{M}\mathbf{F}^*\mathbf{R}) = \sum_{i=1}^{c}\lambda_{ii}e_{ii} \leq \sum_{i=1}^{c}\lambda_{ii}$, and the equality holds when $e_{ii} = 1$ $(1 \leq i \leq c)$. That is to say, $Tr(\mathbf{M}\mathbf{F}^*\mathbf{R})$ reaches its maximum when $\mathbf{E} = \mathbf{I}_c$. Then we obtain the optimal solution of $\mathbf{R}$ as

$$\mathbf{R} = \mathbf{V}\mathbf{U}^T \tag{11}$$

4.2 Update $\mathbf{Y}$ with $\mathbf{R}$ fixed

Let $\mathbf{G} = \mathbf{F}^*\mathbf{R}$. According to problem (9), the optimal solution of $\mathbf{Y}$ can be obtained by solving the following problem

$$\max_{\mathbf{Y}\in\mathcal{B}^{n\times c},\ \mathbf{Y}\mathbf{1}_c=\mathbf{1}_n} Tr(\mathbf{D}_A^{\frac{1}{2}}\mathbf{Y}(\mathbf{Y}^T\mathbf{D}_A\mathbf{Y})^{-\frac{1}{2}}\mathbf{G}^T) \tag{12}$$

which can be rewritten as

$$\max_{\mathbf{Y}\in\mathcal{B}^{n\times c},\ \mathbf{Y}\mathbf{1}_c=\mathbf{1}_n} \sum_{j=1}^{c}\frac{\sum_{i=1}^{n}\sqrt{d_{ii}}y_{ij}g_{ij}}{\sqrt{\mathbf{y}_j^T\mathbf{D}_A\mathbf{y}_j}} \tag{13}$$

Since $\sqrt{\mathbf{y}_j^T\mathbf{D}_A\mathbf{y}_j}$ involves all rows of $\mathbf{Y}$, we propose to sequentially solve $\mathbf{Y}$ row by row and fix the other rows of $\mathbf{Y}$ as constants. Suppose we have obtained the optimal solution $\overline{\mathbf{Y}}$, which has the objective function $\mathcal{J}^{old}(\overline{\mathbf{Y}})$. To solve the i-th row $\mathbf{y}^i \in \mathcal{B}^c$, we only need to consider the increment of the objective function value from $y_{ij} = 0$ to $y_{ij} = 1$. Since $\overline{\mathbf{y}}_j^T\mathbf{D}_A\overline{\mathbf{y}}_j$ and $\sum_{t=1}^{n}\sqrt{d}_{tt}\overline{y}_{tj}g_{tj}$ can be computed once before we solve $\mathbf{y}^i$, we can compute the increment as follows (See Figure 1)

$$\begin{aligned} s_{ij} =& \frac{\sum_{t=1}^{n}\sqrt{d}_{tt}\overline{y}_{tj}g_{tj} + \sqrt{d}_{ii}g_{ij}(1-\overline{y}_{ij})}{\sqrt{\overline{\mathbf{y}}_j^T\mathbf{D}_A\overline{\mathbf{y}}_j + d_{ii}(1-\overline{y}_{ij})}} \\ &- \frac{\sum_{t=1}^{n}\sqrt{d}_{tt}\overline{y}_{tj}g_{tj} - \sqrt{d}_{ii}\overline{y}_{ij}}{\sqrt{\overline{\mathbf{y}}_j^T\mathbf{D}_A\overline{\mathbf{y}}_j - d_{ii}\overline{y}_{ij}}} \end{aligned} \tag{14}$$

Then the optimal solution of $\mathbf{y}^i$ can be obtained as

$$y_{ij} =< l = \arg\max_{j'\in[1,c]} s_{ij'} > \tag{15}$$

where $< . >$ is 1 if the argument is true or 0 otherwise, and s_{ij} is defined in Eq. (14).

4.3 Initialization of $\mathbf{Y}$

We can use the the mapping in [Yu and Shi, 2003] to obtain the initial $\mathbf{Y}$. We first compute an approximate $\mathbf{Y}^*$ as

$$\mathbf{Y}^* = Diag(\mathbf{F}^*(\mathbf{F}^*)^T)^{\frac{1}{2}}\mathbf{F}^* \tag{16}$$

Then the initial discrete solution of $\mathbf{Y}$ is given by

$$y_{ij} =< j = \arg\max_{j'\in[1,c]} y_{ij'}^* > \tag{17}$$

4.4 The Optimization Algorithm

The detailed algorithm to solve problem (9), named Improved Spectral Rotation (ISR), is summarized in Algorithm 1. In the new algorithm, we need $O(r_1(c^3 + r_2nc))$ time to iteratively solve $\mathbf{R}$ and $\mathbf{Y}$ where r_1 is the number of iterations to update $\mathbf{R}$ and r_2 is the average number of iterations to update $\mathbf{Y}$. Considering that $c \ll n$ for large scale data, the computational complexity for obtaining $\mathbf{Y}$ is $O(r_1r_2nc)$. If we use k-means to obtain $\mathbf{Y}$, we need $O(tnc^2)$ time where t is the number of iterations. Here, the discrete solution $\mathbf{Y}$ converges very fast due to its limited solution space so r_2 is usually very small. Therefore, ISR has almost similar computational complexity as k-means for large scale data.

Algorithm 1 Improved Spectral Rotation (ISR) to solve problem (9)

1: **Input:** $\mathbf{F}^*$.
2: Initialize $\mathbf{Y}$ according to Eq. (17).
3: **repeat**
4: Update $\mathbf{R}$ according to Eq. (11), and $\mathbf{G} = \mathbf{F}^*\mathbf{R}$.
5: **repeat**
6: Update $\mathbf{Y}$ according to Eq. (15).
7: **until** $\mathbf{Y}$ does not change
8: **until** problem (9) converges
9: **Output:** the clustering result $\mathbf{Y}$.

5 The Scalable Normalized Cut for Large Data

In this section, we propose a Scalable Normalized Cut (SNC) for large scale data.

5.1 An Efficient Method for Construction Representation Matrix

Assume that we have obtained m anchors $\mathbf{W} \in \mathcal{R}^{d\times m}$ with k-means, the next step is to construct a representation matrix $\mathbf{B} \in \mathcal{R}^{n\times m}$. Inspired from the work in [Nie *et al.*, 2016], we assume that $\mathbf{b}_{ij}$ should be larger if $\mathbf{x}_i$ is closer to $\mathbf{w}_j$ and propose an efficient method to construct $\mathbf{B}$. For the i-th sample $\mathbf{x}_i \in \mathbf{X}$, we propose to obtain $\mathbf{b}^i \in \mathcal{R}^m$ by solving the following problem

$$\min_{\mathbf{b}^i\mathbf{1}=1,\mathbf{b}^i\geq 0} \sum_{j=1}^{m} b_{ij}\left\|\mathbf{x}_i-\mathbf{w}_j\right\|_2^2+\gamma\sum_{j=1}^{m} b_{ij}^2 \tag{18}$$

According to the analysis in [Nie *et al.*, 2016], the optimal solution $\mathbf{b}^i$ to problem (18) is

$$b_{ij}=\begin{cases}\frac{d_{i,k+1}-\|\mathbf{x}_i-\mathbf{w}_j\|_2^2}{kd_{i,k+1}-\sum_{h=1}^{k} d_{i,h}} & \mathbf{x}_j\in\mathcal{N}_k(\mathbf{x}_i)\\ 0 & \text{otherwise}\end{cases} \tag{19}$$

where $d_{i,h}$ is the square of Euclidean distance between $\mathbf{x}_i$ and its h-th nearest neighbor, and $\mathcal{N}_k(\mathbf{x}_i)$ contains the k nearest neighbors of $\mathbf{x}_i$.

After obtaining the representation matrix $\mathbf{B}$, we can compute the affinity matrix $\mathbf{A}$ according to Eq. (1). The following theorem ensures that $\mathbf{A}$ is symmetric and doubly stochastic.

Theorem 1. *Given the representation matrix* $\mathbf{B}$ *computed according to Eq. (19),* $\mathbf{A}$ *computed from Eq. (1) is symmetric and doubly stochastic.*

Proof. According to Eq. (1), we have

$$a_{ij}=\sum_{l=1}^{m}\frac{b_{il}b_{jl}}{\sum_{t=1}^{n} b_{tl}} \tag{20}$$

It can be easily verified that $a_{ij}=a_{ji}$, which indicates that $\mathbf{A}$ computed from Eq. (1) is symmetric.

We can also verify that

$$\sum_{j=1}^{n} a_{ij}=\sum_{l=1}^{m}\frac{b_{il}\sum_{j=1}^{n} b_{jl}}{\sum_{t=1}^{n} b_{tl}}=\sum_{l=1}^{m} b_{il}=1 \tag{21}$$

which implies that $\sum_{i=1}^{n} a_{ij}=\sum_{j=1}^{n} a_{ij}=1$. Therefore, $\mathbf{A}$ is doubly stochastic. □

5.2 The Optimization Model

According to Theorem 1, it can be verified that the degree matrix of $\mathbf{A}$ should be an identity matrix. Then problem (6) can be rewritten as

$$\max_{\mathbf{Y}\in Ind,\mathbf{F}=\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-\frac{1}{2}}} Tr(\mathbf{F}^T\mathbf{A}\mathbf{F}) \tag{22}$$

We also relax $\mathbf{F}$ to continuous matrix, and obtain the optimal solution of $\mathbf{F}$ from the following problem

$$\max_{\mathbf{F}^T\mathbf{F}=\mathbf{I}} Tr(\mathbf{F}^T\mathbf{A}\mathbf{F}) \tag{23}$$

Note that $\mathbf{A}$ can be rewritten as $\mathbf{A}=\mathbf{P}\mathbf{P}^T$ where $\mathbf{P}\in\mathcal{R}^{n\times m}=\mathbf{B}\Delta^{-\frac{1}{2}}$, we can perform SVD on $\mathbf{P}$ instead of $\mathbf{A}$. Suppose the SVD of $\mathbf{P}$ is $\mathbf{P}=\mathbf{U}_P\mathbf{\Sigma}_P\mathbf{V}_P^T$, we have $\mathbf{A}=\mathbf{P}\mathbf{P}^T=\mathbf{U}_P\mathbf{\Sigma}_P^2\mathbf{U}_P^T$, which can be rewritten as

$$\mathbf{A}\mathbf{U}_P=\mathbf{U}_P\mathbf{\Sigma}_P^2 \tag{24}$$

then we know that the optimal solution of F to problem (23) is the c column vectors in $\mathbf{U}_P$ corresponding to c biggest diagonal entries in diagonal matrix $\mathbf{\Sigma}_P^2$.

With the learnt optimal solution of $\mathbf{F}^*$, we can use Algorithm 1 to obtain the final solution of $\mathbf{Y}$. Since the size of $\mathbf{P}$ is $n\times m$, we can obtain $\mathbf{F}^*$ within $O(nm^2)$.

Following the same analysis in [Nie *et al.*, 2011], it can be verified that although $\mathbf{1}_n$ is a trivial vector in $\mathbf{U}_P$, it should be retained in order to generate the whole set of optima.

5.3 The Optimization Algorithm

The detailed algorithm to solve problem (22), named Scalable Normalized Cut (SNC), is summarized in Algorithm 2. Given a data matrix $\mathbf{X}\in\mathcal{R}^{d\times n}$, we need $O(ndmt)$ time to obtain m anchors by k-means where t is the number of iterations, $O(ndm+nm\log(m))$ time to construct $\mathbf{P}$, $O(nm^2)$ time to obtain $\mathbf{F}^*$, and $O(r_1(c^3+r_2nc))$ time to iteratively solve $\mathbf{R}$ and $\mathbf{Y}$ where r_1 is the number of iterations to update $\mathbf{R}$ and r_2 is the average number of iterations to update $\mathbf{Y}$. Here, the discrete solution $\mathbf{Y}$ converges very fast due to its limited solution space so r_2 is usually very small. Considering that $m\ll n$, $d\ll n$ and $c<m$ for large scale data, the overall computational complexity is $O(ndmt+nm^2)$. Therefore, SNC has the same computational complexity as LSC.

Algorithm 2 Scalable Normalized Cut (SNC) to solve problem (22)

1: **Input:** Data matrix $\mathbf{X}\in\mathbb{R}^{d\times n}$, number of nearest neighbors k, number of anchors m.
2: Find m anchors $\mathbf{W}$ using k-means, and construct a sparse representation matrix $\mathbf{B}\in\mathcal{R}^{n\times m}$ with the method introduced in Section 5.1.
3: Obtain $\mathbf{P}\in\mathcal{R}^{n\times m}=\mathbf{B}\Delta^{-\frac{1}{2}}$, where $\Delta\in\mathcal{R}^{m\times m}$ is the degree matrix of $\mathbf{B}$.
4: Perform SVD on $\mathbf{P}$ such that $\mathbf{P}=\mathbf{U}_P\mathbf{\Sigma}_P\mathbf{V}_P^T$, then form $\mathbf{F}^*$ by selecting c column vectors in $\mathbf{U}_P$ which corresponds to c biggest diagonal entries in diagonal matrix $\mathbf{\Sigma}_P^2$.
5: Call Algorithm 1 with input $\mathbf{F}^*$ to obtain the optimal solution of $\mathbf{Y}$.
6: **Output:** the clustering result $\mathbf{Y}$.

6 Experimental results and analysis

In this section, we present the experiments conducted on 14 real-life data sets to demonstrate the efficiency and effectiveness of the proposed method.

6.1 Experiments on ISR

We first compared ISR with k-means and the original spectral rotation for spectral clustering.

Table 1: Characteristics of 8 data sets.

Data sets	Name	No. of samples	No. of features	No. of classes
D_1	colon	62	2000	2
D_2	srbct	63	2308	4
D_3	breast3	95	4869	3
D_4	nci	61	5244	8
D_5	LM	360	90	15
D_6	Coil20Data-25	1440	1024	20
D_7	PalmData25	2000	256	100
D_8	corel-5k	5000	423	50

Table 2: Comparison results of the average clustering results in terms of *Accuracy (NMI)*. The best result on each data set is highlighted in bold.

Data	NCut+KM	NCut+SR	NCut+ISR
D_1	0.648(0.063)	0.688(0.094)	**0.728**(**0.143**)
D_2	0.608(0.442)	**0.614**(0.417)	0.598(**0.465**)
D_3	0.588(0.196)	0.593(0.204)	**0.604**(**0.214**)
D_4	0.713(0.687)	0.691(0.650)	**0.749**(**0.689**)
D_5	0.487(0.638)	0.483(0.620)	**0.497**(**0.644**)
D_6	0.764(**0.853**)	0.705(0.807)	**0.798**(0.853)
D_7	0.761(0.915)	0.861(0.950)	**0.867**(**0.962**)
D_8	0.182(0.286)	0.185(0.273)	**0.192**(**0.289**)

Benchmark data sets

8 benchmark data sets were selected from the UCI Machine Learning Repository and Feiping Nie's page [1]. Table 1 summarizes the characteristics of these 8 data sets.

Results and Analysis

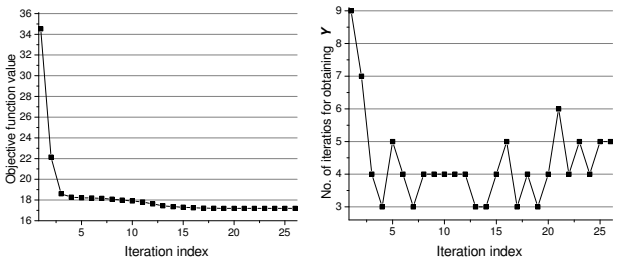
We compared ISR with k-means (KM) and the original spectral rotation (SR) for normalized cut. For each data set, we set five neighborhood parameters $k = \{10, 20, \ldots, 50\}$ to construct five affinity matrices with the method in [Nie *et al.*, 2016], and used these matrices to run three methods in order to perform fair comparison. For each algorithm on each data set, we computed the average accuracy and NMI and show them in Table 2. From these figures, we can see that ISR outperformed other methods in both accuracy and NMI on almost all data sets. Especially on D_1, D_4 and D_8, ISR has over 5% improvement compared to the second best method. This indicates that ISR improves the original spectral rotation.

We selected D_8 to show the convergence curves of the objective function value and the number of iterations for obtaining $\mathbf{Y}$ in each main loop. The results are drawn in Figure 2. From Figure 2(a), we can see that the objective function value drops very fast, indicating Algorithm 1 converges very fast. From Figure 2(b), we can see that the average number of iterations for obtaining $\mathbf{Y}$ is around 4. Therefore, ISR can quickly obtain the final clustering assignments.

6.2 Experiments on SNC

In this subsection, we compare SNC with the original normalized cut and other scalable spectral clustering methods.

[1] http://www.escience.cn/people/fpnie/index.html#



(a) Objective function values of problem (9). (b) No. of iterations for obtaining $\mathbf{Y}$ in each main loop.

Figure 2: Objective function values of problem (9) and no. of iterations for obtaining $\mathbf{Y}$ in each main loop of ISR on D_8.

Table 3: Characteristics of 6 data sets.

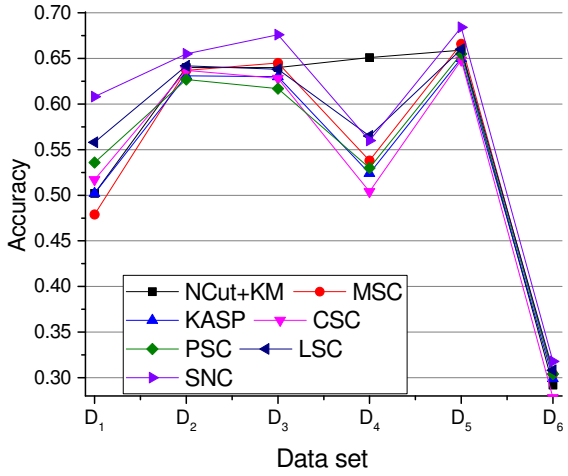
Data sets	Name	No. of samples	No. of features	No. of classes
D_1	segment	2310	19	7
D_2	MnistData-05	3495	784	10
D_3	MnistData-10	6996	784	10
D_4	isolet5	7797	617	26
D_5	USPS	9298	256	10
D_6	letter-recognition	20000	16	26

Benchmark data sets

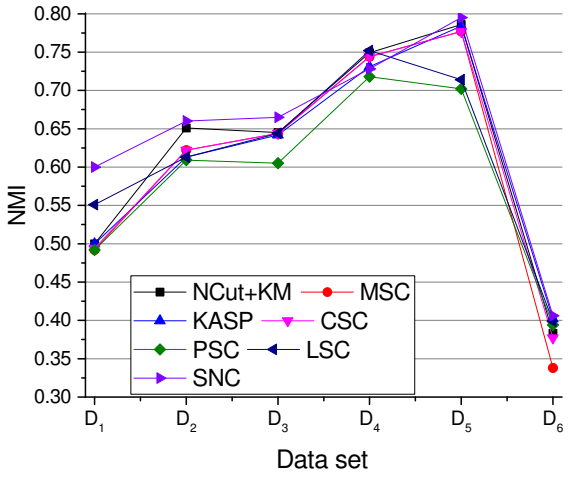
6 large scale benchmark data sets were selected from the UCI Machine Learning Repository and Feiping Nie's page [1]. Table 3 summarizes the characteristics of these 6 data sets.

Results and Analysis

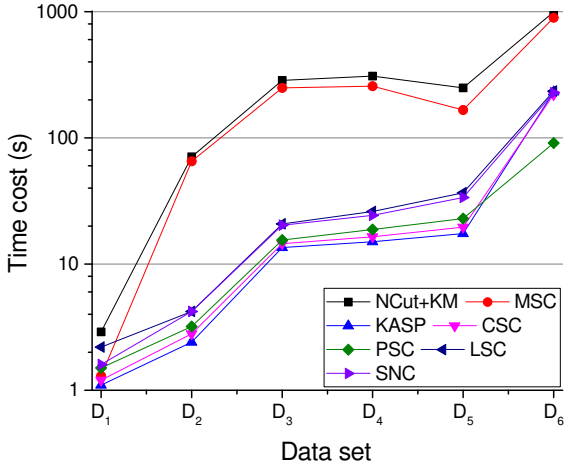
We compared SNC with six spectral clustering methods, including NCut with k-means (NCut+KM) [Ng *et al.*, 2002], multiclass spectral clustering (MSC) [Yu and Shi, 2003], k-means-based approximate spectral clustering (KASP) [Yan *et al.*, 2009], committees-based spectral clustering (CSC) [Shinnou and Sasaki, 2008], parallel spectral clustering (PSC) [Chen *et al.*, 2011] and LSC [Cai and Chen, 2015]. For each data set, we used the same clustering result for anchors generation in KASP, CSC, LSC and SNC where 10 numbers were selected for m. The neighborhood parameters were set as $\{10, 20, \ldots, 50\}$ for all data sets. We used the Gaussian kernel to compute similarities for all methods excluding SNC, where the parameter h was set as the average distance between two points in the data set (used in [Cai and Chen, 2015]). The average clustering performance of seven spectral clustering algorithms are shown in Figure 3. From these figures, we can see that SNC outperformed other methods in accuracy and NMI on almost all data sets. Especially on D_1, SNC has nearly 10% improvement compared to the second best method NCut+KM in terms of both accuracy and NMI. On five data sets, SNC outperformed both NCut+KM and MNC which perform clustering with the similarity matrix computed from the original data. From Figure 3(c), we can see that the time costs of SNC are much smaller than NCut+KM and MNC, especially on D_2, D_3 and D_4. The time costs of SNC are similar as those of LSC. Although SNC spent more time than KASP, CSC and PSC, it produced better results than these methods.



(a) Accuracy results.



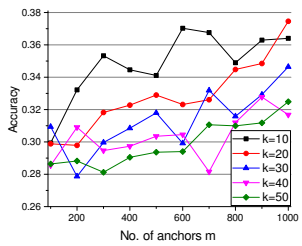
(b) NMI results.



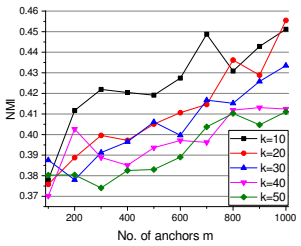
(c) Time cost results.

Figure 3: Comparison results of seven clustering algorithms on six data sets.

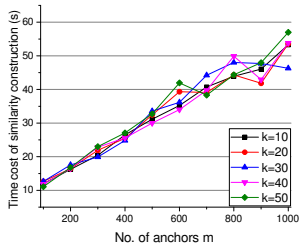
Parameter study



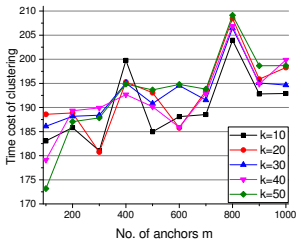
(a) Accuracy versus m and k.



(b) NMI versus m and k.



(c) Time costs of similarity construction versus m and k.



(d) Time costs of clustering versus m and k.

Figure 4: Accuracy, nmi and running time of SNC versus the no. of anchors m and neighborhood parameter k.

We select D_6 to show the relationship between the clustering performance and two parameters m, k in SNC. The results are drawn in Figure 4. From these figures, it can be seen that SNC can achieve better clustering results (in terms of both accuracy and nmi) as both m and k increase. From Figure 4(c), we can see that the time cost of similarity construction grows linearly as m increases, and the time cost of clustering does not change too much as k increases. From Figure 4(d), we can see that the time cost of clustering is insensitive to both m and k. Since the time cost of similarity construction is much smaller than the time cost of clustering, we can say that the total time cost of SNC is nearly insensitive to both m and k.

7 Conclusions

In this paper, we have proposed a Scalable Normalized Cut (SNC) method for large scale data, in which a parameter-free method is proposed to construct a representation matrix, and an Improved Spectral Rotation (ISR) method is proposed to obtain the final clustering assignments. Experimental results show that ISR can obtain better clustering results than k-means and the original spectral rotation. Comparison results with other scalable spectral clustering methods show that our method can obtain better results without increasing running time too much. Therefore, the new method is effective and efficient for large scale data. In the future work, we will study new anchor generation method.

Acknowledgement

This research was supported by NSFC under Grant no.61305059, 61473194 and U1636202.

References

[Cai and Chen, 2015] Deng Cai and Xinlei Chen. Large scale spectral clustering via landmark-based sparse representation. *IEEE Transactions on Cybernetics*, 45(8):1669–1680, 2015.

[Cai *et al.*, 2011] Xiao Cai, Feiping Nie, Heng Huang, and Farhad Kamangar. Heterogeneous image feature integration via multi-modal spectral clustering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1977–1984. IEEE, 2011.

[Chen *et al.*, 2011] Wen-Yen Chen, Yangqiu Song, Hongjie Bai, Chih-Jen Lin, and Edward Y. Chang. Parallel spectral clustering in distributed systems. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 33(3):568–586, 2011.

[Chen *et al.*, 2012] Xiaojun Chen, Yunming Ye, Xiaofei Xu, and Joshua Zhexue Huang. A feature group weighting method for subspace clustering of high-dimensional data. *Pattern Recognition*, 45(1):434–446, 2012.

[Chen *et al.*, 2013] Xiaojun Chen, Xiaofei Xu, Yunming Ye, and Joshua Zhexue Huang. TW-k-means: Automated Two-level Variable Weighting Clustering Algorithm for Multi-view Data. *IEEE Transactions on Knowledge and Data Engineering*, 25(4):932–944, 2013.

[de Souto *et al.*, 2008] Marcilio CP de Souto, Ivan G Costa, Daniel SA de Araujo, Teresa B Ludermir, and Alexander Schliep. Clustering cancer gene expression data: a comparative study. *BMC Bioinformatics*, 9(1):497, 2008.

[Fowlkes *et al.*, 2010] Charless Fowlkes, Serge Belongie, Fan Chung, and Jitendra Malik. Spectral grouping using the nyström method. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 26(2):214–225, 2010.

[Hagen and Kahng, 1992] Lars Hagen and Andrew B. Kahng. New spectral methods for ratio cut partitioning and clustering. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 11(9):1074–1085, 1992.

[Huang *et al.*, 2013] Jin Huang, Feiping Nie, and Heng Hu. Spectral rotation versus k-means in spectral clustering. In *AAAI Conference on Artificial Intelligence*, volume 1-3, 2013.

[Kriegel *et al.*, 2009] Hans-Peter Kriegel, Peer Kröger, and Arthur Zimek. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Transactions on Knowledge Discovery from Data*, 3(1):1–58, 2009.

[Li *et al.*, 2010] Mu Li, James T. Kwok, and Bao Liang Lu. Making large-scale nyström approximation possible. In *International Conference on Machine Learning*, pages 631–638, 2010.

[Liu *et al.*, 2010] Wei Liu, Junfeng He, and Shih Fu Chang. Large graph construction for scalable semi-supervised learning. In *International Conference on Machine Learning*, pages 679–686, 2010.

[Ng *et al.*, 2002] Andrew Y Ng, Michael I Jordan, Yair Weiss, et al. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 2:849–856, 2002.

[Nie *et al.*, 2010] Feiping Nie, Chris Ding, Dijun Luo, and Heng Huang. Improved minmax cut graph clustering with nonnegative relaxation. In *Proceedings of ECML-PKDD*, pages 451–466, 2010.

[Nie *et al.*, 2011] Feiping Nie, Zinan Zeng, Ivor W Tsang, Dong Xu, and Changshui Zhang. Spectral embedded clustering: a framework for in-sample and out-of-sample spectral clustering. *IEEE Transactions on Neural Networks*, 22(11):1796–808, 2011.

[Nie *et al.*, 2016] Feiping Nie, Xiaoqian Wang, Michael Jordan, and Heng Huang. The constrained laplacian rank algorithm for graph-based clustering. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 1969–1976, 2016.

[Samaria and Harter, 1995] F.S. Samaria and A.C. Harter. Parameterisation of a stochastic model for human face identification. In *Proceedings of the Second IEEE Workshop on Applications of Computer Vision*, pages 138–142, 1995.

[Sánchez-García *et al.*, 2014] Rubén J. Sánchez-García, Max Fennelly, Seán Norris, Nick Wright, Graham Niblo, Jacek Brodzki, and Janusz W. Bialek. Hierarchical spectral clustering of power grids. *IEEE Transactions on Power Systems*, 29(5):2229–2237, Sept 2014.

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

[Shinnou and Sasaki, 2008] Hiroyuki Shinnou and Minoru Sasaki. Spectral clustering for a large data set by reducing the similarity matrix size. In *International Conference on Language Resources and Evaluation, Lrec 2008, 26 May - 1 June 2008, Marrakech, Morocco*, pages 201–204, 2008.

[Von Luxburg, 2007] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[Yan *et al.*, 2009] Donghui Yan, Ling Huang, and Michael I Jordan. Fast approximate spectral clustering. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 907–916, 2009.

[Yu and Shi, 2003] Stella X. Yu and Jianbo Shi. Multiclass spectral clustering. In *Proceedings of IEEE International Conference on Computer Vision*, pages 313–319 vol.1, 2003.