Orthogonal and Nonnegative Graph Reconstruction for Large Scale Clustering

Junwei Han[1], **Kai Xiong**[1], **Feiping Nie**[1,2]
[1]Northwestern Ploytechnical University, Xi'an, 710072, P. R. China
[2]University of Texas at Arlington, USA
{junweihan2010, bearkai1992, feipingnie}@gmail.com

Abstract

Spectral clustering has been widely used due to its simplicity for solving graph clustering problem in recent years. However, it suffers from the high computational cost as data grow in scale, and is limited by the performance of post-processing. To address these two problems simultaneously, in this paper, we propose a novel approach denoted by orthogonal and nonnegative graph reconstruction (ONGR) that scales linearly with the data size. For the relaxation of *Normalized Cut*, we add nonnegative constraint to the objective. Due to the nonnegativity, ONGR offers interpretability that the final cluster labels can be directly obtained without post-processing. Extensive experiments on clustering tasks demonstrate the effectiveness of the proposed method.

1 Introduction

Clustering is an important topic in machine learning and data mining. As a branch of clustering, graph clustering has been drawing growing attention which aims to group vertices of the graph into clusters with the expection that there are more edges within each cluster and fewer edges between the clusters [Schaeffer, 2007; Zhang *et al.*, 2014; Wang *et al.*, 2016]. As the field of graph clustering involves too many aspects, we mainly focus on spectral clustering which is considered as a way to solve relaxed versions of graph cut problems including *Normalized Cut* (*Ncut*) and *RatioCut*. Spectral clustering has limited applicability to process large scale data, since it uses eigenvectors of the Laplacian matrix. In general, it takes $O(n^3)$ for eigen-decomposition with n denoting the number of data points.

Generally, there are two major directions to solve the scalability issue. One is to reduce the computational cost of the eigen-decomposition problem. [Fowlkes *et al.*, 2004] firstly adopted the classical Nyström method to extrapolate the complete clustering results using only a small number of samples. The works in [Li *et al.*, 2011] and [Choromanska *et al.*, 2013] also attempted to alleviate the computational burden based on Nyström method with improvements on complexities or with attached performance guarantees. Spectral sparsification can be seen in [Spielman and Teng, 2011] with the computational cost relying on the number of edges in original graph. From the point of view of hardware platform, [Chen *et al.*, 2011] sparsified the matrix via retaining k-nearest neighbors, paralleling both memory use and computation on distributed computers.

Another direction is to reduce the data size by sampling some representative points beforehand. [Yan *et al.*, 2009] developed a general framework that spectral clustering only needs to run on a small subset. [Shinnou and Sasaki, 2008] adopted a committees-based way with a slightly different process to obtain the representatives. [Cai and Chen, 2014] selected a few landmarks for spectral embedding while [Peng *et al.*, 2013] chose some in-sample data for sparse subspace clustering, and the remaining data are represented as a linear combination of the pre-selected points. [Liu *et al.*, 2013] devoted to handling graph data, by generating supernodes to compress the original graph into a sparse bipartite graph.

Most previous works can not achieve good clustering results since they are limited by the performance of post-processing. For example, *kmeans* is a common way to obtain the final cluster labels, while *kmeans* itself is sensitive to the initialization. To get rid of this extra step, and at the same time to be able to handle the scalability issue, we propose a novel approach called orthogonal and nonnegative graph reconstruction (ONGR). It is worthwhile to highlight the main contributions of the paper as follows:

1. ONGR is proposed from the viewpoint of graph reconstruction, which is different from both two kinds of existing methods aiming at handling the scalability issue.
2. ONGR adds nonnegative constraint to the objective, thus offering interpretability that the final cluster labels can be directly obtained without post-processing.
3. Due to the orthogonal and nonnegative constraints, ONGR reconstructs the graph by a structured one which is good for clustering tasks.
4. ONGR has close relationship with spectral clustering and nonnegative matrix factorization. Comprehensive experiments on a variety of datasets show the effectiveness of the proposed method.

Notations: Suppose we have n data points belonging to k clusters with the dimensionality denoted by d. $I_k \in \mathbb{R}^{k\times k}$ is an identity matrix and $\mathbf{0}$ is a zero matrix of proper size. The Frobenius norm is denoted by $\|\cdot\|_F$. Let $G_r = (V_s, E)$ be

an undirected weighted graph with vertices $V_s = \{v_i\}_{i=1}^n$ and edges E. The symmetric similarity matrix of the graph is $W = \{w_{ij}\}_{i,j=1}^n$ and the degree of v_i is defined as $d_i = \sum_{j=1}^n w_{ij}$. The diagonal degree matrix D has $\{d_i\}_{i=1}^n$ on the diagonal. The Laplacian matrix is $L = D - W$, and the normalized one is $\tilde{L} = D^{-1/2}LD^{-1/2}$.

2 Revisit of Spectral Clustering

We first review some concepts in graph cut [Schaeffer, 2007; Von Luxburg, 2007]. Given a subset $A \subset V_s$, the complement of which is denoted by $\bar{A}$, and the size can be measured by the number of vertices denoted by $|A|$, or the sum of degrees of vertices denoted by $vol(A) = \sum_{i\in A} d_i$.

Thus, we can define the cut between A and $\bar{A}$ as $cut(A, \bar{A}) = \sum_{i\in A, j\in\bar{A}} w_{ij}$. Further, the definitions of *RatioCut* [Hagen and Kahng, 1992] and *Ncut* [Shi and Malik, 2000] can be respectively represented as $RatioCut(A_1, \ldots, A_k) = \sum_{i=1}^k \frac{cut(A_i,\bar{A}_i)}{|A_i|}$, $NCut(A_1, \ldots, A_k) = \sum_{i=1}^k \frac{cut(A_i,\bar{A}_i)}{vol(A_i)}$, where k nonempty subsets $A_1, \ldots, A_k$ satisfy $A_i \bigcap A_j = \emptyset (i, j = 1, \ldots, k)$, and $A_1 \bigcup, \ldots, \bigcup A_k = V_s$. Minimizing these two problems is *NP* hard and it is natural to solve the relaxed problems.

Define k cluster indicator vectors $f_j = (f_{1j}, \ldots, f_{nj})'$ by

$$f_{ij} = \begin{cases} 1/\sqrt{vol(A_j)}, & \text{if } v_i \in A_j \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

where $i = 1, \ldots, n, j = 1, \ldots, k$. Let $F \in \mathbb{R}^{n\times k}$ be the *indicator matrix* that consists of the indicator vectors as columns. It is not difficult to verify $F^TDF = I$, $f_i'Lf_i = \frac{cut(A_i,\bar{A}_i)}{vol(A_j}(i = 1, \ldots, k)$. So the minimization problem of *NCut* can be rewritten as

$$\begin{aligned} &\min_{A_1,\ldots,A_k} Tr(F^TLF) \\ &s.t.\ F^TDF = I, F \text{ defined as Eq.(1).} \end{aligned} \quad (2)$$

Substituting $H = D^{1/2}F$ to Eq.(2), the relaxation of minimizing *NCut* can be obtained by allowing the entries of *indicator matrix* to be arbitrary real values, *i.e.*,

$$\min_{H\in\mathbb{R}^{n\times k}} Tr\left(H^T\tilde{L}H\right) s.t.\ H^TH = I, \quad (3)$$

Eq.(3) is exactly the objective of normalized spectral clustering[Ng *et al.*, 2002]. Similarly, to approximate *RatioCut*, the indicator vectors are defined the same as Eq.(1) with $|A_j|$ replacing $vol(A_j)$. We have $F^TF = I$, and the relaxation of minimizing *RatioCut* can be reformulated as

$$\min_{F\in\mathbb{R}^{n\times k}} Tr(F^TLF) \quad s.t.\ F^TF = I. \quad (4)$$

which is the objective of unnormalized spectral clustering. Note that when W is a doubly-stochastic matrix (a nonnegative square matrix satisfies that row sum and column sum all equal to 1), we have $\tilde{L} = I - W = L, H = F$. Then Eq.(3) becomes the same as Eq.(4), which means the equivalence between *RatioCut* and *Ncut* under the condition.

Utilizing a doubly-stochastic similarity matrix is usually good for clustering tasks [Zass and Shashua, 2006; Wang *et al.*, 2010; 2016]. In the subsequent sections, W is by default to be doubly-stochastic, and we try to reconstruct it by a structured graph.

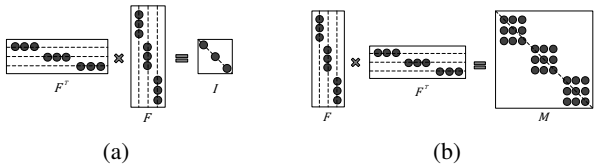


Figure 1: An illustration of the indicator matrix and the reconstructed graph under orthogonal and nonnegative constraints in Eq.(6). For simplicity, we take $n = 9, k = 3$. Black dots mean the non-zero entries, and we have reorganized the rows of F to place data points with the same cluster label continuously.

3 The Proposed Method

3.1 Formulation

In spectral clustering, the eigenvectors of Laplacian matrix can be considered as the relaxed indicator vectors, but it lacks interpretability and thus relies on the post-processing. To get rid of the extra step, we add additional nonnegative constraint to get discrete indicator vectors. The objective with two constraints is as follows:

$$\min_{F^TF=I, F\geq 0} Tr(F^TLF). \quad (5)$$

The *Ncut* problem in Eq.(2) has discrete indicator vectors whose entries are also nonnegative, while spectral clustering relaxes the entries to be any real values. Intuitively, compared to spectral clustering, the objective in Eq.(5) is much closer to *Ncut*, thus our model tends to get better performance.

Due to the nonnegativity, the objective in Eq.(5) offers the interpretability that entries in the *indicator matrix* directly correspond to relationship between data points and clusters. The constraints lead F to be a matrix that there is only one non-zero entry in each row, and the ℓ_2-norm of each column is 1. We conceptually illustrate the constraints in Figure 1(a). The cluster labels can then be obtained by finding the column index of the non-zero entry in each row of F. Eq.(5) is hard to tackle. Considering the computational cost, we take a circuitous way by proposing an approximated model. We first have the following proposition to transform Eq.(5).

Proposition 1. *Solving Eq.(5) is equivalent to solve:*

$$\min_{F^TF=I, F\geq 0} \|W - FF^T\|_F^2. \quad (6)$$

Proof. Note that W is symetric and doubly-stochastic. $Tr(W^TW)$ and $Tr\left(FF^TFF^T\right)$ are two constant terms.

$$\begin{aligned} \text{Eq.(5)} &\Leftrightarrow \max_{F^TF=I, F\geq 0} Tr(F^TWF) \\ &\Leftrightarrow \min_{F^TF=I, F\geq 0} -Tr(W^TFF^T + FF^TW) \\ &\Leftrightarrow \min_{F^TF=I, F\geq 0} Tr[W^TW - W^T(FF^T) - (FF^T)W \\ &\qquad + (FF^T)(FF^T)] \Leftrightarrow \text{Eq.(6)} \end{aligned}$$ □

Eq.(6) is important for developing the new model. It can be considered as a kind of graph reconstructoin with orthogonal and nonnegative constraints. The original constructed graph does not have clear structure since there usually exists noise in the data, while the graph reconstruction is an optimization process to learn a structured graph. We illustrate the reconstruction proecss in Figure 1(b). As we can see, the objective in Eq.(6) tries to reconstruct the similarity matrix by a block diagonal matrix. Such a clear structure [Nie *et al.*, 2014] contains more accurate information about the clusters thus it is good for clustering tasks. Our new model is formulated as follows:

$$\min_{F^TF=I, G\geq 0} \|W - FG^T\|_F^2 + \lambda\|F - G\|_F^2. \tag{7}$$

The first term is the reconstruction term and the second one is the regularization term which forces F and G to be close to each other. λ is the trade-off parameter that balances the two terms, and $G \in \mathbb{R}^{n\times k}$ can be called the *label matrix* that gives the final cluster labels. It can be seen that Eq.(7) is equivalent to Eq.(6) for a large enough value of λ.

F is relaxed to have continuous values, but it differs from spectral clustering since F in our new model has the restriction of being close to a nonnegative matrix G. Compared to Eq.(6), Eq.(7) is easier to solve with less computational cost but still provides interpretability. In a sense, the interpretability of F is passed on to G. Entries in the *label matrix* G can be regarded as a soft relationship between data points and clusters, and cluster labels can be obtained by finding the column index of the largest entry in each row of G.

3.2 Optimization

To solve the non-convex problem in Eq.(7), We divide it into two subproblems that are optimized iteratively. The whole procedure is summarized in Algorithm 1 which converges fast, and in the experimental part we will see that the local optimal solution is good enough due to the effective initializaiton.

Step 1: Update G with F fixed An usual way to solve Eq.(7) with F fixed may use Lagrange multiplier method and the KKT condition. Note that the fixed F still meets the condition of $F^TF = I$. Instead, we take a simpler way to obtain the optimal solution. By adding or removing some constant terms, we can rewrite the problem as follows:

$$\begin{aligned}
&\min_{G\geq 0} \|W - FG^T\|_F^2 + \lambda\|F - G\|_F^2 \\
&\Leftrightarrow \min_{G\geq 0} Tr(G^TG - 2G^TWF) + \lambda Tr(G^TG - 2G^TF) \\
&\Leftrightarrow \min_{G\geq 0} \|G - \frac{WF + \lambda F}{1+\lambda}\|_F^2.
\end{aligned} \tag{8}$$

The optimal solution of the *label matrix* can then be written as

$$G = \left(\frac{WF + \lambda F}{1+\lambda}\right)_+, \tag{9}$$

where $(\cdot)_+$ is to make each entry of the matrix its absolute value.

Step 2: Update F with G fixed Suppose the SVD of $(WG + \lambda G)$ is $U\Lambda V^T$, and $Q = V^TF^TU \in \mathbb{R}^{k\times n}$. Eq.(7) with G fixed can be rewritten as:

$$\begin{aligned}
&\min_{F^TF=I} \|W - FG^T\|_F^2 + \lambda\|F - G\|_F^2 \\
&\Leftrightarrow \max_{F^TF=I} Tr\left(F^T(WG + \lambda G)\right). \\
&\Leftrightarrow \max_{F^TF=I} Tr\left(F^TU\Lambda V^T\right). \\
&\Leftrightarrow \max_{F^TF=I} Tr\left(\Lambda V^TF^TU\right). \\
&\Leftrightarrow \max_{F^TF=I} \sum_i \Lambda_{ii}Q_{ii}.
\end{aligned} \tag{10}$$

Λ_{ii}, Q_{ii} are the (i,i)-th entry of Λ and Q, respectively. Note that Λ_{ii} is singular value and $QQ^T = I_k$, thus we have $\Lambda_{ii} \geq 0$ and $Q_{ii} \leq 1$. Therefore, $\sum_i \Lambda_{ii}Q_{ii} \leq \sum_i \Lambda_{ii}$ and the equality holds when $\{Q_{ii} = 1\}_{i=1}^k$, *i.e.*, $Q = [I_k, \mathbf{0}]$. Recall that $Q = V^TF^TU$, so the optimal solution to Eq.(10) is

$$F = U[I_k; \mathbf{0}]V^T. \tag{11}$$

Algorithm 1 ONGR Algorithm

Input: $X = [x_1, \ldots, x_n] \in \mathbb{R}^{d\times n}$, cluster number k, anchor number m, nearest anchor number s and trade-off parameter λ;
Output: Cluster labels Y.

```
Select m anchors using kmeans or random selection;
Construct a sparse regression matrix Z by Eq.(12);
Initialize F ∈ R^{n×k} by left singular vectors of Ẑ;
repeat
    Update G by Eq.(9);
    Update F by Eq.(11);
until converges.
Find the column index of the largest entry in each row of the label matrix G.
```

Graph Construction The traditional KNN-based way to construct graph is time consuming. A normalized step is also needed to insure Proposition 1. Here we adopt an efficient way to compute a doubly-stochastic similarity matrix.

Given data matrix $X = [x_1, \ldots, x_n] \in \mathbb{R}^{d\times n}$, and the subset $X_{sub} = [u_1, \ldots, u_m] \in \mathbb{R}^{d\times m}$ in which each point plays a role as an anchor. According to [Liu *et al.*, 2010], we can design a sparse regression matrix $Z \in \mathbb{R}^{n\times m}$ as follows:

$$Z_{ij} = \frac{K_h(x_i, u_j)}{\sum_{j'\in\langle i\rangle} K_h(x_i, u_{j'})}, \ (i = 1, \ldots, n; j \in \langle i\rangle) \tag{12}$$

where $\langle i\rangle$ is the set containing the indexes of s nearest anchors of x_i. We choose $K_h(\cdot)$ as the common used Gaussian kernel with self-tuning bandwidth h [Chen *et al.*, 2011]. The doubly-stochastic similarity matrix is then computed by $W = Z\Sigma^{-1}Z^T$, where Σ is a diagonal matrix with the entry $\Sigma_{kk} = \sum_{i=1}^n Z_{ik}$.

Important tips: We do not need to calculate W explicitly since it would be both time and memory consuming. We can utilize the low-rank matrix multiplication to speed up. Concretely, we can compute (WF) and (WG) by $Z\left(\Sigma^{-1}(Z^TF)\right)$ and $Z\left(\Sigma^{-1}(Z^TG)\right)$, respectively.

Table 1: Summary of Time Complexity of Different Methods

Method	Initialization	Construction	Solving & Labeling
Nyström	$O(1)$	$O(mnd)$	$O(m^2n + mnk + nk^2t_0)$
KNN-SC	—	$O(n^2d)$	$O(n^2d + nk^2t_0)$
LSC	$O(E)$	$O(mnd)$	$O(C + nk^2t_0)$
SSSC	$O(E)$	$O(A)$	$O(m^2d + mk^2t_0 + B)$
ONGR	$O(E + C)$	$O(mnd)$	$O((mnk + m^2k + D)t_2)$

t_0: #iterations in *kmeans*, t_H: #iterations in Homotopy optimizer
$A = (m^2d^2 + m^3d)t_H$, $B = md^2 + m^2n$, $C = m^2n + m^3$
$D = nk^2 + k^3$, $E = mndt_1$

Initialization The *indicator matrix* F can be randomly initialized with orthogonal constraint. An alternative way is to use the k eigenvectors corresponding to the k smallest eigenvalues of $\tilde{L}$, or the k eigenvectors corresponding to the k largest eigenvalues of W.

Let $\hat{Z} = Z\Sigma^{-1/2}$, we have $W = \hat{Z}\hat{Z}^T$. Supposing the SVD of $\hat{Z}$ is $\hat{Z} = U_1\Lambda_1V_1^T$, it's easy to find that the left singular vectors of $\hat{Z}$ are the eigenvectors of W. Therefore, we can initialize F by taking the largest k singular vectors in U_1 as columns.

3.3 Computational Complexity

Suppose the number of anchor points is m. It takes $O(mndt_1)$ to find the anchors and takes $O(m^2n + m^3)$ to initialize F by using left singular vectors of $\hat{Z}$, where t_1 denotes the number of iterations in *kmeans*. The construction of Z needs $O(mnd)$ while the alternating and iterative procedure needs $O\left((mnk + m^2k + nk^2 + k^3)t_2\right)$. t_2 denotes the number of iterations, and $O(nk^2 + k^3)$ is the cost of SVD of M. For large scale problem, there usually exist $m > k$ and $n \gg k, m$. Thus the total complexity is about $O(mndt_1 + m^2n + mnkt_2)$ that scales linearly with the data size n. See Table 1 for a brief summary of time complexity of different methods (references can be seen in the section of comparison algorithms).

Except for KNN-SC that costs too much on constructing the similarity matrix, all other methods are linear time complexity while they all depend on the post-processing step. SSSC is more sensitive to the dimensionality d, which makes it unsuitable to high dimensional data such as images. The total complexities of Nyström and LSC are $O(mnd + m^2n + nk^2t_0)$, $O(mndt_1 + m^2n + nk^2t_0)$, respectively.

3.4 Discussions

Our proposed model has similar formulation with non-negative matrix factorization (NMF) [Lee and Seung, 2001; Li and Ding, 2006; Ding *et al.*, 2005]. In this subsection, we compare ONGR with NMF methods and show the essential differences between them.

Given a nonnegative matrix $X \in \mathbb{R}^{d\times n}$, and a reduced rank k, the problem of NMF is formulated as follows:

$$\min_{\tilde{F}\geq 0, \tilde{G}\geq 0} \|X - \tilde{F}\tilde{G}^T\|_F^2, \tag{13}$$

where $\tilde{F} \in \mathbb{R}^{d\times k}$, $\tilde{G} \in \mathbb{R}^{n\times k}$. Due to its nonnegativity, NMF provides interpretability that the cluster labels can be obtained by finding the index of the largest entry in each row of $\tilde{G}$.

Compared to our models in Eq.(6) and Eq.(7) that are based on graph reconstruction, NMF in Eq.(13) is based on data reconstruction and requires the input data to be nonnegative. In many cases, the data points lies on a nonlinear manifold. It is more suitable to reconstruct the graph which describes the relationship between data points, rather than reconstructing the data matrix directly.

Recently, a nonnegative symmetric factorization (SymNMF) [Kuang *et al.*, 2012] of the similarity matrix is proposed as follows:

$$\min_{\tilde{H}\geq 0} \|A - \tilde{H}\tilde{H}^T\|_F^2, \tag{14}$$

where $\tilde{H} \in \mathbb{R}^{n\times k}$, and $A \in \mathbb{R}^{n\times n}$ can be any symmetric matrix representing similarity values. Again, due to the nonnegativity, the column index of the largest entry in each row of $\tilde{H}$ indicates the cluster label.

Comparing our model in Eq.(6), the normalized spectral clustering in Eq.(3) and SymNMF in Eq.(14), it is easy to find they share the common objective, while the constraints are totally different. Specifically, our model in Eq.(6) is a combination of the other two by introducing the orthogonal and nonnegative constraints simultaneously, thus the reconstructed graph has more clear structure. Our new model in Eq.(7) naturally inherits the good property for clustering, while it has low complexity and can be applied to large scale clustering.

There are many other NMF methods, such as Semi-NMF, Convex-NMF and Tri-factorization. They have different constraints or assumptions. These NMF methods are based on matrix multiplication updating rules and they often take a long time to converge. To seek a trade-off between convergence rate and computational cost, SymNMF developes a Newton-like algorithm that still has complexity of $O(n^3k)$ in each iteration.

4 Experiments

4.1 Datasets and Evaluation Metrics

Datasets A variety of datasets are adopted to evaluate the proposed method. They can be downloaded from the UCI Machine Learning Repository [1], the LibSVM Data page [2], and three webpages [3] [4] [5]. Most of these datasets have more than 10,000 samples.

We obtained the MINIST-extend dataset by translating the original images in MINIST by one pixel in each direction like [Liu *et al.*, 2010]. We scaled CoverType to $[0, 1]$ by feature. The letter symbols in Connect-4 were replaced by digits. For USPS, COIL-20, COIL-100, MINIST and its extended version, we firstly scaled them to $[0, 1]$, then we conducted dimensionality reduction by PCA with 95% information reserved, except for MINIST-extend which reserves 90% information. Details can be seen in Table 2.

[1] https://archive.ics.uci.edu/ml/datasets.html
[2] https://www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets/
[3] http://alumni.cs.ucsb.edu/ wychen/sc.html
[4] http://yann.lecun.com/exdb/mnist/index.html
[5] http://www.cs.columbia.edu/CAVE/software/

Table 2: Description of Datasets

dataset	#samples	#Dim.	#Classes
USPS	9,298	35	10
PenDigits	10,992	16	10
MINIST	70,000	154	10
COIL-20	1,440	84	20
COIL-100	7,200	211	100
Connect-4	67,557	42	3
Seismic	98,528	50	3
RCV1	193,844	47,236	103
CoverType	581,012	54	7
MINIST-extend	630,000	93	10

Evaluation Metrics The clustering quality is measured by *Normalized Mutual Information* (*NMI*) and *Accuracy* (*Acc*) [Cai *et al.*, 2005]. The values of *NMI* and *Acc* range from 0 to 1 with higher score corresponding to better performance. Note that the clustering accuracy is the average performance of label matching results between ground truth labels and predicted labels, which is different from the classification accuracy.

4.2 Comparison Algorithms

We provide two versions of ONGR in which ONGR-K uses *kmeans* to find anchors and ONGR-R adopts random selection. The comparison algorithms are (1) Nyström [Fowlkes *et al.*, 2004]: Spectral clustering that uses Nyström method with orthogonalization. (2) KNN-SC [Chen *et al.*, 2011]: Spectral clustering that uses KNN to construct similarity matrix. (3) LSC [Cai and Chen, 2014]: LSC-K uses *kmeans* for landmark selection, and LSC-R randomly selects landmarks. (4) SSSC [Peng *et al.*, 2013]: Scalable sparse subspace clustering. Due to the requirement for nonnegativity of data matrix or high computational cost, NMF methods are not considered as the comparison methods.

4.3 Experimental Setting

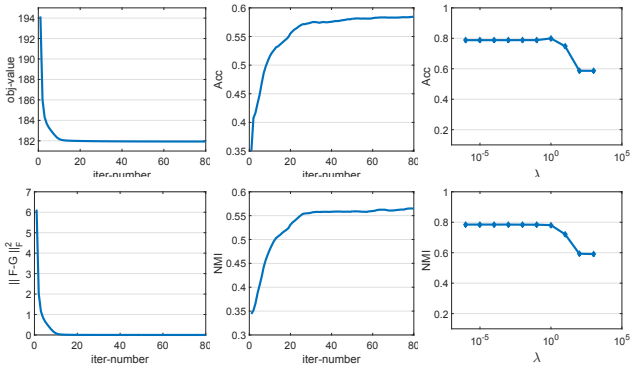
There are three parameters in ONGR, namely m, s and λ. LSC needs to tune parameters p, r. Nyström has parameter n_1, and KNN-SC has parameter k. The three parameters n_2, λ_1 and δ are from SSSC.

For fair comparison, we took the same *kmeans* centroids as anchors or landmarks in ONGR-K and LSC-K, and also took the same random selection points in ONGR-R, LSC-R, Nyström and SSSC. For m, p, n_1, n_2, we searched their value in the range of [100,1200] with step size 100, while searching s and r in the range of [2,8] with step size 1. For KNN-SC, we chose k in the range of [5,20] with step size 5. As suggested by the authors of SSSC, we searched λ_1 and δ among $\{10^{-7}, 10^{-6}, 10^{-5}\}$ and $\{10^{-3}, 10^{-2}, 10^{-1}\}$, respectively. For our trade-off parameter λ, we searched $\log \lambda$ in the range of [-6,3] with step size 1.

We ran each algorithm except for ONGR for 20 times under each parameter setting. Note that given the pre-selected points, ONGR is the only method that has stable performance. For the last three datasets in Table 2, we did not test ONGR-K and LSC-K since *kmeans* becomes costly for finding anchors or landmarks. We also did not test KNN-SC due to the unbearable running time.

Table 3: Running time (s)

Dataset	KNN-SC	Nyström	SSSC	LSC-R	ONGR-R
USPS	9.65	5.84	89.76	2.19	**1.40**
PenDigits	28.04	33.03	110.98	21.67	**19.29**
MINIST	1401.41	40.88	217.68	**31.95**	39.40
CoverType	–	**168.55**	463.22	235.61	202.46
MINIST-extend	–	178.24	1095.78	166.55	**147.41**



(a) COIL-20 (b) MINIST-extend (c) USPS

Figure 2: (a) curves of the objective value of Eq.(7) and the squared Frobenius norm of $(F - G)$ (b) curves of clustering performance (c) effect of parameter λ

Convergence judgement: With data growing in scale, it is a heavy burden for us to compute the objective value of Eq.(7). Recall that we do not calculate W explicitly. We adopted another criterion, in which the $(i+1)$-th iteration is terminated when just a very small percentage of data points change their predicted labels comparing to the i-th iteration. Empirically, we set the threshold to be 0.001.

4.4 Experimental Results

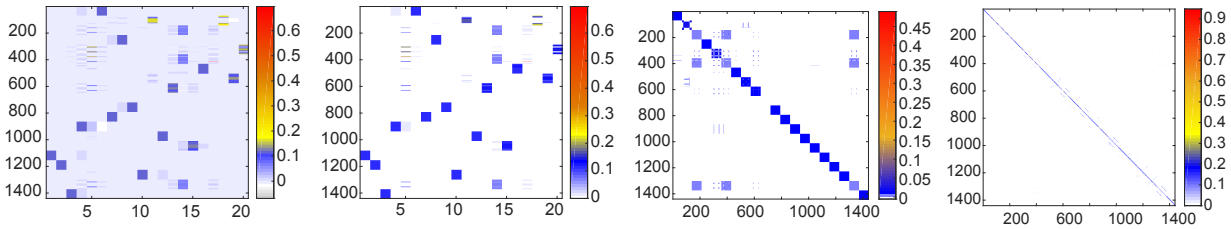
Table 3 records the running time corresponding to the best performance on five datasets. The results are consistent with the complexity analysis. Table 4 reports the clustering performance along with the standard deviation (std), and we can see that ONGR achieves the best or second best results no matter what the metric is. The average results of each algorithm on all datasets is also reported, showing the superiority of ONGR more clearly.

From Table 3 and Table 4, it can be concluded that ONGR achieves stable and much better performance in the shortest or relatively less time. Specifically, within about 20 and 200 seconds, ONGR-R gains 8.14% and 9.24% increment of accuracy over the second best results on PenDigits and CoverType, respectively. Comparing LSC and ONGR, the two methods with the common graph construction, we see that ONGR-R exceeds 5.98% than LSC-R and ONGR-K exceeds 7.41% than LSC-K with respect to accuracy, which shows the advantage of removing post-processing.

To verify the convergence of the proposed method, in the upper part of Figure 2(a), we plotted the objective value of Eq.(7) on the relatively small dataset COIL-20. The curve of the squared Frobenius norm of $(F - G)$ demonstrates that F

Table 4: Clustering Performance (% $\pm std$)

Metric	Dataset	KNN-SC	Nyström	SSSC	LSC-R	LSC-K	ONGR-R	ONGR-K
Acc	USPS	66.84 ± 3.05	69.52 ± 2.13	53.85 ± 0.69	75.67 ± 5.05	77.00 ± 7.20	78.82	**80.59**
	PenDigits	64.15 ± 0.15	72.33 ± 2.49	74.92 ± 0.00	79.16 ± 3.21	79.97 ± 6.11	87.30	**88.02**
	MINIST	68.72 ± 0.03	55.38 ± 3.13	53.01 ± 0.35	69.82 ± 5.23	76.21 ± 6.20	70.75	**78.59**
	COIL-20	82.22 ± 0.00	63.50 ± 3.00	61.38 ± 1.74	71.19 ± 4.79	72.89 ± 6.66	87.08	**87.92**
	COIL-100	59.81 ± 0.49	46.66 ± 1.54	43.94 ± 1.29	51.60 ± 1.59	57.45 ± 2.59	54.60	**67.13**
	Connect-4	42.68 ± 0.15	36.43 ± 0.05	$\mathbf{65.82 \pm 0.00}$	40.79 ± 2.80	40.03 ± 2.82	55.57	52.61
	Seismic	67.69 ± 0.01	67.21 ± 0.00	66.52 ± 0.00	67.58 ± 0.44	67.81 ± 0.12	**68.54**	68.42
	RCV1	–	16.94 ± 0.72	14.22 ± 0.00	16.47 ± 0.38	–	**17.49**	–
	CoverType	–	27.00 ± 1.06	44.06 ± 0.00	41.87 ± 2.01	–	**53.30**	–
	MINIST-extend	–	47.25 ± 2.47	55.74 ± 0.00	58.72 ± 5.09	–	**59.26**	–
	mean	(64.59 ± 0.55)	50.22 ± 1.66	53.35 ± 0.41	57.29 ± 3.06	(67.34 ± 4.53)	**63.27**	(**74.75**)
NMI	USPS	80.45 ± 1.31	65.19 ± 0.93	55.93 ± 0.56	77.48 ± 2.86	80.64 ± 2.34	78.48	**82.76**
	PenDigits	78.93 ± 1.27	66.65 ± 1.09	73.51 ± 0.00	79.84 ± 2.26	81.85 ± 2.74	83.50	**84.42**
	MINIST	76.60 ± 0.07	48.04 ± 1.27	53.55 ± 0.11	66.73 ± 2.29	77.33 ± 2.36	69.05	**79.50**
	COIL-20	91.15 ± 0.00	76.50 ± 1.39	78.09 ± 1.15	90.31 ± 2.89	90.90 ± 2.37	95.18	**96.35**
	COIL-100	83.80 ± 0.17	76.15 ± 0.58	69.11 ± 0.48	77.29 ± 0.53	82.96 ± 0.67	79.27	**88.16**
	Connect-4	0.18 ± 0.00	0.24 ± 0.01	0.24 ± 0.00	0.25 ± 0.09	0.22 ± 0.10	**0.58**	0.32
	Seismic	27.60 ± 0.02	27.52 ± 0.01	25.12 ± 0.00	29.85 ± 0.45	29.93 ± 0.83	31.90	**32.20**
	RCV1	–	$\mathbf{25.81 \pm 0.27}$	17.85 ± 0.00	23.65 ± 0.21	–	24.19	–
	CoverType	–	13.94 ± 0.00	20.58 ± 0.00	19.56 ± 0.84	–	**21.05**	–
	MINIST-extend	–	36.22 ± 0.88	54.75 ± 0.00	55.51 ± 1.61	–	**56.39**	–
	mean	(62.67 ± 0.41)	43.63 ± 0.64	44.87 ± 0.23	52.05 ± 1.40	(63.40 ± 1.63)	**53.96**	(**66.24**)



(a) indicator matrix F (b) label matrix G (c) reconstructed graph FG^T (d) constructed graph W

Figure 3: A practical illustration of F, G, FG^T, and W in Eq.(7). We have reorganized the rows of F and G (rows and columns of W) to place samples with the same predicted (ture) cluster label continuously.

and G do get close to each other in each iteration. Thus the model in Eq.(7) gets closer and closer to the one in Eq.(6), and the inheritance of good clustering quality from Eq.(6) to Eq.(7) is guaranteed. As can be seen, ONGR does converge fast within about 30 iterations.

In Figure 2(b), we plotted the curves of clustering performance on MINIST-extend. It can be seen that the performance is getting better as the iteration number increases, showing the effectiveness of the optimization process. Note that in order to completely present the curves, we set the maximum iteration number to be 80 and did not terminate the iteration.

As λ is the only parameter introduced by the model in Eq.(7), in Figure 2(c), we examined the effect of λ to clustering performance on USPS. Parameters m and s were set to be 800 and 4, respectively. Under the condition, we see that ONGR is pretty robust to λ.

Moreover, in Figure 3, we tested ONGR-R on COIL-20 to give a practical illustration of graph reconstruction. As we can see, compared to the constructed graph, the reconstructed graph has clear structure which contains more accurate information about the clusters. However, the zero rows in the *label matrix* lead to the missing block in the reconstructed graph, *i.e.*, zeros rows will destroy the structure. This is a topic that we will explore to further improve the clustering performance.

5 Conclusion

In this paper, we have proposed a novel approach called ONGR for large scale clustering, which is based on the viewpoint of graph reconstruction. With orthogonal and nonnegative constraints, the reconstructed graph naturally has clear structure about the clusters. Due to the nonnegativity, the interpretability is provided and the post-processing is no longer needed. Given the anchors, ONGR has stable and much better performance than other state-of-the-art methods, demonstrated by extensive experiments.

In the future, we plan to study the sparsity of *label matrix* to avoid zero rows. We are also going to propose a robust version to better deal with outliers. The ℓ_1-norm and $\ell_{2,1}$-norm may be considered to achieve robustness and sparsity.

Acknowledgments

This work was supported in part by the National Science Foundation of China under Grants 61522207 and 61473231.

References

[Cai and Chen, 2014] Deng Cai and Xinlei Chen. Large scale spectral clustering via landmark-based sparse representation. *IEEE Transactions on Cybernetics*, 45(8):1669–1680, 2014.

[Cai *et al.*, 2005] Deng Cai, Xiaofei He, and Jiawei Han. Document clustering using locality preserving indexing. *TKDE*, 17(12):1624–1637, 2005.

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

[Choromanska *et al.*, 2013] Anna Choromanska, Tony Jebara, Hyungtae Kim, Mahesh Mohan, and Claire Monteleoni. Fast spectral clustering via the nyström method. In *ALT*, pages 367–381. Springer, 2013.

[Ding *et al.*, 2005] Chris HQ Ding, Xiaofeng He, and Horst D Simon. On the equivalence of nonnegative matrix factorization and spectral clustering. In *SDM*, volume 5, pages 606–610. SIAM, 2005.

[Fowlkes *et al.*, 2004] Charless Fowlkes, Serge Belongie, Fan Chung, and Jitendra Malik. Spectral grouping using the nystrom method. *TPAMI*, 26(2):214–225, 2004.

[Hagen and Kahng, 1992] Lars Hagen and Andrew B Kahng. New spectral methods for ratio cut partitioning and clustering. *TCAD*, 11(9):1074–1085, 1992.

[Kuang *et al.*, 2012] Da Kuang, Chris Ding, and Haesun Park. Symmetric nonnegative matrix factorization for graph clustering. In *SDM*, pages 106–117. SIAM, 2012.

[Lee and Seung, 2001] Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *NIPS*, pages 556–562. MIT, 2001.

[Li and Ding, 2006] Tao Li and Chris Ding. The relationships among various nonnegative matrix factorization methods for clustering. In *ICDM*, pages 362–371. IEEE, 2006.

[Li *et al.*, 2011] Mu Li, Xiao-Chen Lian, James T Kwok, and Bao-Liang Lu. Time and space efficient spectral clustering via column sampling. In *CVPR*, pages 2297–2304. IEEE, 2011.

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

[Liu *et al.*, 2013] Jialu Liu, Chi Wang, Marina Danilevsky, and Jiawei Han. Large-scale spectral clustering on graphs. In *IJCAI*, pages 1486–1492. Morgan Kaufmann, 2013.

[Ng *et al.*, 2002] Andrew Y Ng, Michael I Jordan, Yair Weiss, et al. On spectral clustering: Analysis and an algorithm. In *NIPS*, volume 2, pages 849–856. MIT, 2002.

[Nie *et al.*, 2014] Feiping Nie, Xiaoqian Wang, and Heng Huang. Clustering and projected clustering with adaptive neighbors. In *SIGKDD*, pages 977–986. ACM, 2014.

[Peng *et al.*, 2013] Xi Peng, Lei Zhang, and Zhang Yi. Scalable sparse subspace clustering. In *CVPR*, pages 430–437. IEEE, 2013.

[Schaeffer, 2007] Satu Elisa Schaeffer. Graph clustering. *Computer Science Review*, 1(1):27–64, 2007.

[Shi and Malik, 2000] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *TPAMI*, 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 *LREC*, pages 201–204, 2008.

[Spielman and Teng, 2011] Daniel A Spielman and Shang-Hua Teng. Spectral sparsification of graphs. *SIAM Journal on Computing*, 40(4):981–1025, 2011.

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

[Wang *et al.*, 2010] Fei Wang, Ping Li, and Arnd Christian Konig. Learning a bi-stochastic data similarity matrix. In *ICDM*, pages 551–560. IEEE, 2010.

[Wang *et al.*, 2016] Xiaoqian Wang, Feiping Nie, and Heng Huang. Structured doubly stochastic matrix for graph based clustering. In *SIGKDD*, pages 1245–1254. ACM, 2016.

[Yan *et al.*, 2009] Donghui Yan, Ling Huang, and Michael I Jordan. Fast approximate spectral clustering. In *SIGKDD*, pages 907–916. ACM, 2009.

[Zass and Shashua, 2006] Ron Zass and Amnon Shashua. Doubly stochastic normalization for spectral clustering. In *NIPS*, pages 1569–1576. MIT, 2006.

[Zhang *et al.*, 2014] Yan-Ming Zhang, Kaizhu Huang, Xinwen Hou, and Cheng-Lin Liu. Learning locality preserving graph from data. *IEEE Transactions on Cybernetics*, 44(11):2088–2098, 2014.