Fast Proximal Gradient Descent for Support Regularized Sparse Graph

Abstract

Sparse graphs built by sparse representation has been demonstrated to be effective in clustering high-dimensional data. Albeit the compelling empirical performance, the vanilla sparse graph ignores the geometric information of the data by performing sparse representation for each datum separately. In order to obtain a sparse graph aligned with the local geometric structure of data, we propose a novel Support Regularized Sparse Graph, abbreviated as SRSG, for data clustering. SRSG encourages local smoothness on the neighborhoods of nearby data points by a well-defined support regularization term. We propose a fast proximal gradient descent method to solve the non-convex optimization problem of SRSG with the convergence matching the Nesterov's optimal convergence rate of first-order methods on smooth and convex objective function with Lipschitz continuous gradient. Extensive experimental results on various real data sets demonstrate the superiority of SRSG over other competing clustering methods.

1 INTRODUCTION

Clustering methods based on pairwise similarity, such as K-means [Duda et al., 2000], Spectral Clustering [Ng et al., 2001] and Affinity Propagation [Frey and Dueck, 2007], segment the data in accordance with certain similarity measure between data points. The performance of similarity-based clustering largely depends on the similarity measure. Among various similarity-based clustering methods, graph-based methods [Schaeffer, 2007] are promising which often model data points and data similarity as vertices and edge weight of the graph respectively. Sparse graphs, where only a few edges of nonzero weights are associated with each vertex, are effective in clustering high-dimensional data. Existing works, such as Sparse Subspace Clustering (SSC) [Elhamifar and Vidal, 2013] and ℓ^1-graph [Yan and Wang, 2009, Cheng et al., 2010], build sparse graphs by sparse representation of each point. In these sparse graphs, the vertices represent the data points, an edge is between two vertices whenever one contributes to the spare representation of the other, and the weight of the edge is determined by the associated sparse codes. A theoretical explanation is provided by SSC, which shows that such sparse representation recovers the underlying subspaces from which the data are drawn under certain assumptions on the data distribution and angles between subspaces. When such assumptions hold, data belonging to different subspaces are disconnected in the sparse graph. A sparse similarity matrix is then obtained as the weighted adjacency matrix of the constructed sparse graph by ℓ^1-graph or SSC, and spectral clustering is performed on the sparse similarity matrix to obtain the data clusters. In the sequel, we refer to ℓ^1-graph and SSC as vanilla sparse graph. Vanilla sparse graph has been shown to be robust to noise and capable of producing superior results for high-dimensional data, compared to spectral clustering on the similarity produced by the widely used Gaussian kernel.

Albeit compelling performance for clustering, vanilla sparse graph is built by performing sparse representation for each data point independently without considering the geometric information of the data. High dimensional data always lie in low dimensional submanifold. The Manifold assumption [Belkin et al., 2006] has been employed in the sparse graph literature with an effort in learning a sparse graph aligned with the local geometric structure of the data. For example, Laplacian Regularized ℓ^1-graph (LR-ℓ^1-graph) is proposed in [Yang et al., 2014a,b] to obtain locally smooth sparse codes in the sense of ℓ^2-distance so as to improve the performance of vanilla sparse graph. The locally smooth sparse codes in LR-ℓ^1-graph is achieved by penalizing large ℓ^2-distance between the sparse codes of two nearby data points. However, the locally smooth sparse codes in the sense of ℓ^2-distance does not encode the local geometric structure of the

Submitted to the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023). **To be used for reviewing only**.

data into the construction of sparse graph. Intuitively, it is expected that nearby data points, or vertices, in a data manifold could exhibit locally smooth neighborhood according to the geometric structure of the data. Namely, nearby points are expected to have similar neighbors in the constructed sparse graph.

The support of the sparse code of a data point determines the neighbors it selects, and the nonzero elements of the sparse code contribute to the corresponding edge weights. This indicates that ℓ^2-distance is not a suitable distance measure for sparse codes in our setting, and one can easily imagine that two sparse codes can have very small ℓ^2-distance while their supports are quite different, meaning that they choose different neighbors. Motivated by the manifold assumption on the local sparse graph structure, we propose a novel Support Regularized Sparse Graph, abbreviated as SRSG, which encourages smooth local neighborhood in the sparse graph. The smooth local neighborhood is achieved by a well-defined support distance between sparse codes of nearby points in a data manifold.

The contributions of this paper are as follows. First, we propose Support Regularized Sparse Graph (SRSG) which is capable of learning a sparse graph with its local neighborhood structure aligned to the local geometric structure of the data manifold. Secondly, we propose an efficient and provable optimization algorithm, Fast Proximal Gradient Descent with Support Projection (FPGD-SP), to solve the non-convex optimization problem of SRSG. Albeit the non-smoothness and nonconvexity of the optimization problem of SRSG, FPGD-SP still achieves Nesterov's optimal convergence rate of first-order methods on smooth and convex problems with Lipsschitz continuous gradient.

The remaining parts of the paper are organized as follows. Vanilla sparse graph (ℓ^1-graph) and LR-ℓ^1-graph are introduced in the next section, and then the detailed formulation of SRSG is illustrated. We then show the clustering performance of SRSG, and conclude the paper. Throughout this paper, bold letters are used to denote matrices and vectors, and regular lower letter is used to denote scalars. Bold letter with superscript indicates the corresponding column of a matrix, e.g. $\mathbf{Z}^i$ indicates the i-th column of the matrix $\mathbf{Z}$, and the bold letter with subscript indicates the corresponding element of a matrix or vector. $\sigma_{\max}(\cdot)$ and $\sigma_{\min}(\cdot)$ indicate the maximum and minimum singular value of a matrix. $\|\cdot\|_F$ and $\|\cdot\|_p$ denote the Frobenius norm and the ℓ^p-norm, and $\mathrm{diag}(\cdot)$ indicates the diagonal elements of a matrix. $\mathrm{supp}(\mathbf{v})$ denotes the support of a vector $\mathbf{v}$, which is the set of indices of nonzero elements of $\mathbf{v}$.

2 PRELIMINARIES: VANILLA SPARSE GRAPH AND ITS LAPLACIAN REGULARIZATION

Vanilla sparse graph based methods [Yan and Wang, 2009, Cheng et al., 2010, Elhamifar and Vidal, 2009, 2013, Wang and Xu, 2013, Soltanolkotabi et al., 2014] apply the idea of sparse coding to build sparse graphs for data clustering. Given the data $\boldsymbol{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{d\times n}$, robust version of vanilla sparse graph first solves the following optimization problem for some weighting parameter $\lambda_{\ell^1} > 0$ to obtain the sparse representation for each data point $\mathbf{x}_i$:

$$\min_{\mathbf{Z}^i\in\mathbb{R}^n, \mathbf{Z}^i_i=0} \|\mathbf{x}_i - \boldsymbol{X}\mathbf{Z}^i\|_2^2 + \lambda_{\ell^1}\|\mathbf{Z}^i\|_1, \quad i \in [n], \tag{1}$$

then construct a coefficient matrix $\mathbf{Z} = [\mathbf{Z}^1, \ldots, \mathbf{Z}^n] \in \mathbb{R}^{n\times n}$ with element $\mathbf{Z}_{ij} = \mathbf{Z}^j_i$, where $\mathbf{Z}^i$ is the i-th column of $\mathbf{Z}$. The diagonal elements of $\mathbf{Z}$ are zero so as to avoid trivial solution $\mathbf{Z} = \mathbf{I}_n$ where $\mathbf{I}_n$ is a $n \times n$ identity matrix.

A vanilla sparse graph $G = (\boldsymbol{X}, \mathbf{W})$ is then built where $\boldsymbol{X}$ is the set of vertices, $\mathbf{W}$ is the weighted adjacency matrix of G. The edge weight $\mathbf{W}_{ij}$ is set by the sparse codes according to

$$\mathbf{W}_{ij} = (|\mathbf{Z}_{ij}| + |\mathbf{Z}_{ji}|)/2, \quad 1 \le i, j \le n. \tag{2}$$

Finally, the data clusters are obtained by performing spectral clustering on the vanilla sparse graph G with sparse similarity matrix $\mathbf{W}$. In SSC and its geometric analysis [Elhamifar and Vidal, 2009, 2013, Soltanolkotabi et al., 2014], it is proved that the sparse representation (1) for each datum recovers the underlying subspaces from which the data are generated when the subspaces satisfy certain geometric properties in terms of the principle angle between different subspaces. When these required assumptions hold, data belonging to different subspaces are disconnected in the sparse graph, leading to the success of the subspace clustering. In practice, however, one can often empirically try the same formulation to obtain satisfactory results even without checking the assumptions.

High dimensional data often lie on or close to a submanifold of low intrinsic dimension, and existing clustering methods benefit from learning data representation aligned to its underlying manifold structure. While vanilla sparse graph demonstrates better performance than many traditional similarity-based clustering methods, it performs sparse representation for each datum independently without considering the geometric information and manifold structure of the entire data. On the other hand, in order to obtain the data embedding that accounts for the geometric information and manifold structure of the data, the manifold assumption [Belkin et al., 2006] is usually employed [Liu et al., 2010, He et al., 2011, Zheng et al., 2011, Gao et al., 2013].

3 SUPPORT REGULARIZED SPARSE GRAPH

In this section, we propose Support Regularized Sparse Graph (SRSG) which learns locally smooth neighborhoods in the sparse graph by virtue of locally consistent support of the sparse codes. Instead of imposing smoothness in the sense of ℓ^2-distance on the sparse codes in the existing LR-ℓ^1-graph, SRSG encourages locally smooth neighborhoods so as to capture the local manifold structure of the data in the construction of the sparse graph. A side effect of locally smooth neighborhoods is robustness to noise or outliers by encouraging nearby points on the manifold to choose similar neighbors in the sparse graph. Note that ℓ^2-distance based graph regularization cannot enjoy this benefit since small ℓ^2-distance between the sparse codes of nearby data points does not guarantee their consistent neighborhood in the sparse graph. The optimization problem of SRSG is

$$\min_{\mathbf{Z}\in\mathbb{R}^{n\times n},\mathrm{diag}(\mathbf{Z})=\mathbf{0}} L(\mathbf{Z}) = \sum_{i=1}^{n} \|\mathbf{x}_i - \boldsymbol{X}\mathbf{Z}^i\|_2^2 + \gamma \mathbf{R}_{\mathbf{S}}(\mathbf{Z}), \quad (3)$$

where $\mathbf{R}_{\mathbf{S}}(\mathbf{Z}) = \frac{1}{2}\sum_{i,j=1}^{n} \mathbf{S}_{ij} d(\mathbf{Z}^i, \mathbf{Z}^j)$ is the support regularization term, $\mathbf{S}$ is the adjacency matrix of the KNN graph, $\gamma > 0$ is the weighting parameter for support regularization term. Each data point $\mathbf{x}_i$ is normalized to have unit ℓ^2-norm. $d(\mathbf{Z}^i, \mathbf{Z}^j)$ is the support distance between two sparse codes $\mathbf{Z}^i$ and $\mathbf{Z}^j$ which is defined as

$$d(\mathbf{Z}^i, \mathbf{Z}^j) = \sum_{1\le m\le n, m\neq i,j} (\mathbb{1}_{\mathbf{Z}^i_m=0, \mathbf{Z}^j_m\neq 0} + \mathbb{1}_{\mathbf{Z}^i_m\neq 0, \mathbf{Z}^j_m=0}). \quad (4)$$

SRSG encourages nearby data points to have similar neighborhoods by penalizing large support distance between every pair of nearby points in a data manifold. It can be seen from (4) that a small support distance between $\mathbf{Z}^i$ and $\mathbf{Z}^j$ indicates that the indices of their nonzero elements are mostly the same. By the construction of sparse graph (2), it indicates that the two points $\mathbf{x}_i$ and $\mathbf{x}_j$ choose similar neighbors. Figure 1 further illustrates the effect of support regularization.

We use coordinate descent to optimize (3) with respect to $\mathbf{Z}^i$, i.e. in each step the i-th column of $\mathbf{Z}$, while fixing all the other sparse codes $\{\mathbf{Z}^j\}_{j\neq i}$. In each step of coordinate descent, the optimization problem for $\mathbf{Z}^i$ is

$$\min_{\mathbf{Z}^i\in\mathbb{R}^n, \mathbf{Z}^i_i=0} F(\mathbf{Z}^i) = \|\mathbf{x}_i - \boldsymbol{X}\mathbf{Z}^i\|_2^2 + \gamma \mathbf{R}_{\mathbf{S}}(\mathbf{Z}^i), \quad (5)$$

where $\mathbf{R}_{\mathbf{S}}(\mathbf{Z}^i) = \sum_{j=1}^{n} \mathbf{S}_{ij} d(\mathbf{Z}^i, \mathbf{Z}^j)$. Note that $\mathbf{R}_{\mathbf{S}}(\mathbf{Z}^i)$ can be written as $\mathbf{R}_{\mathbf{S}}(\mathbf{Z}^i) = \sum_{k=1}^{n} c_{ki} \mathbb{1}_{\mathbf{Z}^i_k\neq 0}$ up to a constant, where $c_{ki} = \sum_{j=1}^{n} \mathbf{S}_{ij} \mathbb{1}_{\mathbf{Z}^j_k=0} - \sum_{j=1}^{n} \mathbf{S}_{ij} \mathbb{1}_{\mathbf{Z}^j_k\neq 0}$.

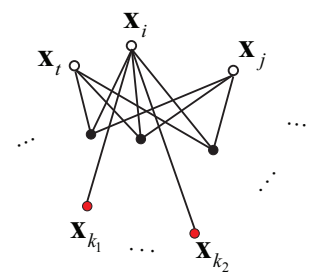

Figure 1: During the construction of support regularized sparse graph, point $\mathbf{x}_i$ is among the K nearest neighbors of $\mathbf{x}_t$ and $\mathbf{x}_j$. $\mathbf{Z}^t$ and $\mathbf{Z}^j$ have the same support denoted by the three black dots ($\mathbf{x}_{k_1}$, $\mathbf{x}_{k_2}$ and $\mathbf{x}_{k_3}$), suggesting the correct neighbors of $\mathbf{x}_i$. By penalizing support distance between nearby points, $\mathbf{x}_i$ is encouraged to choose the three black dots as neighbors in the sparse graph while discarding the wrong neighbors marked in red.

It should be emphasized that SRSG does not use the ℓ^1-norm, i.e. $\|\mathbf{Z}^i\|_1$ to impose sparsity on $\mathbf{Z}^i$. Instead, as will be illustrated in Section 3.2, our proposed fast PGD method always finds a sparse solution efficiently. (5) is equivalent to

$$\min_{\mathbf{Z}^i\in\mathbb{R}^n, \mathbf{Z}^i_i=0} F(\mathbf{Z}^i) = \|\mathbf{x}_i - \boldsymbol{X}\mathbf{Z}^i\|_2^2 + \gamma \sum_{k=1}^{n} c_{ki} \mathbb{1}_{\mathbf{Z}^i_k\neq 0}. \quad (6)$$

Problem (6) is non-convex due to the non-convex regularization term $\sum_{k=1}^{n} c_{ki} \mathbb{1}_{\mathbf{Z}^i_k\neq 0}$, and an optimization algorithm is supposed to find a critical point for this problem. The definition of critical point for general Frechet subdifferentiable functions is defined as follows.

Definition 1. (Subdifferential and critical points [Rockafellar and Wets, 2009]) Given a non-convex function $f\colon \mathbb{R}^d \to \mathbb{R}\cup\{+\infty\}$ which is a proper and lower semi-continuous function,

- for a given $\mathbf{x} \in \mathrm{dom} f$, its Frechet subdifferential of f at $\mathbf{x}$, denoted by $\tilde{\partial} f(\mathbf{x})$, is the set of all vectors $\mathbf{u} \in \mathbb{R}^d$ which satisfy

$$\liminf_{\mathbf{y}\neq\mathbf{x}, \mathbf{y}\to\mathbf{x}} \frac{f(\mathbf{y}) - f(\mathbf{x}) - \langle \mathbf{u}, \mathbf{y}-\mathbf{x}\rangle}{\|\mathbf{y}-\mathbf{x}\|} \ge 0.$$

- The limiting-subdifferential of f at $\mathbf{x} \in \mathbb{R}^d$, denoted by $\partial f(\mathbf{x})$, is defined by

$$\partial f(\mathbf{x}) = \{\mathbf{u} \in \mathbb{R}^d \colon \exists \mathbf{x}^k \to \mathbf{x}, f(\mathbf{x}^k) \to f(\mathbf{x}), \tilde{\mathbf{u}}^k \in \tilde{\partial} f(\mathbf{x}^k) \to \mathbf{u}\}.$$

The point $\mathbf{x}$ is a critical point of f if $\mathbf{0} \in \partial f(\mathbf{x})$.

Before stating optimization algorithms that solve (6), we introduce a simpler problem below with a simplified objective $\tilde{F}$. Compared to (6), it has regularization for $\mathbf{Z}^i_k$ only for

positive c_{ki}:

$$\min_{\mathbf{z}\in\mathbb{R}^n,\mathbf{z}_i=0} \tilde{F}(\mathbf{z}) = \|\mathbf{x}_i - \boldsymbol{X}\mathbf{z}\|_2^2 + \gamma \sum_{k:\ 1\le k\le n, c_{ki}>0} c_{ki}\mathbb{I}_{\mathbf{z}_k\neq 0}, \tag{7}$$

where $\mathbf{Z}^i$ is replaced by a simpler notation $\mathbf{z}$.

Proposition 1 in Section A of the supplementary states that a critical point to problem (7) has a limiting-subdifferential arbitrary close to $\mathbf{0}$. As a result, we resort to solve (7) instead of (5). In the following subsections, we first introduce the vanilla Proximal Gradient Descent (PGD) method for (7) with provable convergence, then propose a novel and fast PGD method with improved convergence rate matching Nesterov's optimal convergence rate for first-order methods on smooth and convex problems. In the sequel, we define $f(\mathbf{z}) \triangleq \|\mathbf{x}_i - \boldsymbol{X}\mathbf{z}\|_2^2$ and $h_{\gamma,c}(\mathbf{z}) \triangleq \gamma \sum_{k:\ 1\le k\le n, c_{ki}>0} c_{ki}\mathbb{I}_{\mathbf{z}_k\neq 0}$.

3.1 OPTIMIZATION BY VANILLA PGD

Inspired by recent advances in solving non-convex optimization problems by proximal linearized method [Bolte et al., 2014], we first introduce the vanilla Proximal Gradient Descent (PGD) method to solve the non-convex problem (7). In the following text, the superscript with bracket indicates the iteration number of the PGD method or the iteration number of the coordinate descent without confusion.

In m-th ($m \ge 1$) iteration of the vanilla PGD method, gradient descent is performed on the square loss term f of (7) together with a proximal mapping operator denoted by prox:

$$\begin{aligned}
\mathbf{z}^{(m)} &= \mathrm{prox}_{sh_{\gamma,c}}(\mathbf{z}^{(m-1)} - s\nabla f(\mathbf{z}^{(m-1)})) \\
&= \arg\min_{\mathbf{u}\in\mathbb{R}^n,\mathbf{u}_i=0} \frac{1}{2s}\|\mathbf{u} - (\mathbf{z}^{(m-1)} - s\nabla f(\mathbf{z}^{(m-1)}))\|_2^2 + h_{\gamma,c}(\mathbf{z}) \\
&= T_{s,\gamma,c}(\mathbf{z}^{(m-1)} - s\nabla f(\mathbf{z}^{(m-1)})),
\end{aligned} \tag{8}$$

where $s > 0$ is the step size, $T_{s,\gamma,c}$ is an element-wise hard thresholding operator. For $1 \le k \le n$,

$$[T_{s,\gamma,c}(\mathbf{u})]_k = \begin{cases} 0 & : \ |\mathbf{u}_k| \le \sqrt{2s\gamma c_{ki}} \text{ and } c_{ki} > 0, \text{ or } k = i \\ \mathbf{u}_k & : \ \text{otherwise} \end{cases}$$

The vanilla PGD method starts from $m = 1$ and continue until the sequence $\{\tilde{F}(\mathbf{z}^{(m)})\}$ converges or maximum iteration number is achieved. Theorem 1 below shows that vanilla PGD has a convergence rate of $\mathcal{O}(\frac{1}{m})$ with a proper choice of the step size s.

Theorem 1. Let $\{\mathbf{z}^{(m)}\}$ be the sequence generated by the vanilla PGD with updating rule (17). Then there exists a constant G such that $\|\nabla f(\mathbf{x}^{(m)})\|_2 \le G$ for all $m \ge 1$. Suppose $s \le \min\{\frac{2\tau}{G^2}, \frac{1}{L}\}$ with $L = \sigma_{\max}(\boldsymbol{X}^\top\boldsymbol{X})$ and $\tau = \gamma \min_{k:\ 1\le k\le n, c_{ki}>0} c_{ki}$, then there exists a finite $m_0 > 0$, for all $m > m_0$,

$$0 \le \tilde{F}(\mathbf{z}^{(m)}) - \tilde{F}(\mathbf{z}^*) \le \frac{1}{2s(m - m_0 + 1)}\|\mathbf{z}^{(m_0)} - \mathbf{z}^*\|_2^2, \tag{9}$$

where $\mathbf{z}^*$ is a critical point of $\tilde{F}$.

3.2 FAST PROXIMAL GRADIENT DESCENT BY SUPPORT PROJECTION

Inspired by Nesterov's accelerated Proximal Gradient Descent (PGD) method [Nesterov, 2005, 2013], we propose a novel and fast PGD method to solve (7). The proposed fast PGD algorithm, Fast Proximal Gradient Descent with Support Projection (FPGD-SP), is described in Algorithm 1. In Algorithm 1, $\mathbb{P}_{\mathbf{A}}(\mathbf{u})$ is a novel support projection operator which returns a vector whose elements with indices in $\mathbf{A}$ are the same as those in $\mathbf{u}$, while all the other elements vanish. Namely, $[\mathbb{P}_{\mathbf{A}}(\mathbf{u})]_k = \mathbf{u}_k$ if $k \in \mathbf{A}$, and $[\mathbb{P}_{\mathbf{A}}(\mathbf{u})]_k = 0$ otherwise. Theorem 2 below shows the optimal convergence rate of $\mathcal{O}(\frac{1}{m^2})$ achieved by FPGD-SP. We define $\mathcal{C}^i = \{k\colon 1 \le k \le n, c_{ki} > 0\}$ and we use $\mathcal{C}$ to denote $\mathcal{C}^i$.

Algorithm 1 Fast Proximal Gradient Descent with Support Projection (FPGD-SP)

Input: $\mathbf{v}^{(0)} \in \mathbb{R}^n$, sequence $\{\alpha_k\}$ where $\alpha_k = \frac{2}{k+1}$ for any $k \ge 1$, step size $s > 0$, positive sequence $\{\lambda_k\}$.

1. Set the initial point $\mathbf{z}^{(0)} = \mathbf{v}^{(0)}$ and $k = 1$.
2. Set
$$\mathbf{m}^{(k)} = (1 - \alpha_k)\mathbf{z}^{(k-1)} + \alpha_k\mathbf{v}^{(k-1)}. \tag{10}$$
3. Compute $\mathbf{z}^{(k)}$ as
$$\mathbf{z}^{(k)} = \mathrm{prox}_{sh_{\gamma,c}}(\mathbf{m}^{(k)} - s\nabla f(\mathbf{m}^{(k)})). \tag{11}$$
4. If $\mathrm{supp}(\mathbf{z}_{\mathcal{C}}^{(k)}) = \mathrm{supp}(\mathbf{z}_{\mathcal{C}}^{(k-1)})$, compute
$$\mathbf{v}^{(k)} = \mathbb{P}_{(\mathcal{C}\cap\mathrm{supp}(\mathbf{z}^{(k)}))\cup\bar{\mathcal{C}}}(\mathbf{v}^{(k-1)} - \lambda_k\nabla f(\mathbf{m}^{(k)})). \tag{12}$$
Otherwise, set
$$\mathbf{v}^{(k)} = \mathbf{z}^{(k)}. \tag{13}$$
5. Set $k \leftarrow k + 1$ and go to step 1.

Theorem 2. Let $\{\mathbf{z}^{(m)}\}$ be the sequence generated by Algorithm (1), and the sequence $\{\lambda_k\}$ satisfy $\lambda_{k+1} \ge \frac{k+1}{k}\lambda_k$ for all $k \ge 1$. Then for any $k \ge 1$, λ_k can be chosen such that $\mathrm{supp}(\mathbf{v}^{(k)}) = \mathrm{supp}(\mathbf{z}^{(k)})$. Moreover, there exists a constant G' such that $\|\nabla f(\mathbf{m}^{(m)})\|_2 \le G'$ for all $m \ge 1$. Suppose $s \le \min\{\frac{2\tau}{G'^2}, \frac{1}{L}\}$ with $L = \sigma_{\max}(\boldsymbol{X}^\top\boldsymbol{X})$ and $\tau = \gamma \min_{k:\ 1\le k\le n, c_{ki}>0} c_{ki}$, then there exists a finite $m_0 > 0$, for all $m > m_0$,

$$0 \le \tilde{F}(\mathbf{z}^{(m)}) - \tilde{F}(\mathbf{z}^*) \le \frac{\|\mathbf{v}^{(0)} - \mathbf{z}^*\|_2^2}{m(m+1)}, \tag{14}$$

Algorithm 2 Learning SRSG

Input: the data set $\boldsymbol{X} = \{\mathbf{x}_i\}_{i=1}^n$, the number of clusters c, the parameter γ and K for SRSG, maximum iteration number M_c for coordinate descent, and maximum iteration number M_p for FPGD-SP, stopping threshold ε.

$r = 1$, initialize the sparse code matrix as $\mathbf{Z}^{(0)} = \mathbf{Z}^{(\ell^1)}$.
while $r \leq M_c$ **do**
    Obtain $\mathbf{Z}^{(r)}$ from $\mathbf{Z}^{(r-1)}$ by coordinate descent. In i-th ($1 \leq i \leq n$) step of the r-th iteration of coordinate descent, solve (7) by FPGD-SP descrbibed in Algorithm 1.
    if $|L(\mathbf{Z}^{(r)}) - L(\mathbf{Z}^{(r-1)})| < \varepsilon$ **then**
        break
    else
        $r = r + 1$.
    end if
end while
Obtain the sub-optimal sparse code matrix $\mathbf{Z}^*$ when the above iterations converge or maximum iteration number is achieved.
Build the pairwise similarity matrix by symmetrizing $\mathbf{Z}^*$: $\mathbf{W}^* = \frac{|\mathbf{Z}^*|+|\mathbf{Z}^*|^\top}{2}$
Output: the sparse graph whose weighted adjacency matrix is $\mathbf{W}^*$.

where $\mathbf{z}^*$ is a critical point of $\tilde{F}$.

Theorem 2 shows that FPGD-SP has a faster convergence rate, $\mathcal{O}(\frac{1}{m^2})$, than that of the vanilla PGD which is $\mathcal{O}(\frac{1}{m})$. This result is truly remarkable because $\mathcal{O}(\frac{1}{m^2})$ is the Nesterovâ€™s optimal convergence rate of first-order methods on smooth and convex problems with Lipschitz continuous gradient. While the objective function $\tilde{F}$ is non-convex and nonsmooth, FPGD-SP still achieves the Nesterovâ€™s optimal convergence rate. The proof of Theorem 2 is based on the idea that when the step size for gradient descent is small enough, the support of $\mathbf{z}^{(k)}$ shrinks during iterations of FPGD-SP, so the optimization through FPGD-SP can be divided into a finite number of stages wherein the support of $\mathbf{z}^{(k)}$ belonging to the same stage remains unchanged. Therefore, the objective function $\tilde{F}$ is convex when restricted to a single stage. The optimal convergence rate $\mathcal{O}(\frac{1}{m^2})$ in Theorem 2 is achieved on the final stage. Thanks to the property of support shrinkage, the result of FPGD-SP is always sparser than the initialization point $\mathbf{z}^{(0)}$, so SRSG does not need the ℓ^1-norm to impose sparsity on the solutions to (3) or (7). In our experiments, the initialization point $\mathbf{z}^{(0)}$ is sparse, which can be chosen as the sparse code generated by vanilla sparse graph. Due to its faster convergence rate than the vanilla PGD, we employ FPGD-SP to solve (7) for each step of coordinate descent in the construction of SRSG.

In practice, the iteration of Algorithm 1 is terminated when a maximum iteration number is achieved or certain stopping criterion is satisfied. When the FPGD-SP method converges or terminates, the step of coordinate descent for problem (7) for some $\mathbf{Z}^i$ is finished and the coordinate descent algorithm proceeds to optimize other sparse codes. We initialize $\mathbf{Z}$ as $\mathbf{Z}^{(0)} = \mathbf{Z}_{\ell^1}$ and $\mathbf{Z}_{\ell^1}$ is the sparse codes generated by vanilla sparse graph through solving (1) with some proper weighting parameter λ_{ℓ^1}. In all the experimental results shown in the next section, we empirically set $\lambda_{\ell^1} = 0.1$. The data clustering algorithm by SRSG is described in Algorithm 2. Spectral clustering is performed on the output SRSG produced by Algorithm 2 to generate data clusters for data clustering. More details about Theorem 1 and Theorem 2 and their proofs are in Section A of this paper.

3.3 TIME COMPLEXITY OF BUILDING SRSG USING FPGD-SP

Let the maximum iteration number of coordinate descent be M_c, and maximum iteration number be M_p for the FPGD-SP method used to solve (5). It can be verified that each iteration of Algorithm 1 has a time complexity of $\mathcal{O}(dn)$ where s_0 is the cardinality of the support of the initialization point for FPGD-SP. The overall time complexity of running the coordinate descent for SRSG is $\mathcal{O}(M_c M_p n^2 d)$.

4 EXPERIMENTAL RESULTS

The superior clustering performance of SRSG is demonstrated by extensive experimental results on various data sets. SRSG is compared to K-means (KM), Spectral Clustering (SC), ℓ^1-graph, Sparse Manifold Clustering and Embedding (SMCE) [Elhamifar and Vidal, 2011], and LR-ℓ^1-graph introduced in Section 2.

4.1 EVALUATION METRIC

Two measures are used to evaluate the performance of the clustering methods, i.e. the accuracy and the Normalized Mutual Information (NMI) [Zheng et al., 2004]. Let the predicted label of the datum $\mathbf{x}_i$ be $\hat{y}_i$ which is produced by the clustering method, and y_i is its ground truth label. The accuracy is defined as

$$Accuracy = \frac{\mathbb{I}_{\Omega(\hat{y}_i) \neq y_i}}{n}, \tag{15}$$

where $\mathbb{I}$ is the indicator function, and Ω is the best permutation mapping function by the Kuhn-Munkres algorithm [Plummer and Lovász, 1986]. The more predicted labels match the ground truth ones, the more accuracy value is obtained.

Let $\hat{X}$ be the index set obtained from the predicted labels $\{\hat{y}_i\}_{i=1}^n$, X be the index set from the ground truth labels $\{y_i\}_{i=1}^n$, $H(\hat{X})$ and $H(X)$ be the entropy of $\hat{X}$ and X, then

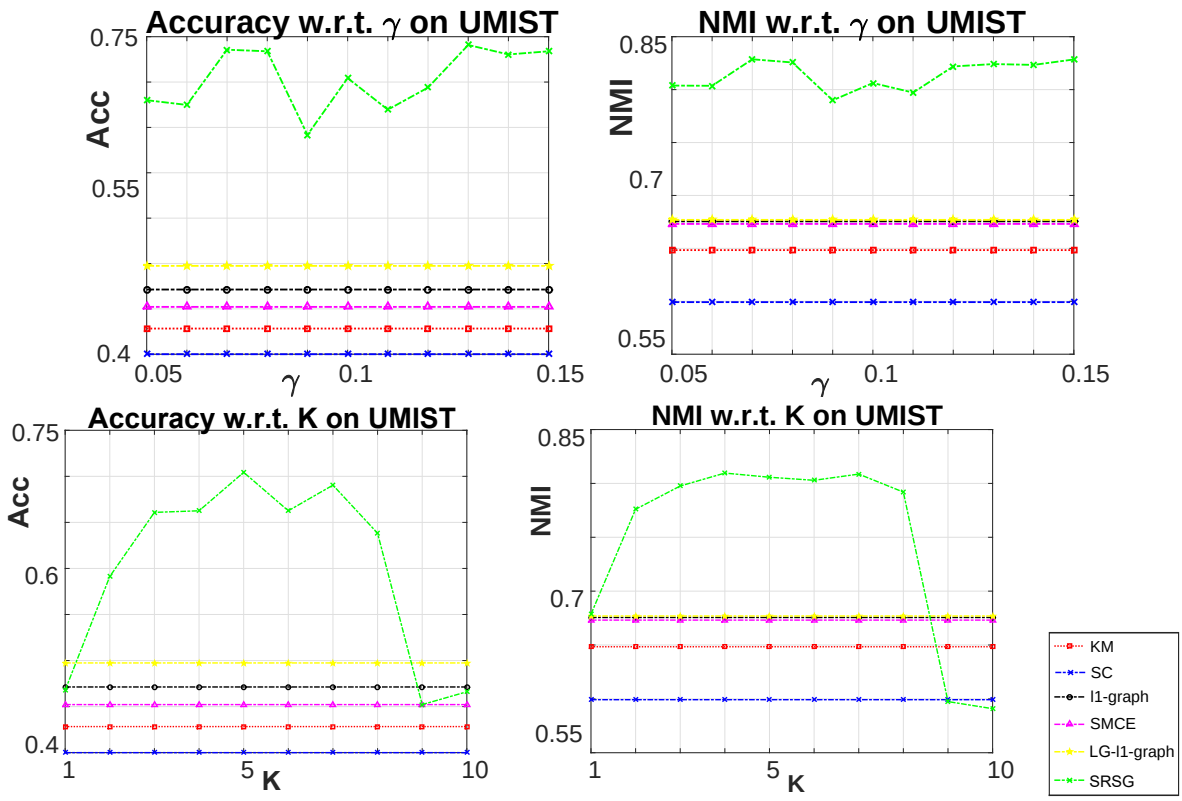

Figure 2: Parameter sensitivey on the UMIST Face Data, from left to right: Accuracy with respect to different values of γ; NMI with respect to different values of γ; Accuracy with respect to different values of K; NMI with respect to different values of K

the normalized mutual information (NMI) is defined as

$$NMI(\hat{X}, X) = \frac{MI(\hat{X}, X)}{\max\{H(\hat{X}), H(X)\}}, \tag{16}$$

where $MI(\hat{X}, X)$ is the mutual information between $\hat{X}$ and X.

4.2 CLUSTERING ON UCI DATA SETS

We conduct experiments on three real data sets from UCI machine learning repository [A. Asuncion, 2007], i.e. Heart, Ionosphere, Breast Cancer (Breast), to reveal the clustering performance of SRSG on general data sets. The clustering results on these three data sets are shown in Table 1.

4.3 CLUSTERING ON COIL-20 AND COIL-100 DATA

COIL-20 Database has 1440 images of resolution 32×32 for 20 objects, and the background is removed in all images. The dimension of this data is 1024. Its enlarged version, COIL-100 Database, contains 100 objects with 72 images of resolution 32×32 for each object. The images of each object were taken 5 degrees apart when each object was rotated on a turntable. The clustering results on these two data sets are shown in Table 2. It can be observed that LR-ℓ^1-graph produces better clustering accuracy than ℓ^1-graph, since graph regularization produces locally smooth sparse codes aligned to the local manifold structure of the data. Using the ℓ^0-norm in the graph regularization term to render the sparse graph that is better aligned to the geometric structure of the data, SRSG always performs better than all other competing methods.

4.4 CLUSTERING ON YALE-B, CMU PIE, CMU MULTI-PIE, UMIST FACE DATA

The Extended Yale Face Database B contains face images for 38 subjects with 64 frontal face images taken under different illuminations for each subject. CMU PIE face data contains cropped face images of size 32×32 for 68 persons, and there are around 170 facial images for each person under different illumination and expressions, with a total number of 11554 images. CMU Multi-PIE (MPIE) data [Gross et al., 2010] contains the facial images captured in four sessions. The UMIST Face Database consists of 575 images of size 112×92 for 20 people. Each person is shown in a range of poses from profile to frontal views - each in a separate directory labelled $1a, 1b, \ldots, 1t$ and images are numbered

Table 1: Clustering Results on three UCI Data Sets

Data Set	Measure	KM	SC	ℓ^1-graph	SMCE	LR-ℓ^1-graph	SRSG
Heart	AC	0.5889	0.6037	0.6370	0.5963	0.6259	**0.6481**
	NMI	0.0182	0.0269	0.0529	0.0255	0.0475	**0.0637**
Ionosphere	AC	0.7095	0.7350	0.5071	0.6809	0.7236	**0.7635**
	NMI	0.1285	0.2155	0.1117	0.0871	0.1621	**0.2355**
Breast	AC	0.8541	0.8822	0.9033	0.8190	**0.9051**	**0.9051**
	NMI	0.4223	0.4810	0.5258	0.3995	0.5249	**0.5333**

Table 2: Clustering Results on COIL-20 and COIL-100 Database. c in the leftmost column indicates that the first c clusters of the entire data set are used for clustering.

COIL-20 / # Clusters	Measure	KM	SC	ℓ^1-graph	SMCE	LR-ℓ^1-graph	SRSG
c = 4	AC	0.6625	0.6701	**1.0000**	0.7639	0.7188	**1.0000**
	NMI	0.5100	0.5455	**1.0000**	0.6741	0.6129	**1.0000**
c = 8	AC	0.5157	0.4514	0.7986	0.5365	0.6858	**0.9705**
	NMI	0.5342	0.4994	0.8950	0.6786	0.6927	**0.9581**
c = 12	AC	0.5823	0.4954	0.7697	0.6806	0.7512	**0.8333**
	NMI	0.6653	0.6096	0.8960	0.8066	0.7836	**0.9160**
c = 16	AC	0.6689	0.4401	0.8264	0.7622	0.8142	**0.8750**
	NMI	0.7552	0.6032	0.9294	0.8730	0.8511	**0.9435**
c = 20	AC	0.6504	0.4271	0.7854	0.7549	0.7771	**0.8208**
	NMI	0.7616	0.6202	0.9148	0.8754	0.8534	**0.9297**
COIL-100 / # Clusters	Measure	KM	SC	ℓ^1-graph	SMCE	LR-ℓ^1-graph	SRSG
c = 20	AC	0.5875	0.4493	0.5340	0.6208	0.6681	**0.9250**
	NMI	0.7448	0.6680	0.7681	0.7993	0.7933	**0.9682**
c = 40	AC	0.5774	0.4160	0.5819	0.6028	0.5944	**0.8465**
	NMI	0.7662	0.6682	0.7911	0.7919	0.7991	**0.9484**
c = 60	AC	0.5330	0.3225	0.5824	0.5877	0.6009	**0.7968**
	NMI	0.7603	0.6254	0.8310	0.7971	0.8059	**0.9323**
c = 80	AC	0.5062	0.3135	0.5380	0.5740	0.5632	**0.7970**
	NMI	0.7458	0.6071	0.8034	0.7931	0.7934	**0.9240**
c = 100	AC	0.4928	0.2833	0.5310	0.5625	0.5493	**0.7425**
	NMI	0.7522	0.5913	0.8015	0.8057	0.8055	**0.9105**

consecutively as they were taken. The clustering results on these four face data sets are shown in Table 3. We conduct an extensive experiment on the popular face data sets in this subsection, and we observe that SRSG always achieve the highest accuracy, and best NMI for most cases, revealing the outstanding performance of our method and the effectiveness of manifold regularization on the local sparse graph structure. Figure 3 in the supplementary demonstrates that the sparse graph generated by SRSG effectively removes many incorrect neighbors for many data points through local smoothness of the sparse graph structure, compared to ℓ^1-graph.

4.5 PARAMETER SETTING

There are two essential parameters for SRSG, i.e. γ for the ℓ^0 regularization term and K for building the adjacency matrix of the KNN graph. We use the sparse codes generated by ℓ^1-graph with weighting parameter $\lambda_{\ell^1} = 0.1$ in (1) to initialize both SRSG and LR-ℓ^1-graph, and set $\lambda = \gamma = 0.1$ in (3) and $K = 5$ for SRSG empirically throughout all the experiments. The maximum iteration number $M = 100$ and the stopping threshold $\varepsilon = 10^{-5}$. The weighting parameter for the ℓ^1-norm in both ℓ^1-graph and LR-ℓ^1-graph, and the regularization weight γ_{ℓ^2} for LR-ℓ^1-graph is chosen from $[0.1, 1]$ for the best performance.

In order to investigate how the performance of SRSG varies with parameter γ and K, we vary the weighting parameter γ and K, and illustrate the result in Figure 2. The performance of SRSG is noticeably better than other competing algorithms over a relatively large range of both λ and K, which demonstrate the robustness of our algorithm with respect to the parameter settings. We also note that a too small K (near to 1) or too big K (near to 10) results in under regularization and over regularization.

Figure 3 in the supplementary demonstrates that the sparse graph generated by SRSG effectively removes many incor-

Table 3: Clustering Results on Various Face Datasets, where CMU Multi-PIE contains the facial images captured in four sessions (S1 to S4)

Yale-B / # Clusters	Measure	KM	SC	ℓ^1-graph	SMCE	LR-ℓ^1-graph	SRSG
c = 10	AC	0.1780	0.1937	0.7580	0.3672	0.4563	**0.8750**
	NMI	0.0911	0.1278	0.7380	0.3264	0.4578	**0.8134**
c = 15	AC	0.1549	0.1748	0.7620	0.3761	0.4778	**0.7754**
	NMI	0.1066	0.1383	0.7590	0.3593	0.5069	**0.7814**
c = 20	AC	0.1227	0.1490	0.7930	0.3542	0.4635	**0.8376**
	NMI	0.0924	0.1223	0.7860	0.3789	0.5046	**0.8357**
c = 30	AC	0.1035	0.1225	0.8210	0.3601	0.5216	**0.8475**
	NMI	0.1105	0.1340	0.8030	0.3947	0.5628	**0.8652**
c = 38	AC	0.0948	0.1060	0.7850	0.3409	0.5091	**0.8500**
	NMI	0.1254	0.1524	0.7760	0.3909	0.5514	**0.8627**
CMU PIE / # Clusters	Measure	KM	SC	ℓ^1-graph	SMCE	LR-ℓ^1-graph	SRSG
c = 20	AC	0.1327	0.1288	0.2329	0.2450	0.3076	**0.3294**
	NMI	0.1220	0.1342	0.2807	0.3047	0.3996	**0.4205**
c = 40	AC	0.1054	0.0867	0.2236	0.1931	0.3412	**0.3525**
	NMI	0.1534	0.1422	0.3354	0.3038	0.4789	**0.4814**
c = 68	AC	0.0829	0.0718	0.2262	0.1731	0.3012	**0.3156**
	NMI	0.1865	0.1760	0.3571	0.3301	**0.5121**	0.4800
CMU Multi-PIE / # Clusters	Measure	KM	SC	ℓ^1-graph	SMCE	LR-ℓ^1-graph	SRSG
MPIE S1	AC	0.1167	0.1309	0.5892	0.1721	0.4173	**0.6815**
	NMI	0.5021	0.5289	0.7653	0.5514	0.7750	**0.8854**
MPIE S2	AC	0.1330	0.1437	0.6994	0.1898	0.5009	**0.7364**
	NMI	0.4847	0.5145	0.8149	0.5293	0.7917	**0.9048**
MPIE S3	AC	0.1322	0.1441	0.6316	0.1856	0.4853	**0.7138**
	NMI	0.4837	0.5150	0.7858	0.5155	0.7837	**0.8963**
MPIE S4	AC	0.1313	0.1469	0.6803	0.1823	0.5246	**0.7649**
	NMI	0.4876	0.5251	0.8063	0.5294	0.8056	**0.9220**
UMIST Face / # Clusters	Measure	KM	SC	ℓ^1-graph	SMCE	LR-ℓ^1-graph	SRSG
c = 4	AC	0.4848	0.5691	0.4390	0.5203	**0.5854**	**0.5854**
	NMI	0.2889	0.4351	0.4645	0.3314	**0.4686**	0.4640
c = 8	AC	0.4330	0.4789	0.4836	0.4695	0.5399	**0.6948**
	NMI	0.5373	0.5236	0.5654	0.5744	0.5721	**0.7333**
c = 12	AC	0.4478	0.4655	0.4505	0.4955	0.5706	**0.6967**
	NMI	0.6121	0.6049	0.5860	0.6445	0.6994	**0.7929**
c = 16	AC	0.4297	0.4539	0.4124	0.4747	0.4700	**0.6544**
	NMI	0.6343	0.6453	0.6199	0.6909	0.6714	**0.7668**
c = 20	AC	0.4216	0.4174	0.4087	0.4452	0.4991	**0.7026**
	NMI	0.6377	0.6095	0.6111	0.6641	0.6893	**0.8038**

rect neighbors for many data points through local smoothness of the sparse graph structure, compared to the vanilla sparse graph (ℓ^1-graph).

5 CONCLUSION

We propose a novel Support Regularized Sparse Graph (SRSG) for data clustering, which employs manifold assumption to align the sparse codes of vanilla to the local manifold structure of the data. We use coordinate descent to optimize the objective function of SRSG and propose a novel and fast Proximal Gradient Descent (PGD) with Support Projection to perform each step of the coordinate descent. Our FPGD-SP solves the non-convex optimization problem of SRSG with a proved convergence rate matching Nesterov's optimal convergence rate for first-order methods on smooth and convex problems. The effectiveness of SRSG for data clustering is demonstrated by extensive experiment on various real data sets.

References

D.J. Newman A. Asuncion. UCI machine learning repository, 2007.

Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006.

Jérôme Bolte, Shoham Sabach, and Marc Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Math. Program.*, 146(1-2): 459–494, August 2014.

Bin Cheng, Jianchao Yang, Shuicheng Yan, Yun Fu, and Thomas S. Huang. Learning with l1-graph for image analysis. *IEEE Transactions on Image Processing*, 19(4): 858–866, 2010.

Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification (2Nd Edition)*. Wiley-Interscience, 2000.

Ehsan Elhamifar and René Vidal. Sparse subspace clustering. In *CVPR*, pages 2790–2797, 2009.

Ehsan Elhamifar and René Vidal. Sparse manifold clustering and embedding. In *NIPS*, pages 55–63, 2011.

Ehsan Elhamifar and René Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(11):2765–2781, 2013.

Brendan J. Frey and Delbert Dueck. Clustering by passing messages between data points. *Science*, 315:2007, 2007.

Shenghua Gao, Ivor Wai-Hung Tsang, and Liang-Tien Chia. Laplacian sparse coding, hypergraph laplacian sparse coding, and applications. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(1):92–104, 2013.

Ralph Gross, Iain Matthews, Jeffrey Cohn, Takeo Kanade, and Simon Baker. Multi-pie. *Image Vision Comput.*, 28 (5):807–813, May 2010.

Xiaofei He, Deng Cai, Yuanlong Shao, Hujun Bao, and Jiawei Han. Laplacian regularized gaussian mixture model for data clustering. *Knowledge and Data Engineering, IEEE Transactions on*, 23(9):1406–1418, Sept 2011. ISSN 1041-4347.

Jialu Liu, Deng Cai, and Xiaofei He. Gaussian mixture model with local consistency. In *AAAI*, 2010.

Yu. Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–152, May 2005.

Yu. Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1):125–161, Aug 2013.

Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, pages 849–856, 2001.

D. Plummer and L. Lovász. *Matching Theory*. North-Holland Mathematics Studies. Elsevier Science, 1986.

R. Tyrrell Rockafellar and Roger J-B Wets. *Variational Analysis*, volume 317. Springer Science & Business Media, 2009.

Satu Elisa Schaeffer. Survey: Graph clustering. *Comput. Sci. Rev.*, 1(1):27–64, August 2007.

Mahdi Soltanolkotabi, Ehsan Elhamifar, and Emmanuel J. Candès. Robust subspace clustering. *Ann. Statist.*, 42(2): 669–699, 04 2014.

Yu-Xiang Wang and Huan Xu. Noisy sparse subspace clustering. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, pages 89–97, 2013.

Shuicheng Yan and Huan Wang. Semi-supervised learning by sparse representation. In *SDM*, pages 792–801, 2009.

Yingzhen Yang, Zhangyang Wang, Jianchao Yang, Jiawei Han, and Thomas Huang. Regularized l1-graph for data clustering. In *Proceedings of the British Machine Vision Conference*. BMVA Press, 2014a.

Yingzhen Yang, Zhangyang Wang, Jianchao Yang, Jiangping Wang, Shiyu Chang, and Thomas S. Huang. Data clustering by laplacian regularized l1-graph. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada.*, pages 3148–3149, 2014b.

Miao Zheng, Jiajun Bu, Chun Chen, Can Wang, Lijun Zhang, Guang Qiu, and Deng Cai. Graph regularized sparse coding for image representation. *IEEE Transactions on Image Processing*, 20(5):1327–1336, 2011.

Xin Zheng, Deng Cai, Xiaofei He, Wei-Ying Ma, and Xueyin Lin. Locality preserving clustering for image database. In *Proceedings of the 12th Annual ACM International Conference on Multimedia*, MULTIMEDIA '04, pages 885–891, New York, NY, USA, 2004. ACM.