Manifold Proximal Linear Methods for Sparse Spectral Clustering with Application to Single-Cell Data Analysis

June 11, 2019

Abstract

As one of the most important unsupervised learning methods, spectral clustering first computes the eigenvectors of the normalized Laplacian matrix to obtain the low-dimensional embedding matrix U of high-dimensional data and then perform the grouping of observations. To enhance the interpretability that $UU^\top$ is a block diagonal matrix, sparse spectral clustering imposes the sparsity of $UU^\top$ when solving the orthogonal embedding matrix U. The exact formulation of sparse spectral clustering is a challenging nonsmooth manifold optimization problem with the orthogonal constraint on U and the complex sparse regularization on $UU^\top$. However, existing algorithms only solves the convex or smooth relaxation of sparse spectral clustering and lack convergence guarantees for the nonsmooth manifold optimization. In this paper, we propose a new manifold proximal linear (ManPL) method to efficiently and accurately solve the exact sparse spectral clustering and its extension based on learning multiple similarity matrices. We provide a unified convergence analysis of the proposed ManPL method without imposing any assumptions on the iterates. The numerical performance is demonstrated in simulation studies and a real application to single-cell data analysis.

1 Introduction

In many machine learning tasks, people need to identify the subgroups (or clustering) on given datasets, which is a classic unsupervised learning problem. K-means [FHT01] and spectral clustering [FHT01] are most popular clustering methods. The standard k-means method, known as Lloyd's algorithm, has the advantage of linear scalability with increasing number of data points, which allows the method to work on large data sets. However,

there are limitations for k-means method. Firstly, it is greedy, and it is not guaranteed to find the global minimum by the algorithm. Secondly, k-means works for the case when all members of each cluster are in close proximity to each other in Euclidean space. However, k-means may fail to capture the local graph structure such as a circle spanning the whole space [KAH19]. In another words, k-means can only handle convex clustering tasks. To overcome these challenges, spectral clustering (SC) is proposed. In SC, first we calculate a similarity matrix based on the local graph structure of data points, then we calcualte Laplacian matrix out of similarity matrix, and eigenvectors of the Laplaician matrix can be used for k-means clustering. SC is easy to implement, and it can be solved efficiently by using modern optimization techniques [NJW02]. Comparing to k-means, spectral clustering is able to identify more local connected structures such as a line or circle spanning the whole space, and can be applied in more general cases. Also, spectral clustering can be viewed as a generalization of k-means, when we use Euclidean distance to construct similarity matrix, it will be regressed to k-means clustering. What's more, spectral clustering can be formulated as a convex optimization problem, thus it is flexible to tune parameter and impose regularization on this framework. [LYL16] observes that in the ideal case, the outer product of eigenvector $\boldsymbol{U}$ of Laplaician matrix should be block diagonal thus sparse matrix. It motivates sparse spectral clustering (SSC), which impose regularization on the sparsity of $\boldsymbol{U}\boldsymbol{U}^T$. SSC performs more stable against high-dimensionality. Based on SSC, [PZ18] proposed a extension of kernel SSC, which provides more flexibility in similarity matrix estimation. And it serves better for non-linear metric.

However, there are still limitations for current SSC method and kernel SSC method in literature. Firstly, the calculated eigenvectors should be orthogonal to each other, which introduces a challenging manifold constrained problem. For the sake of sparsity, ℓ_1 penalty is added on $\boldsymbol{U}\boldsymbol{U}^T$. The challenge is that exact ℓ_1 penaly on $\boldsymbol{U}\boldsymbol{U}^T$ doesn't have a closed form proximal mapping. Current methods do not really resolve the non-convex and non-smooth issue well, but seeking for proper convex relaxation, from which post thresholding is needed and it may affect the accuracy of the clustering result. Also, for the methods without proposing convex relaxation, they don't usually have a convergence guarantee on the constrained nonconvex set [Che+18]; Secondly, the method requires hyper parameter such as the distance normalization parameter for kernel estimator. However, in real world application such as single cell data, we observe that data is very high-dimensional, and the noise level between different embedded groups can be inconsistent. Therefore it is not realistic to preselect a set of hyperparameter such that it can capture the pattern of the

whole data. Rare cell types can be hidden inside a large cluster, or some cell types may be identified wrongly due to the bad choice of hyperparameters.

Our Contribution. In this paper, we want to address these two challenges in sparse spectral clustering (SSC) problem, and make the method be adapted better to single cell data or other complicated data. Instead of solving a relaxed optimization problem, we propose a novel manifold proximal linear method to solve the optimization problem with exact manifold constraint, exact sparse penalty, and analyze the convergence property of our algorithm. We also showed that it can be further extended to multi-kernel setting, such that we can propose multi combinations of hyperparameter, to capture the data pattern using the combination of different choice of hyperparameters. We showed that our algorithm performs better than existing SSC method in both simulation study and real single cell data application.

Organization. The rest of this paper is organized as follows: In section 2 we summerize existing methods on SSC with convex relaxations and smoothing. We introduce our new model in 3 and discuss how we choose the parameters. And then, We introduce the concept of manifold optimization and describe the formulation of our proposed ManPL algorithm in section 4. The convergence analysis is also included. After that we introduce our strategy on efficiently solving the our new model on sparse spectral clustering in 4.3; in the last section, we present the numerical result by comparing our method with other different SSC methods with simulation on artificial datasets, UCI datasets and single cell datasets. Finally, we draw some conclusions.

2 Existing Methods

The basic idea of spectral clustering algorithm is first introduced by Andrew Ng [NJW02]: Given a set of data points $X = [x_1, \ldots, x_n] \in \mathbb{R}^{p\times n}$, where n is the number of samples and p is the dimensionality of the data, spectral clustering (SC) uses the similarity matrix $S = (s_{ij}) \in \mathbb{R}^{n\times n}$, where $s_{ij} \geq 0$ represents a measure of the similarity between data points x_i and x_j. For SC to perform well, it is important to choose an appropriate similarity matrix S. Gaussian function $K(x_i, x_i) = \exp\left(-\|x_i - x_j\|^2 / (2\sigma^2)\right)$ is one of the most widely used functions to construct S (i.e. $s_{ij} = K(x_i, x_j)$), where $\|x_i - x_j\|$ is the Euclidean distance between x_i and x_j and σ controls the width of the neighborhoods. To partition data X into k clusters, SC solves the following optimization problem:

$$\min_{\boldsymbol{U}\in\mathbb{R}^{n\times k}} \left\langle \boldsymbol{U}\boldsymbol{U}^\top, L\right\rangle, \text{ s.t. } \boldsymbol{U}^\top\boldsymbol{U} = I_k$$

where $L = I - D^{-1/2}SD^{-1/2}$ and $D = \operatorname{diag}(d_{11}, \ldots, d_{mn})$ is a diagonal matrix with

$d_{ii} = \sum_{j=1}^{n} s_{ij}$. Finally, compute $\hat{U} \in \mathbb{R}^{n\times k}$ by normalizing each row of U to have unit Euclidean length, treat the rows of $\hat{U}$ as data points in $\mathbb{R}^k$, and cluster them into k groups by k-means or other clustering algorithm.

In the ideal case, we need to have affinity matrix S is block diagonal, i.e., $w_{ij} = 0$ if x_i and x_j are from different clusters. Let $C \in \mathbb{R}^{n\times k}$ denotes the indicator matrix whose row entries indicate to which group the points belong. That is, if x_i belongs to the group $l, c_{il} = 1$ and $c_{ij} = 0$ for all $j \neq l$. Then, for any orthogonal matrix $R \in \mathbb{R}^{k\times k}$, we have $\hat{\boldsymbol{U}} = CR$. In this case, $\hat{\boldsymbol{U}}\hat{\boldsymbol{U}}^\top$ is block diagonal. Hence, $\hat{\boldsymbol{U}}\hat{\boldsymbol{U}}^\top$ implies the true membership of the data clusters and it is naturally sparse. Note that $\hat{\boldsymbol{U}}$ is obtained by normalizing each row of $\boldsymbol{U}$ and thus $\boldsymbol{U}\boldsymbol{U}^\top$ is also sparse. However, such a block diagonal or sparse property may not appear in real scenarios since the affinity matrix W is usually not block diagonal. This motivates the sparse regularization on $\boldsymbol{U}\boldsymbol{U}^\top$ and thus leads to the model [LYL16]:

$$\min_{\boldsymbol{U}\in\mathbb{R}^{n\times k}} \left\langle L, \boldsymbol{U}\boldsymbol{U}^\top \right\rangle + \beta \left\| \boldsymbol{U}\boldsymbol{U}^\top \right\|_0 \text{ s.t. } \boldsymbol{U}^\top\boldsymbol{U} = I_k$$

Where $\|\cdot\|_0$ is the ℓ_0 norm which represents the number of nonzero entries in the input matrix. However, the key challenge is that problem is both nonconvex and NP-hard due to the ℓ_0 norm, which is difficult to solve. Many method has been proposed to resolve the nonconvex and nonsmooth issues in the proposed problem.

Existing methods to boost the performance of SSC can be summarized into two categories:

1. Improve the SC accuracy when a data similarity matrix is fixed, which are stressed in the following two optimization problems [Lu+18; LYL16]:

$$\begin{aligned} P* &= \arg\min_{P\in\mathbb{R}^{n\times n}} \langle L, P\rangle + \|P\|_1 \text{ s.t. } 0 \preceq P \preceq I, \mathrm{Tr}(P) = k \\ U^* &= \arg\min_{\boldsymbol{U}\in\mathbb{R}^{n\times k}} \|P^* - \boldsymbol{U}\boldsymbol{U}^T\|_F^2 \text{ s.t. } \boldsymbol{U}^T\boldsymbol{U} = I_k \end{aligned} \tag{2.1}$$

$$\boldsymbol{U}^* = \arg\min_{\boldsymbol{U}\in\mathbb{R}^{n\times k}} \langle L, \boldsymbol{U}\boldsymbol{U}^T\rangle + g_\sigma(\boldsymbol{U}\boldsymbol{U}^T) \text{ s.t. } \boldsymbol{U}^T\boldsymbol{U} = I_k \tag{2.2}$$

In which $\|\cdot\|_1$ is the vector ℓ_1 norm, $g_\sigma(\cdot)$ is the smoothed ℓ_1 norm [Bec17; Lu+18]. The constraint in (2.1) is the convex hull of the steifel manifold,which can be viewed as a convex relaxation. From which we can see that the performance is boosted by adding sparse constraint on $\boldsymbol{U}\boldsymbol{U}^T$ when data is noisy. ADMM is proposed on solve these both optimization problems [Lu+18]. The problem for (2.1) is that optimization on relaxed version of non-convex set doesn't always return an optimal solution; the problem for (2.2) is that it is well known that the proposed ADMM algorithm on solving composite function doesn't have a convergence guarantee on the stiefel manifold.

2. Construct an appropriate similarity matrix to improve the clustering performance, which can be summarized in the following optimization problem [PZ18]:

$$\begin{aligned} P^* = &\underset{P\in\mathbb{R}^{n\times n}}{\arg\min}\, c\|P\|_F^2 - \left\langle \sum_{i=1}^{d} w_i G_i, P \right\rangle + \|P\|_1 \\ &\text{s.t. } 0 \preceq P \preceq I, \mathrm{Tr}(P) = k, \sum_{i=1}^{d} w_i = 1,\ \forall i\ w_i \geq 0 \\ U^* = &\underset{U\in\mathbb{R}^{n\times k}}{\arg\min}\, \|P^* - \boldsymbol{U}\boldsymbol{U}^T\|_F^2 \text{ s.t. } \boldsymbol{U}^T\boldsymbol{U} = I_k \end{aligned} \tag{2.3}$$

In which each G_i is the normalized similarity matrix. c is a small positive number, adding this term can provide desired convergence property for the algorithm. We can see that (2.3) still can perform well when data has inconsistant noise but it has the same problem as (2.1) arising from the convex relaxation and post thresholding.

In all, from a numerical optimization perspective, convex relaxation and smoothing techniques are added on the objective functions of SC on those new methods mentioned above, which won't usually give us the optimal clustering result. Due to the needs from sparse spectral clustering, we proposed a new statistical learning model based on the classical spectral clustering problem.

3 Exact Multiple Kernel Sparse Spectral Clustering

Based on the first approach, we modify the SC framework by imposing exact sparse structure on the non-relaxed target matrix. This is motivated by the observation that this structure is essential for better clustering performance, but is not often obtained by SC when the data includes high level noise, also not obtained by using any convex relaxations. Relating to the second approach, we utilize multiple affinity matrices (denoted by $(\boldsymbol{G}_1, \ldots, \boldsymbol{G}_d)$) to construct a robust similarity matrix from linear combinations on the atoms with proper corresponding weights $\boldsymbol{w} = (w_1, \ldots, w_d)$. This can help to obtain more accurate and robust clustering results, even effective in real scenarios when the data includes many missing values and imbalanced similarities.

Motivated by these challenges and the needs to overcome the numerical issues from non-convexity and non-smoothness of the objective function, we consider the *exact multiple kernel sparse spectral clustering (EMKSSC)* model, which can be formulated by following

optimization problem:

$$\begin{aligned}&\min_{\boldsymbol{U}\in\mathbb{R}^{n\times k},\boldsymbol{w}\in\mathbb{R}^d} c\|\boldsymbol{U}\|_F^2-\left\langle\sum_{j=1}^d w_j\boldsymbol{G}_j,\boldsymbol{U}\boldsymbol{U}^T\right\rangle+\lambda\|\boldsymbol{U}\boldsymbol{U}^T\|_1+\rho\sum_{j=1}^d w_j\log w_j\\&s.t.\ \ \boldsymbol{U}^T\boldsymbol{U}=\boldsymbol{I}_k;\ \sum_{j=1}^d w_j=1;\quad w_j\geq 0\text{ for }j=1,\ldots,d.\end{aligned}\tag{3.1}$$

In which, the vector $\boldsymbol{w}$ lies in the probability simplex. Each kernel matrix $\boldsymbol{L}_j$ is chosen in the following way: for samples $\tilde{i}$ and $\tilde{j}$ with $1\leq\tilde{i}\leq\tilde{j}\leq n$

$$\begin{aligned}\boldsymbol{G}_j(\tilde{i},\tilde{j})&=\exp\left(-\frac{\left\|x_{\tilde{i}}-x_{\tilde{j}}\right\|^2}{2\epsilon_{\tilde{i}\tilde{j}}^2}\right)\\\epsilon_{\tilde{i}\tilde{j}}&=\frac{\sigma\left(\mu_{\tilde{i}}+\mu_{\tilde{j}}\right)}{2},\mu_{\tilde{i}}=\frac{\sum_{k\in\mathrm{KNN}(\tilde{i})}\left\|x_{\tilde{i}}-x_k\right\|}{k}\end{aligned}\tag{3.2}$$

where KNN $(\tilde{i})$ represents a set of sample indices that are the top k nearest neighbors of the sample $x_{\tilde{i}}$. We use the ℓ_1 penalty to encourage the sparsity of block diagonal matrix $\boldsymbol{U}\boldsymbol{U}^T$. In this settings, we directly promote the sparsity on the matrix $\boldsymbol{U}\boldsymbol{U}^T$. We can see that the problem is nonconvex and nonmooth, which brings the challenge to be efficiently solved.

In this paragraph, we explain the motivation of our multi-kernel framework based on K-NN graph and its advantage over single kernel. Because in multi-kernel settings, the graph information is based on the K-nearest neighbour (KNN) information. We want to make sure that the KNN graph can capture the structure of the data, so we choose the K in KNN to be a little bit larger than largest number of members in one cluster of the data. For example, if the data has 150 element and the size of each cluster is 50, then we choose K in KNN to be 55 to 60. The remaining thing is to determine each Laplacian matrix. The Laplacian matrix is heavily determined by the choice of hyper parameter (k,σ). However, it is hard to know what choice of parameter is the best beforehand. And in biological applications, such as single cell data, the pattern of different kinds of cells can be different, they have difference noise level and corruption rates which means the subgroup pattern cannot all be captured by a single kernel matrix determined by one set of parameters. This is the main motivation of introducing multi-kernel into our setting. At the same time, we jointly estimate the spectral embedding and their weights corresponding to each kernel. This framework guarantees us that we can use sufficient number of kernels to capture the data pattern, and the redundant kernels will be assigned low weight, thus they will not hurt the model accuracy. The choice of number of kernels is a trade-off between covering sufficient large range of hyper parameter space, but should not overfit the data.

Based on the needs of efficiently and accuracy for the optimization procedure, we also proposed a new proximal gradient method to solve it by combining the power of proximal linear method and manifold optimization. The proposed manifold proximal linear method (ManPL) is based on the the proximal linear method with a retraction operation to keep the iterations feasible with respect to the manifold constraint. Each step of ManPL algorithm involves a well-structured convex optimization problem, which can be solved efficiently by using semi-smooth newton based proximal point algorithm (Newton-PPA). We will prove that the algorithm converges to a stationary point of the proposed algorithm. Iteration complexity of ManPL also has also been analyzed for obtained an ϵ stationary point. The numerical results on benchmark machine learning datasets shows that one can obtain favorably clustering results by using our EMKSSC model.

4 Proximal Linear Method on Riemannian Manifold

Before we introduce the structure of the proposed ManPL algorithm, we want to introduce the concept and historical method of manifold optimization.

4.1 Nonsmooth Manifold Optimization

Optimization over Riemannian manifolds has recently drawn a lot of attention recently due to its applications in many different fields [AMS08], including low rank matrix completion [BA11], phase retrieval and blind source separation. The basic procedure of manifold optimization has been proposed as the following: Given a smooth manifold $\mathcal{M} \subset \mathbb{R}^n$ and a differentiable function $f : \mathcal{M} \to \mathbb{R}$, the procedure of the gradient descent algorithm for solving $\min_{x\in\mathcal{M}} f(x)$ is as follows:

- Step 1. Consider $f(x)$ as a function from $\mathbb{R}^n$ to $\mathbb{R}$ and calculate the Euclidean gradient $\nabla f(x)$
- Step 2. Calculate its Riemannian gradient $\mathbf{grad}\, f(x)$, which is the direction of steepest ascent of $f(x)$ among all directions in the tangent space $T_x\mathcal{M}$. This direction is given by $P_{T_x\mathcal{M}}\nabla f(x)$, where $P_{T_x}\mathcal{M}$ is the projection operator to the tangent space $T_x\mathcal{M}$.
- Step 3. Define a retraction R_x that maps the tangent space back to the manifold, i.e. $R_x : T_x\mathcal{M} \to \mathcal{M}$, where $\mathbf{Retr}_x$ needs to satisfy the conditions. In particular, $\mathbf{Retr}_x(0) = x, \mathbf{Retr}_x(y) = x + y + o(\|y\|^2)$ as $y \to 0$, and $\mathbf{Retr}_x$ needs to be smooth.

Then the update of the gradient descent algorithm x^+ is defined by:

$$x^+ = \underset{x}{\mathbf{Retr}}\left(-\eta P_{T_x\mathcal{M}}\nabla f(x)\right)$$

In many machine learning applications the objective function is not smooth, for example, in sparse principle components analysis problem [ZHT06], one can't directly calculate the Riemannian Gradient of the objective function. Chen propose the Proximal Gradient Method on Manifold (ManPG) [Che+18] algorithm to dealing with this kind of problem. The ManPG algorithm states as the follows:

Consider we want to solve

$$\min F(X) = f(X) + h(X) \text{ s.t. } X \in \mathcal{M}$$

in which f is L_f-smooth and h is nonsmooth and L_h Lipstchiz continuous. In the kth iterations, ManPG computes a descent direction D_k by solving the following optimization problem:

$$\min_D \langle \mathbf{grad}\, f(X_k), D\rangle + \frac{1}{2t}\|D\|_2^2 + h(X_k + D) \text{ s.t. } D \in T_{X_k}\mathcal{M} \tag{4.1}$$

with $t \leq L^{-1} = \min(L_f^{-1}, L_h^{-1})$. After we find the optimal D_k, we will use amijo line search to find an optimal stepsize α and apply retraction to update X_{k+1} for the next iteration.

$$X_{k+1} = \underset{X_k}{\mathbf{Retr}}(\alpha D_k) \tag{4.2}$$

For more information on ManPG, please refer [Che+18].

4.2 Proximal Linear Algorithm

In many machine learning problem like robust phase retrieval [DR17] and robust blind deconvolution [Cha+19], people usually encountered with minimizing composite functions:

$$\min_x F(x) = g(x) + h(c(x)) \tag{4.3}$$

in which g is a convex function, can be smooth or nonsmooth, h is a convex and L-Lipstchiz continuous function, c is a C^1 smooth function with a β-Lipstchiz continuous Jacobian map. People use so called proximal-linear method to solve this problem. One can view this as natural extension of the proximal gradient method. In each iteration, we want to solve the subproblem with stepsize t:

$$x_{k+1} = \arg\min_x \Big\{ g(x) + h\Big(c(x_k) + \nabla c(x_k)(x - x_k)\Big) + \frac{1}{2t}\|x - x_k\|_2^2 \Big\} \tag{4.4}$$

The scheme (4.4) reduced to the popular prox-gradient algorithm [Bec17] for additive composite minimization. One can shows that with the optimal choice $t = (\beta L)^{-1}$, the prox-linear algorithm will find a point x satisfying $\|t^{-1}(x - x_k)\| \leq \epsilon$ after at most $\mathcal{O}(\frac{L\beta}{\epsilon^2}(F(x_0) - \inf F(x)))$ iterations [DP18].

In order to solve many machine learning problems with the manifold constraint, by the inspiration of [Che+18], we proposed new algorithm based on proximal linear algorithm and manifold optimization:

Consider the following optimization problem:

$$\min_X F(x) = g(X) + h(c(X)) \text{ s.t. } X \in \mathcal{M} \tag{4.5}$$

In which $g : \mathbb{R}^d \to \mathbb{R}$ is convex and L_g-smooth; $h : \mathbb{R}^m \to \mathbb{R}$ is a Lipschitz function with constant L_h; $c : \mathbb{R}^d \to \mathbb{R}^m$ is a C^1 smoothing mapping with L_c-Lipschitz continuous Jacobian Map. $\mathcal{M}$ is some manifold constraint. In order to deal with the manifold constraint, we need to ensure that the descent direction lies in the tangent space. This motivates ManPL alorithm first compute the descent direction D_k by solving the following optimization problem with step-size $t \leq \min((L_g)^{-1}, (L_c L_h)^{-1})$:

$$\min_D \langle \mathbf{grad}\, g(X_k), D\rangle + \frac{1}{2t}\|D\|_2^2 + h(c(X_k) + \mathbf{grad}\, c(X_k)D) \text{ s.t. } D \in T_{X_k}\mathcal{M} \tag{4.6}$$

Note that (4.5) is different from Proximal Linear method in the Euclidean setting in two places: the Euclidean gradient ∇g is changed to Riemannian gradient $\mathbf{grad}\, g$, the descent direction D_k is changed to the tangent space of the manifold at point X_k.

By following the definition of Riemannian gradient [AMS08], we will have:

$$\langle \mathbf{grad}\, g(X_k), D\rangle = \langle \nabla g(X_k), D\rangle \quad \forall D \in T_{X_k}\mathcal{M}$$

Now we can rewrite the problem as:

$$\min_D \langle \nabla g(X_k), D\rangle + \frac{1}{2t}\|D\|_2^2 + h(c(X_k) + \nabla c(X_k)D) \text{ s.t. } D \in T_{X_k}\mathcal{M} \tag{4.7}$$

As a result one don't need to compute the Riemannian gradient $\mathbf{grad}\, f$ but only the Euclidian gradient ∇f is needed. Also one can view the problem (4.7) as a proximal gradient step restrict to the tangent space $T_{X_k}\mathcal{M}$. Since $X_k + \alpha D_k$ does not necessary lies on the manifold $\mathcal{M}$, one will perform a retraction back to $\mathcal{M}$. The routine of ManPL will be described in Algorithm 1, which also evolves a Amijo line search method procedure to determine the

stepsize α. And we will show that the method is well-defined, which means it will terminate after finite number of steps.

Algorithm 1: Manifold Proximal Linear Method (ManPL)

Input: Initial point $X_0 \in \mathcal{M}, \gamma \in (0,1), \delta \in (0,1)$, Lipstchiz constant L

for k = *1,...,N* **do**

- Calculate D_k by solving the subproblem (4.7) with $t \in (0, 1/L]$
- Set $\alpha = 1$

  while $F(\mathbf{Retr}(\alpha D_k)) > F(X_k) - \delta\alpha\|D_k\|_2^2$

  Update α to $\gamma\alpha$

  end
- Set $X_{k+1} = \mathbf{Retr}_{X_k}(\alpha D_k)$

end

4.2.1 Semi-smooth Newton Based PPA method on Solving Subproblem

In order to solve the subproblem more efficiently and more accurately, we need to find a good optimization strategy. From this purpose, we will choose Semi-smooth Newton (SSN) method to solve the subproblem in each update of ManPL. The notion of semi-smoothness was originally introduced by Mifflin [Mif77] for real valued functions and later extended to vector-valued mappings by Qi and Sun [QS93]. The SSN method has recently received significant amount of attention due to its success in solving structured convex problems to a high accuracy in problems such as LASSO [YST13], convex clustering [WST10], and SDP [ZST10]. The numerical on SSN shows that it is faster than most proposed first order methods [Xia+18].

Inspired by the idea of [YST13], we will proposed a semi-smooth newton based proximal point algorithm (Newton-PPA) for solving the subproblem with $\mathcal{M} = \mathrm{St}(n,r) = \{X : X \in \mathbb{R}^{n\times r}, X^TX = I_r\}$. The tangent space to $\mathcal{M} = \mathrm{St}(n,r)$ is given by:

$$T_X\mathcal{M} = \{V | V^TX + X^TV = 0\}$$

We can define an operator $\mathcal{A}_k(X) = V^TX + X^TV$, which is an linear operator. First, let's denote the following parameters: $C_1 = t\nabla g(X_k), D = z, L = \nabla c(X_k), C_2 = c(X_k)$.

Now we are turn to solve the following problem:

$$\arg\min_{z} \frac{1}{2t}\|z + C_1\|_2^2 + h(Lz + C_2) \text{ s.t. } \mathcal{A}z = 0 \tag{4.8}$$

Consider $y = Lz$, We can write down the Lagrangian associated with (4.8) to be:

$$\mathcal{L}(z, y; \lambda_1, \lambda_2) = \frac{1}{2t}\|z + C_1\|_2^2 + h(y + C_2) - \langle \lambda_1, \mathcal{A}(z)\rangle - \langle \lambda_2, Lz - y\rangle$$

Following the strong convexity, (4.8) is equivalent to

$$\min_{z,y} f(z, y) = \min_{z,y} \max_{\lambda_1,\lambda_2} \mathcal{L}(z, y; \lambda_1, \lambda_2) \tag{4.9}$$

Where f is the essential objective function of (4.8), which is defined by:

$$f(z, y) = \max_{\lambda_1,\lambda_2} \mathcal{L}(z, y; \lambda_1, \lambda_2) \tag{4.10}$$

In the following, we concentrate on (4.9) , to which apply the PPA. First we compute the Moreau-Yosida regularization of the essential objective function f. After applying the PPA algorithm on the the essential function, we will have the following result:

$$(z_{k+1}, y_{k+1}) \approx \min_{u,v} f(u, v) + \frac{1}{2\beta}\left(\|u - z_k\|_2^2 + \|v - y_k\|_2^2\right) \tag{4.11}$$

For simplicity, here the proximal parameter β is assumed to be constant. The saddle point formulation of (4.8) is given by

$$\max_{\lambda_1,\lambda_2} \min_{u,v} \mathcal{L}(u, v, \lambda_1, \lambda_2) + \frac{1}{2\beta}\left(\|u - z_k\|_2^2 + \|v - y_k\|_2^2\right) \tag{4.12}$$

which implies dual problem of (4.11) is given by

$$\max_{\lambda_1,\lambda_2} \Theta_\beta(z, y; \lambda_1, \lambda_2) \tag{4.13}$$

Our strategy is that, at each iterations, we first (approximately) solve the dual problem to obtain the optimal dual variables. By using the first order optimal condition (FOC), we will have:

$$-\nabla_{\lambda_1,\lambda_2} \Theta_\beta(z, y; \lambda_1^*, \lambda_2^*) = 0 \tag{4.14}$$

and then we update the corresponding primal variable by using the corresponding proximal mapping formulated by the PPA.

Now we return to find the root of the equation:

$$E(\lambda_1, \lambda_2) \equiv -\nabla_{\lambda_1,\lambda_2} \Theta_\beta(z^*, y^*; \lambda_1^*, \lambda_2^*)$$

We will introduce how we apply SSN on finding the root of $E(\lambda_1, \lambda_2)$. In order to apply SSN, we need to compute the generalized Jacobian J_k of $E(\lambda_1, \lambda_2)$. After several steps of manipulations, one can show that the generalized Jacobian of $E(\lambda_1, \lambda_2)$ is given by:

$$J_k = \frac{t\beta}{t+\beta} \begin{bmatrix} \mathcal{A} \\ L \end{bmatrix} \begin{bmatrix} \mathcal{A} \\ L \end{bmatrix}^T + \beta \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & M \end{bmatrix} \tag{4.15}$$

where M is a diagonal matrix which is defined by:

$$(M(y))_{ii} = \begin{cases} 1 \text{ if } |C_2 + y^k - \beta\lambda_2| > \mu\beta \\ 0 \text{ otherwise} \end{cases}$$

when $h(X) = \mu\|X\|_1$. It should be pointed out that J_k can be singular, therefore, vanilla SSN can't be applied directly and we need to resort to adapted regularized SSN (ASSN) proposed by [Xia+18] and it turns out very suitable for solving the proposed subproblem. ASSN first computes the Newton's direction d_k by solving the following equation:

$$(J_k + \eta I)d_k = -E(\lambda_1, \lambda_2) \tag{4.16}$$

with a regularization parameter η. ASSN designed a strategy to decide whether to accept this d_k or not. Roughly speaking, if $\|E(\lambda_1^+, \lambda_2^+)\|_2$ is sufficiently decreased from $\|E(\lambda_1, \lambda_2)\|_2$, then we accept d_k, where:

$$\begin{bmatrix} \lambda_1^+ \\ \lambda_2^+ \end{bmatrix} = \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} + d_k \tag{4.17}$$

The challenge in solving the system of linear equation is that inverting the matrix $J_k + \eta I$ is costing when the dimension of J_k is large. Since the matrix $\frac{t\beta}{t+\beta} \begin{bmatrix} \mathcal{A} \\ L \end{bmatrix} \begin{bmatrix} \mathcal{A} \\ L \end{bmatrix}^T$ is low rank, one can try to use binomial inverse theorem to reduced the dimension of the proposed system of linear equations. As a result, one can use the following equation to compute the inverse of $J + \eta I$:

$$\begin{aligned} & (J_k + \eta I)^{-1} \\ =& \frac{t+\beta}{t\beta} \left(\begin{bmatrix} \mathcal{A} \\ L \end{bmatrix} \begin{bmatrix} \mathcal{A} \\ L \end{bmatrix}^T + \mathcal{D} \right)^{-1} \\ =& \frac{t+\beta}{t\beta} \left(\mathcal{D}^{-1} - \mathcal{D}^{-1} \begin{bmatrix} \mathcal{A} \\ L \end{bmatrix} \left(I + \begin{bmatrix} \mathcal{A} \\ L \end{bmatrix}^T \mathcal{D}^{-1} \begin{bmatrix} \mathcal{A} \\ L \end{bmatrix} \right)^{-1} \begin{bmatrix} \mathcal{A} \\ L \end{bmatrix}^T \mathcal{D}^{-1} \right) \end{aligned} \tag{4.18}$$

Where $\mathcal{D} = \frac{t+\beta}{t}\begin{bmatrix} 0 & 0 \\ 0 & M \end{bmatrix} + \frac{t+\beta}{t\beta}\eta I$ is a diagonal matrix, which is also invertiable. By using strategies list above, one can solve the the subproblem with super linear convergence rate and higher accuracy than other first order method [Xia+18].

4.2.2 Convergence Analysis of ManPL algorithm

We will the following assumptions on the objective functions before we carry out the convergence analysis of the ManPL. Our convergence analysis consists of three parts. First in Lemma 4.1, we show that D_k in (4.6) is a descent direction for the objective function (4.7); Second, in lemma 4.2 the objective function show that D_k is a descent direction for the objective function of (4.5). In other words, There is a sufficient decrease of the function value of objective function when we update the variable X. Third, we establish the global convergence of ManPL in Theorem 4.1. All the details of the proof can be found in the appendix in the paper.

We first make the following assumption on the objective function F.

Assumption 4.1. *The objective function F of (4.5) satisfies the following properties.*

- *$F(X)$ is coercive, i.e. $F(X) \to +\infty$. An immediate consequence of this is that the sub-level set $\{X|F(X) \leq F(X_0)\}$ is bounded*
- *$F(X)$ is lower bounded on $\mathcal{M}$, i.e., there exists a constant F_* such that $F(X) \geq F_*$ for all $X \in \mathcal{M}$*

Lemma 4.1. *(Descent lemma) Let D_k be the minimizer of (4.7), the following holds for any $\alpha \in [0, 1)$, with $t < \left(\min\left((L_g)^{-1}, (L_c L_h)^{-1}\right)\right)$,*

$$g(X_k + \alpha D_k) - g(X_k) + h\Big(c(X_k) + \alpha \nabla c(X_k) D_k\Big) - h\Big(c(X_k)\Big) \leq -\frac{\alpha}{2t}\Big\|D_k\Big\|_2^2 \tag{4.19}$$

The following lemma shows that $\{F(X_k)\}$ is monotonically decreasing, where $\{X_k\}$ is generated by Algorithm 1.

Lemma 4.2. *(Sufficient decrease property) Consider D_k is the optimal solution of (4.7). Denote $X_k^+ = X_k + \alpha D_k$, $X_{k+1} = \boldsymbol{Retr}_{X_k}(\alpha D_k)$, there exist a constant $\beta_k(\alpha) > 0$ such that*

$$F(X_{k+1}) - F(X_k) \leq -\beta_k(\alpha)\|D_k\|_F^2 \tag{4.20}$$

This motivates the definition of the stationary point of the algorithm.

Definition 4.1. *A point $X \in \mathcal{M}$ is called a stationary point of problem (4.5) if it satisfies the first order necessary condition. i.e.* $0 \in \boldsymbol{grad}\, c(X)\partial h\Big(c(X)\Big) + \boldsymbol{grad}\, g(X)$.

Definition 4.2. *X_k is called an ϵ-stationary point of (4.5), if D_k returned by (4.7) satisfies $\|D_k\|_F \leq \epsilon$.*

The following lemma shows that if Algorithm 1 can't find a suitable descent direction when solving 4.7, then a stationary point is found.

Lemma 4.3. *If $D_k = 0$, then X_k is a stationary point of problem (4.5).*

From the above lemma, We can thus use $\|D_k\|_F \leq \epsilon$ as the stopping criterion of Algorithm 1 with $t = 1/L$. From Lemma 4.20, we obtain the following result which is similar to the one in [**Theorem 2**; BA11] for manifold optimization with smooth objectives, which is also analog to the one in [**Theorem 5.6**; Che+18] and [**Theorem 3.3**; Che+19] for nonsmooth objectives.

By combining all the lemma and assumptions above, one can shows the iteration complexity when finding a stationary point of (4.5).

Theorem 4.1. *The limit point of the sequence $\{X_k\}$ generated by the proposed algorithm is a stationary point of (4.5). Denote $\overline{\beta} = \frac{1}{k}\sum_{i=1}^{k} \beta_i(\alpha)$. Furthermore, the proposed algorithm returns $\{X_k\}$ satisfying $\|D_k\|_F \leq \epsilon$ in at most $(F(X_k) - F^*)/(\overline{\beta}\epsilon^2)$ iterations, in which F^* is the optimal value of (4.5).*

4.3 Algorithm on EMKSSC and Convergence Analysis

We now introduce how we solve the proposed EMKSSC problem. Inspired by the idea from (PALM) [BST14], we first update $\boldsymbol{U}$ by using the ManPL algorithm; after we update the variable $\boldsymbol{U}$, we will updated the weights $\boldsymbol{w}$ followed by a closed form solution [PZ18]. One can view our routine as an extension of PALM. By combining the analysis of ManPL algorithm, we can show that the objective function value of EMKSSC can converge to a stationary point. By following the convergence analysis of ManPL, we can show the convergence of algorithm on EMKSSC.

Lemma 4.4. *(Sufficient Decrease Condition) Consider D_k is the optimal solution get from the solving the proximal linear subproblem of (EMKSSC) problem. Denote $\boldsymbol{U}_k^+ = \boldsymbol{U}_k + \alpha D_k$,*

$\boldsymbol{U}_{k+1} = \boldsymbol{Retr}_{\boldsymbol{U}_k}(\alpha D_k)$, *in each iteration of the update, there exist a constant* $\tilde{\beta}_k(\alpha) > 0$ *such that*

$$\boldsymbol{F}(\boldsymbol{U}_{k+1}, \boldsymbol{w}_{k+1}) - \boldsymbol{F}(\boldsymbol{U}_k, \boldsymbol{w}_k) \leq -\tilde{\beta}_k(\alpha)\|D_k\|_F^2 \tag{4.21}$$

Theorem 4.2. *The limit point of the sequence* $\{\boldsymbol{U}_k, \boldsymbol{w}_k\}$ *generated by the proposed algorithm is a stationary point of* (3.1). *Denote* $\overline{\beta} = \frac{1}{k}\sum_{i=1}^{k} \tilde{\beta}_i(\alpha)$. *Furthermore, the proposed algorithm start with* U_0 *obtained by SC and with any weights* $\boldsymbol{w}_0$ *belongs to the probability simplex, returns* $\{\boldsymbol{U}_k\}$ *satisfying* $\|D_k\|_F \leq \epsilon$ *in at most* $(\boldsymbol{F}(\boldsymbol{U}_0, \boldsymbol{w}_0) - \boldsymbol{F}^*)/(\overline{\beta}\epsilon^2)$ *iterations, in which* $\boldsymbol{F}^*$ *is the optimal value of* (3.1).

5 Simulation Studies

Before we choose multiple kernel for the simulation, we want to show how we choose the multiple kernels.

5.1 Numerical Results with Single Kernel

In this section, we will compare our algorithms with historical methods, including traditional SC and different SSC. We will use the following abbreviations:

- SC: spectral clustering [NJW02]
- SSC: sparse spectral clusteing by [LYL16]
- NCSSC: nonconvex sparse spectral clustering by [Lu+18]
- ENSSC: exact nonconvex sparse spectral clustering proposed by us with single kernel
- MKSSC: multiple kernel sparse spectral clustering by: [PZ18]
- NCMKSSC: nonconvex multiple kernel sparse spectral clustering by combining methods in [Lu+18; PZ18]
- EMKSSC: our proposed multiple kernel method with sparse spectral clustering

Here we highlight each algorithm by its characteristics and solvers they used:

Name	Smoothing	Relaxation	Kernels	Solver	Reference
SC	No	No	Single	Eigen-solver	[NJW02]
SSC	No	Yes	Single	ADMM	[LYL16]
NCSSC	Yes	No	Single	ADMM	[Lu+18]
ENSSC	No	No	Single	ManPL	Novel
MKSSC	No	Yes	Multiple	ADMM	[PZ18]
NCMKSSC	Yes	No	Multiple	ADMM	[Lu+18; PZ18]
EMKSSC	No	No	Multiple	ManPL	Novel

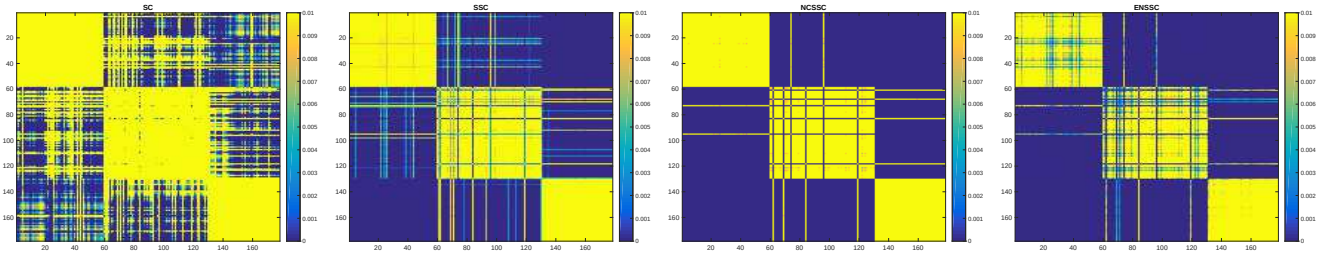
5.2 Simulation from Benchmark Datasets

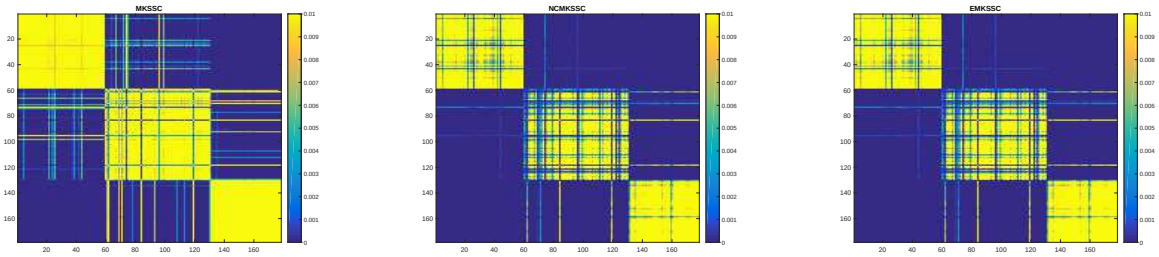
Here we use the UCI machine learning datasets [DG17] to compare the result between our method with nonconvex spectral clustering method with the full view proposed by Lu [LYL16] and the full view spectral clustering [NJW02]. For all the testing datasets we will using the NMI [FHT01] as our metric.

By following the construction of similarity matrix suggested by [PZ18] with smartly tuning the parameters, we have the following results:

Category	SC	SSC	NSSC	ENSSC	MKSSC	NCMKSSC	EMKSSC	Clusters
Wine	0.8650	0.8650	0.8782	0.8782	0.8854	0.8854	**0.8926**	3
Iris	0.7496	0.7582	0.7582	0.7582	0.7665	0.7601	**0.7705**	3
Glass	0.3165	0.3418	0.2047	0.3471	0.2656	0.3315	**0.3644**	6

Table 1: NMI score for different Datasets with different algoithms





MKSSC NCMKSSC EMKSSC

Figure 1: Resulting $\boldsymbol{U}\boldsymbol{U}^T$ matrix on UCI wine data from different algorithms, all the data are scaled from 0 to 0.01

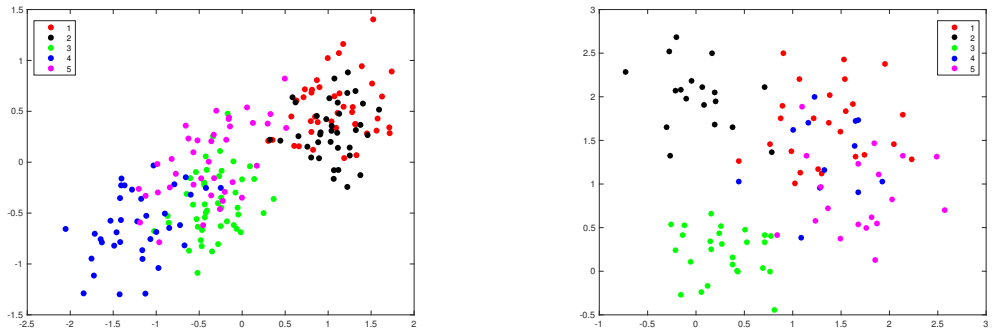
From Table 1, We can see that the proposed multiview method has higher NMI results over other method with full view similarity matrix in most cases. From Figure 1, we can also see that sparse spectral clustering achieves sparse block structure on UCI wine data, and our proposed EMKSSC estimator is able to recover most of the true block structure comparing to all other methods. For example, NCSSC has worse performance on second block, while ENSSC and NCMKSSC has slightly worse performance on both first and second blocks.

5.3 Numerical Simulation on EMKSSC

In this section, we consider two simulation setting to evaluate the performance of our estimators comparing to others followed by the idea of [PZ18]. In setting 1, the points are well clustered in optimal two dimensional space. And all the clusters span a circle in the two-dimensional space. We projected the 2-dimensional data to p-dimensional space, and adding heterogeneous random noises in p dimensional space. Through this process, the data is noisy in p-dimensional space, and level of noise in different dimensions are different. It is crucial to use sparse spectral clustering to identify the low-dimensional space. And considering the heterogeneity, multi kernels are necessary to identify different patterns varying from different dimensions. Ideally, our ENMKSSC can help recover the 2-dimensional space, which cluster the data well. And the noise in this projected 2-dimensional space will be mostly removed, and we can achieve a satisfying clustering result. In setting 2, we consider a sparse setting similar to genetic study. For each cluster, we randomly generated in a d-dimensional space, and we fill in all zeros in other $p - d$ dimensions , parallel to the noisy predictors in genetic study. Each cluster corresponds to different column of B. For X in each membership, we add a heterogeneous noise on the column of B, corresponding to their clusters. In this way, it is important to identify the original cluster of each data point. Similarly, due to the compli-

cation and heterogeneity of the noises, ENMKSSC is desired for clustering problem under this setting. And we compare our performance with NMKSSC and MKSSC methods. To compare all these methods fairly, we tune regularization parameter λ through oracle tuning. In the tuning, we generate two independent dataset from same generating procedures as we proposed, namely training set and testing set. Then we use training set to estimate $\boldsymbol{U}$, and use testing set to evaluate the clustering accuracy through NMI score. We replicate this process for 50 times. Then we pick λ such that we can minimize the average clustering error on testing dataset.

The detailed data generating procedure is provided as follows:



(a) Setting 1 (b) Setting 2

Figure 2: Example of data distribution for two settings

Setting 1

- We randomly generate C points in the 2-dimensional latent space spanning a 2-dimensional circle, regrading them as the center for each cluster.
- We randomly generate n points as follows. For each point, we randomly select one center point among these C points in the previous step as its center, and then add an independent noise to it. The noise of each point can be different to address the importance of multi kernel matrices.
- We project the generated k-dimensional data to a p-dimensional space through a pre-generated linear projection matrix P.
- We add additional noise ϵ to the post projection points for all p-dimension.

In setting 1, we set the noise level to be 40% of the radius of the embedded dimension. In the following experiments, we set that number of embedded clusters to be 5 and we perform the experiment of SC and MKSSC on the same data.

Setting 2

- We generate a left singular matrix $B \in \mathbb{R}^{C\times d}$ with independent Gaussian random elements for each entries, where $d < p$, where the variance for each row is different.Then we fill in other $p-d$ column as zero, i.e. $B' = \left[B, 0_{C\times(p-d)}\right]$
- Generate the cluster label $z_i \in [C]$ of the ith sample by random assignment to one group. Then generate membership matrix $Z \in \mathbb{R}^{n\times C}$ with $Z_{ij} = 1_{\{z_i=j\}}$
- Generate $X = ZB + W$, where W is standard Gaussian noise matrix.

In setting 2, we set the noise level to be 20% of the radius of the embedded dimension. In the following experiments, we set that number of embedded clusters to be 5 and we perform the experiment of SC and MKSSC on the same data. In figure 2, we can see a realization of the simulated data for both settings in randomly selected 2-dimensions. We can see that different clusters mix together and the variability between clusters and different points varies. Traditional methods cannot cluster them effectively. The simulation result for two settings are summarized in following tables and Figure 3.

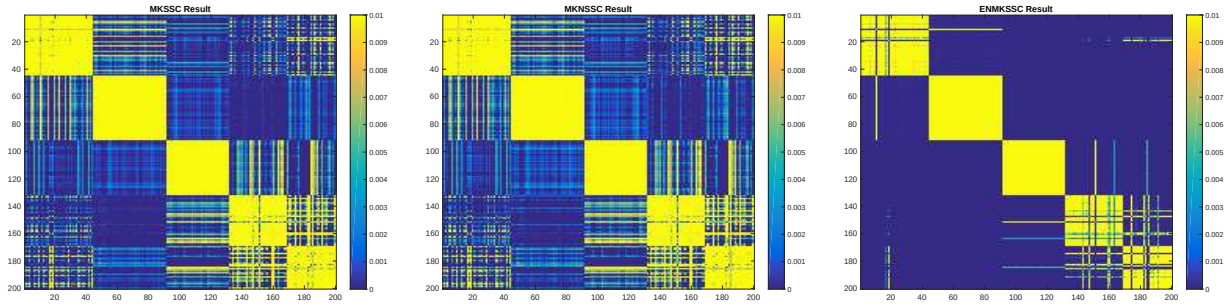
n	p	MKSSC	NCMKSSC	EMKSSC
100	250	0.6606(1.5e-2)	0.8514(1.4e-2)	0.8685(1.8e-2)
100	300	0.8872(2.1e-2)	0.8198(1.7e-2)	0.8994(2.1e-2)
100	500	0.7604(2.0e-2)	0.8217(1.3e-2)	0.8228(1.1e-2)
200	250	0.8864(1.6e-2)	0.8940(1.0e-2)	0.9017(0.7e-2)
200	300	0.8615(2.1e-2)	0.8816(1.1e-2)	0.8894(0.8e-2)
200	500	0.7930(1.5e-2)	0.8480(1.2e-2)	0.8660(0.2e-2)

Table 2: NMI results for Setting 1

In simulation, we find that the choice of λ is not very sensitive in a reasonable range around $5e^{-3}$. The average NMI and standard error is reported based on 15 replications. From the heatmaps, we can see that our ENMKSSC achieves a significant more sparse estimation, comparing to both MKSSC and NMKSSC. From the NMI tables, we find that since our estimator solves exact manifold constraint, we are able to estimate $\boldsymbol{U}$ more accurate, and

n	p	MKSSC	NCMKSSC	EMKSSC
100	250	0.7205(1.1e-2)	0.7223(1.9e-2)	0.7325(1.1e-2)
100	300	0.6764(1.2e-2)	0.8284(1.5e-2)	0.8664(0.7e-2)
100	500	0.6253(2.3e-2)	0.7289(1.3e-2)	0.7308(1.1e-2)
200	250	0.1329(1.1e-2)	0.8694(2.3e-2)	0.8932(1.2e-2)
200	300	0.1767(2.2e-2)	0.8003(1.2e-2)	0.9352(0.8e-2)
200	500	0.2628(1.2e-2)	0.6048(2.1e-2)	0.6475(0.6e-2)

Table 3: NMI results for Setting 2



MKSSC NMKSSC ENMKSSC

Figure 3: Resulting $\boldsymbol{U}\boldsymbol{U}^T$ Matrix from Setting 1 of Proposed Artificial Data with , data in all the settings are scaled from 0 to 1, the noise is set to be 30 percent of the radius of the data

consistently higher NMI scores over different sample size and dimensions. From figure 3, the advantage of our proposed ENMKSSC estimator is also clear. We can see that both MKSSC and NMKSSC suffer from high level of noises between different blocks. While our ENMKSSC method achieves a very clear 5-block structure with much less noise. It indicates that solving the exact manifold constrains help us reduce significant amount of noise and bias in spectral clustering, and lead us to a better clustering result.

6 Application to Single-Cell Data Analysis

In this section, we consider a real application to single-cell data. Clustering cells and identifying subgroups are important topics in scRNA-seq analysis. [PZ18] studied the performance of spectral clustering and multi kernel spectral clustering with convex relaxation. It is found that when single kernel is applied, the clustering result is sensitive to the choice of number of

neighbors and scaling parameter σ. Therefore multi kernel SSC can help alleviate this concern. However, [PZ18] use relaxed convex constraints to mimic manifold constraint, which may affect the clustering performance in single-cell data analysis. In single-cell data analysis, the number of cells group can be large[PZ18], and the data is noisy, high-dimensional with missing ones[BK16; Bre+13]. Thus the bias introduced by convex relaxation will also be enlarged. Thus it would be interesting to study whether the exact manifold constrained SSC can help address this concern. With these motivations,we would like to further investigate whether our proposed exact manifold constrained sparse spectral clustering can further improve the performance.

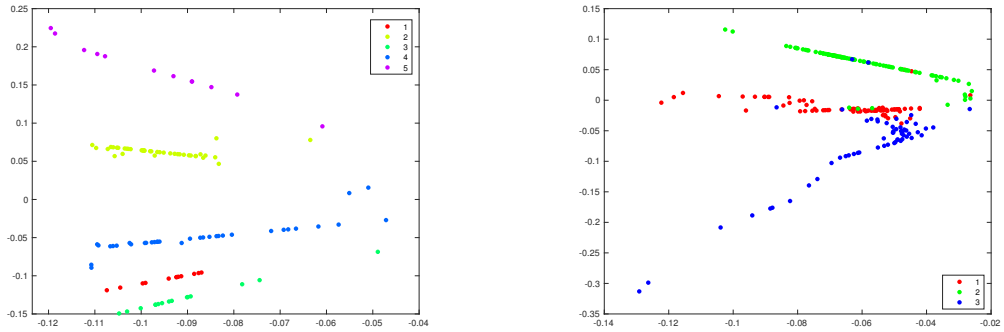
We apply our multi kernel sparse spectral clustering method to single-cell RNA-seq dataset, to identify the single cell types. Each dataset represents several types of dynamic processes such as cell differentiation, and each RNA-seq data contains cells for which the labels were known from prior knowledge. The number of clusters are provided in following table 4. We applied MKSSC, NCMKSSC, EMKSSC to seven different RNA-seq datasets studied in [PZ18]. From the oracle tuning result in simulation, we see that if we set λ in the reasonable range, the clustering result is not sensitive to λ. Therefore we can unify the choice of λ according to the choice in [PZ18]. The NMI results are reported in the following table.

Category	MKSSC	NCMKSSC	EMKSSC	Clusters
Deng [Den+14]	0.7319	0.7389	**0.7464**	7
Ting [Tin+14]	0.9283	0.9524	**0.9755**	5
Treutlein [Tre+14]	0.7674	0.7229	**0.8817**	5
Buettner [Bue+15]	0.7929	0.8744	**0.8997**	3
Ginhoux [Sch+15]	0.6206	0.6398	**0.6560**	3
Pollen [Pol+14]	0.9439	0.9372	**0.9631**	11

Table 4: NMI results from real biological data sets

From the table, we can see that our estimator can consistently perform better than MKSSC and NCMKSSC over all datasets. And we can scale well to sample size to be reasonably large. The following plot represents the first two principle directions of the clustering result of Ting's data.

From table 4, we can see that our method significantly improve NMI score comparing to existing constrain-relaxed methods. As discussed, the improvement comes from that we reduce the bias from constrain relaxation, especially the number of clusters is large. And the



(a) Ting (b) Ginhoux

Figure 4: Clustering results from EMKSSC for single cell datasets

2-D visualization shows that our low dimension embedding well separate different classes, achieving a better embedding comparing with the ones in [PZ18], and the boundary between different classes are very clear. Even in the more challenging Ginhous dataset, although we have some misclassfied points, we still separate three clusters more clear comparing to the result in [PZ18]. Our clustering result can be used to further analyze the single-cell data clustering problem in an accurate and efficient way.

7 Conclusion

In this paper, we proposed the ManPL algorithm to solve the sparse spectral clustering problem without any smoothing and convex relaxation on the nonsmooth and nonconvex object. With certain assumptions on the objective function, We prove that our algorithm converges globally to the stationary point of (4.5). Moreover, we analyzed the iteration complexity of our algorithm for obtaining an ϵ-stationary solution. With the power of semi-smooth Newton's PPA on finding the descent direction, we can ensure that ManPL algorithm performs better on the clustering task for different type of datasets than historical method on sparse spectral clustering. In this paper, we only analyze that the performance of our algorithm on Stiefel manifold. We also believe that our ManPL algorithm can apply to more general manifold structures. It remains an interesting question that whether the matrix $\mathcal{A}$ in (4.8) can be computed efficiently for other interesting manifold.

A Preliminaries

Lemma A.1. *[Che+18] Let $\overline{\mathcal{M}} := \{X \in \mathcal{M} | \|X\|_F \leq \psi\}$ be a compact subset of $\mathcal{M}$ and $K := \{\xi \in T\mathcal{M} | \|\xi\|_F \leq \phi\}$ be a bounded subset of the tangent bundle $\mathrm{t}\mathcal{M}$, where $\psi, \phi > 0$ are given scalars. For all $X \in \overline{\mathcal{M}}$ and $\xi \in K$ there exist constants $M_1 > 0$ and $M_2 > 0$ such that the following two inequalities hold:*

$$\|\mathrm{Retr}_X(\xi) - X\|_F \leq M_1 \|\xi\|_F, \forall X \in \overline{\mathcal{M}}, \xi \in K \tag{A.1}$$

$$\|\mathrm{Retr}_X(\xi) - (X + \xi)\|_F \leq M_2 \|\xi\|_F^2, \forall X \in \overline{\mathcal{M}}, \xi \in K \tag{A.2}$$

Assumption A.1. *The objective function F of (4.5) satisfies the following properties.*

- *$F(X)$ is coercive, i.e. $F(X) \to +\infty$. An immediate consequence of this is that the sub-level set $\{X | F(X) \leq F(X_0)\}$ is bounded*
- *$F(X)$ is lower bounded on $\mathcal{M}$, i.e., there exists a constant F_* such that $F(X) \geq F_*$ for all $X \in \mathcal{M}$*

B Proofs

B.1 Convergence Analysis of ManPL

Before we presents the result in ManPL, we will use the following useful lemma in proximal linear algorithm.

Lemma B.1. *[DP18] By using the assumption on h and c, we will have the following result:*

$$-\frac{L_c L_h}{2}\|y - x\|_2^2 \leq h\Big(c(y)\Big) - h\Big(c(x) + \nabla c(x)(y - x)\Big) \leq \frac{L_c L_h}{2}\|y - x\|_2^2 \tag{B.1}$$

Proof. Since

$$
\begin{aligned}
&\Big|h(c(y)) - h(c(x) + \nabla c(x)^T(y-x))\Big| \\
\leq & L_h \|c(y) - \big(c(x) + \nabla c(x)^T(y-x)\big)\|_2 \\
= & L_h \left\| \int_0^1 \left(\nabla c(x + t(y-x)) - \nabla c(x)\right)(y-x) dt \right\|_2 \\
\leq & L_h \int_0^1 \| \left(\nabla c(x + t(y-x)) - \nabla c(x)\right) \|_2 \|(y-x)\|_2 dt \\
\leq & L_h L_c \left(\int_0^1 t dt \right) \|y-x\|_2^2 = \frac{L_c L_h}{2} \|x-y\|_2^2
\end{aligned}
$$

□

In this section, we analyze the convergence and iteration complexity of our ManPL algorithm. We made the following assumptions.

Lemma B.2. *Let D_k be the minimizer of (4.7), the following holds for any $\alpha \in [0,1)$, with $t < \Big(\min\Big((L_g)^{-1}, (L_c L_h)^{-1}\Big)\Big)$,*

$$
g(X_k + \alpha D_k) - g(X_k) + h\Big(c(X_k) + \alpha \nabla c(X_k) D_k\Big) - h\Big(c(X_k)\Big) \leq -\frac{\alpha}{2t} \Big\| D_k \Big\|_2^2 \tag{B.2}
$$

Proof. Since D_k is the minimizer of (4.7), we will have for any $\alpha \in [0,1)$:

$$
\begin{aligned}
&\langle \nabla g(X_k), \alpha D_k \rangle + \frac{1}{2t} \|\alpha(D_k)\|_2^2 + h\Big(c(X_k) + \alpha \nabla c(X_k)(D_k)\Big) \\
\geq & \langle \nabla g(X_k), D_k \rangle + \frac{1}{2t} \|(D_k)\|_2^2 + h\Big(c(X_k) + \nabla c(X_k)(D_k)\Big)
\end{aligned}
$$

which implies that:

$$
(1-\alpha)\langle \nabla g(X_k), D_k \rangle + \frac{1-\alpha^2}{2t} \|D_k\|_2^2 + h\Big(c(X_k) + \nabla c(X_k)(D_k)\Big) - h\Big(c(X_k) + \alpha \nabla c(X_k) D_k\Big) \leq 0
$$

Using the convexity of h, we will have:

$$
\langle \nabla g(X_k), D_k \rangle + \frac{1+\alpha}{2t} \|D_k\|_2^2 + h\Big(c(X_k) + \nabla c(X_k)(D_k)\Big) - h\Big(c(X_k)\Big) \leq 0 \tag{B.3}
$$

Consider $\alpha \to 1$, we will have:

$$
\langle \nabla g(X_k)), D_k \rangle + h\Big(c(X_k) + \nabla c(X_k)(D_k)\Big) - h\Big(c(X_k)\Big) \leq -\frac{1}{t} \|D_k\|_2^2 \tag{B.4}
$$

By using the L_g smoothness of g, convexity of h and $\alpha \leq 1$, we can have that:

$$\begin{aligned} & g(X_k + \alpha D_k) - g(X_k) + h\Big(c(X_k) + \alpha \nabla c(X_k) D_k\Big) - h\Big(c(X_k)\Big) \\ \leq & \langle \nabla g(X_k), \alpha D_k \rangle + \frac{\alpha}{2t}\|D\|_F^2 + \alpha\Big(h\Big(c(X_k) + \nabla c(X_k) D_k\Big) - h\Big(c(X_k)\Big)\Big) \\ \leq & -\frac{\alpha}{2t}\|D_k\|_2^2 \end{aligned} \tag{B.5}$$

□

The following lemma will be useful for the proving the following result.

Lemma B.3. *Consider D_k is the optimal solution of (4.7). Denote $X_k^+ = X_k + \alpha D_k$, $X_{k+1} = \boldsymbol{Retr}_{X_k}(\alpha D_k)$, there exist a constant $\beta_k(\alpha) > 0$ such that*

$$F(X_{k+1}) - F(X_k) \leq -\beta_k(\alpha)\|D_k\|_F^2 \tag{B.6}$$

Proof. We will prove by induction. For $k = 0$, the set $\Omega = \{X | F(X) \leq F(X_0)\}$ and D_0 is bounded, which is due to Assumption A.1 and lemma B.2. First, by using the convexity of h and Lipchitz continuity of c, we can show that:

$$\begin{aligned} & h\Big(c(X_{k+1})\Big) - h\Big(c(X_k) + \alpha \nabla c(X_k)(D_k)\Big) \\ = & h\Big(c(X_{k+1})\Big) - h\Big(c(X_k) + \nabla c(X_k)(X_{k+1} - X_k)\Big) \\ & + h\Big(c(X_k) + \nabla c(X_k)(X_{k+1} - X_k)\Big) - h\Big(c(X_k) + \nabla c(X_k)(X_k^+ - X_k)\Big) \end{aligned} \tag{B.7}$$

$$\begin{aligned} \overset{(B.1),(A.1)}{\leq} & \frac{\alpha^2 L_h L_c}{2} M_1 \|D_k\|_2^2 + L_h \|\nabla c(X_k)(X_{k+1} - X_k^+)\|_2 \\ \overset{(A.2)}{\leq} & \frac{\alpha^2 L_c L_h}{2} M_1 \|D_k\|_2^2 + L_h L_c M_2 \alpha^2 \|D_k\|_2^2 = \Big(\frac{1}{2} M_1 + M_2\Big) L_c L_h \alpha^2 \|D_k\|_2^2 \end{aligned}$$

Since $\nabla g(X)$ is L_g Lipschiz continuous, we will have:

$$\begin{aligned} g(X_{k+1}) - g(X_k) &\leq \langle \nabla g(X_k), X_{k+1} - X_k \rangle + \frac{L_g}{2}\|X_{k+1} - X_k\|_2^2 \\ &= \langle \nabla g(X_k), X_{k+1} - X_k^+ + X_k^+ - X_k \rangle + \frac{L_g}{2}\|X_{k+1} - X_k\|_2^2 \\ &\overset{(A.2)}{\leq} M_2 \|\nabla g(X_k)\|_2 \|\alpha D_k\|_2^2 + \alpha \langle \nabla g(X_k), D_k \rangle + \frac{M_1 L_g}{2}\|\alpha D_k\|_2^2 \\ &\leq c_0 \alpha^2 \|D_k\|_2^2 + \alpha \langle \nabla g(X_k), D_k \rangle \end{aligned} \tag{B.8}$$

Since $\nabla g(X)$ is continuous on the compact set Ω, there exist a constant $G_1 \geq 0$, such that $\|\nabla g(X)\|_2 \leq G_1$ for all $X \in \Omega$. Also, we denote $c_0 = M_2 G_1 + \frac{1}{2} M_1 L_g$.

Now we want to show that the value of objective function converges, by using lemma B.2, we will have:

$$
\begin{aligned}
& F(X_{k+1}) - F(X_k) \\
\overset{(B.8)}{\leq} & \alpha \langle \nabla g(X_k), D_k \rangle + c_0 \alpha^2 \|D_k\|^2 + h\Big(c(X_{k+1})\Big) - h\Big(c(X_k) + \alpha \nabla c(X_k)(D_k)\Big) \\
& + h\Big(c(X_k) + \alpha \nabla c(X_k)(D_k)\Big) - h\Big(c(X_k)\Big) \\
\overset{(B.7)}{\leq} & \alpha \langle \nabla g(X_k), D_k \rangle + c_0 \alpha^2 \|D_k\|_2^2 + (\frac{1}{2} M_1 + M_2) L_c L_h \alpha^2 \|D_k\|_2^2 \\
& + \alpha \left(h\Big(c(X_k) + \nabla c(X_k)(D_k)\Big) - h\Big(c(X_k)\Big) \right) \\
\overset{(B.4)}{\leq} & \left[\left(c_0 + \left(\frac{1}{2} M_1 + M_2 \right) L_c L_h \right) \alpha^2 - \alpha \tilde{L} \right] \|D_k\|_F^2
\end{aligned}
$$

Where $\tilde{L} = \Big(\min\Big((L_g)^{-1}, (L_c L_h)^{-1}\Big)\Big)^{-1}$.

Now we want to define the function:

$$\beta(\alpha) = -\Big(c_0 + \Big(\frac{1}{2} M_1 + M_2\Big) L_c L_h\Big)\alpha^2 + \tilde{L}\alpha$$

with $\overline{\alpha} = \dfrac{\tilde{L}}{2\Big(c_0 + \Big(\frac{1}{2} M_1 + M_2\Big) L_c L_h\Big)}$, by using the result above, we can show that:

$$F(X_{k+1}) - F(X_k) \leq -\beta_k(\alpha)\|D_k\|_F^2 \quad \text{if } 0 \leq \alpha \leq \min\{1, \overline{\alpha}\} \tag{B.9}$$

where

$$\beta_k(\alpha) = \begin{cases} \beta(\alpha_1), \text{if } \overline{\alpha} \leq 1 \\ \beta(1), \text{if } \overline{\alpha} > 1 \end{cases}$$

Thus, the result (B.3) holds for $k = 0$. From assumption A.1 by using induction, we can show that (B.3) holds for all $k \geq 1$.

□

Definition B.1. *X_k is called an ϵ-stationary point of (4.5), if D_k returned by (4.7) satisfies $\|D_k\|_F \leq \epsilon$.*

Lemma B.4. *If $D_k = 0$, then X_k is a stationary point of problem (4.5).*

Proof. Consider the optimality conditions for the subproblem of ManPL is given by:

$$0 \in \mathbf{grad}\, c(X_k) \cdot \partial h(c(X_k) + \mathbf{grad}\, c(X_k)^T(D_k)) + \frac{1}{t} D_k + \mathbf{grad}\, g(X_k)$$

If $D_k = 0$, it follows that:

$$0 \in \mathbf{grad}\, c(x_k) \partial h(c(x_k)) + \mathbf{grad}\, g(X_k)$$

which is the first-order optimal condition of (4.5) □

Theorem B.1. *The limit point of the sequence $\{X_k\}$ generated by the proposed algorithm is a stationary point of (4.5). Denote $\overline{\beta} = \frac{1}{k}\sum_{i=1}^{k} \beta_k(\alpha)$. Furthermore, the proposed algorithm returns $\{X_k\}$ satisfying $\|D_k\|_F \leq \epsilon$ in at most $(F(X_k) - F^*)/(\overline{\beta}\epsilon^2)$ iterations.*

Proof. By using the lemma, it follows that any limit point of $\{X_k\}$ is a stationary point. Moreover, since the sub-level set $\Omega = \{X | F(X) \leq F(X_0)\}$ is compact, thus there exist at least one limit point of the sequence $\{X_k\}$.

Furthermore, suppose that Algorithm 1 doesn't terminate after K iterations, that is, $\|D_k\|_F > \epsilon$ for all $k = 0, 1, ..., K-1$. In this case, we have

$$F(X_0) - F^* \geq F(X_0) - F(X_K) \geq \overline{\beta} \sum_{k=0}^{K-1} \|D_k\|_F^2 > \overline{\beta} K \epsilon^2.$$

Therefore, Algorithm 1 terminates after $K \geq (F(X_0) - F^*)/(\overline{\beta}\epsilon^2)$ iterations when it finds an ϵ-stationary point. □

B.2 Convergence Analysis of algorithm on EMKSSC

Lemma B.5. *Consider D_k is the optimal solution get from the solving the ManPL algorithm of (EMKSSC) problem. Denote $\boldsymbol{U}_k^+ = \boldsymbol{U}_k + \alpha D_k$, $\boldsymbol{U}_{k+1} = \boldsymbol{Retr}_{\boldsymbol{U}_k}(\alpha D_k)$, there exist a constant $\tilde{\beta}_k(\alpha) > 0$ such that*

$$\boldsymbol{F}(\boldsymbol{U}_{k+1}, \boldsymbol{w}_{k+1}) - \boldsymbol{F}(\boldsymbol{U}_k, \boldsymbol{w}_k) \leq -\tilde{\beta}_k(\alpha) \|D_k\|_F^2 \tag{B.10}$$

Proof. In this part we only need to show that $\boldsymbol{f}(\boldsymbol{U}, \boldsymbol{w}_{k+1}) = \sum_{i=1}^{n} \boldsymbol{w}_i \langle \boldsymbol{L}_i, \boldsymbol{U}\boldsymbol{U}^T \rangle$ is Lipschitz smooth and every remaining steps will followed by the proof of lemma B.3. Since for every $\boldsymbol{w} \in \mathbb{R}^d$, we will have:

$$
\begin{aligned}
&\|\nabla_{\boldsymbol{U}} \boldsymbol{f}(\boldsymbol{U}_1, \boldsymbol{w}) - \nabla_{\boldsymbol{U}} \boldsymbol{f}(\boldsymbol{U}_2, \boldsymbol{w})\|_F \\
\leq & 2\| \sum_{i=1}^{d} w_i \boldsymbol{L}_i (\boldsymbol{U}_1 - \boldsymbol{U}_2)\|_F \\
= & 2\|(I_p \otimes \sum_{i=1}^{d} w_i \boldsymbol{L}_i)(\mathbf{vec}(\boldsymbol{U}_1) - \mathbf{vec}(\boldsymbol{U}_2))\|_2 \\
\leq & 2\|I_p \otimes \sum_{i=1}^{d} w_i \boldsymbol{L}_i\|_2 \|U_1 - U_2\|_F = 2\left\|I_p\right\|_2 \left\|\sum_{i=1}^{d} w_i \boldsymbol{L}_i\right\|_2 \|\boldsymbol{U}_1 - \boldsymbol{U}_2\|_F \\
\leq & 2 \sum_{i=1}^{d} w_i \|\boldsymbol{L}_i\|_2 \|\boldsymbol{U}_1 - \boldsymbol{U}_2\|_F \leq 2\|\boldsymbol{U}_1 - \boldsymbol{U}_2\|_F
\end{aligned}
$$

By using $\|\boldsymbol{L}_i\|_2 = 1$ for every i. This implies that for every fixed $\boldsymbol{w}$, $f(\boldsymbol{U}, \boldsymbol{w})$ is Lipschitz smooth with constant 2. By combining the result that the $\boldsymbol{w}$ subproblem always has a sufficient descent, the results will be followed by the proof of ManPL. □

Theorem B.2. *The limit point of the sequence $\{\boldsymbol{U}_k\}$ generated by the proposed algorithm on solving (EMKSSC) is a stationary point of (4.5). Denote $\overline{\beta} = \frac{1}{k}\sum_{i=1}^{k} \beta_k(\alpha)$. Furthermore, the proposed algorithm returns $\{\boldsymbol{U}_k\}$ and the weights $\boldsymbol{w}$ satisfying $\|D_k\|_F \leq \epsilon$ in at most $(\boldsymbol{F}(\boldsymbol{U}_0, \boldsymbol{w}_0) - \boldsymbol{F}^*)/\overline{\beta}\epsilon^2$ iterations.*

Proof. The proof can be followed by the convergence analysis on ManPL (4.1). □

References

[AMS08] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton, NJ: Princeton University Press, 2008, pp. xvi+224. ISBN: 978-0-691-13298-3.

[BA11] Nicolas Boumal and Pierre-antoine Absil. "RTRMC: A Riemannian trust-region method for low-rank matrix completion". In: *Advances in neural information processing systems*. 2011, pp. 406–414.

[Bec17] Amir Beck. *First-order methods in optimization*. Vol. 25. SIAM, 2017.

[BK16] Rhonda Bacher and Christina Kendziorski. "Design and computational analysis of single-cell RNA-sequencing experiments". In: *Genome biology* 17.1 (2016), p. 63.

[Bre+13] Philip Brennecke et al. "Accounting for technical noise in single-cell RNA-seq experiments". In: *Nature methods* 10.11 (2013), p. 1093.

[BST14] Jérôme Bolte, Shoham Sabach, and Marc Teboulle. "Proximal alternating linearized minimization for nonconvex and nonsmooth problems". In: *Mathematical Programming* 146.1-2 (2014), pp. 459–494.

[Bue+15] Florian Buettner et al. "Computational analysis of cell-to-cell heterogeneity in single-cell RNA-sequencing data reveals hidden subpopulations of cells". In: *Nature biotechnology* 33.2 (2015), p. 155.

[Cha+19] Vasileios Charisopoulos et al. "Composite optimization for robust blind deconvolution". In: *arXiv preprint arXiv:1901.01624* (2019).

[Che+18] Shixiang Chen et al. "Proximal Gradient Method for Manifold Optimization". In: *arXiv preprint arXiv:1811.00980* (2018).

[Che+19] Shixiang Chen et al. "An Alternating Manifold Proximal Gradient Method for Sparse PCA and Sparse CCA". In: (2019), pp. 1–28. URL: http://arxiv.org/abs/1903.11576.

[Den+14] Qiaolin Deng et al. "Single-cell RNA-seq reveals dynamic, random monoallelic gene expression in mammalian cells". In: *Science* 343.6167 (2014), pp. 193–196.

[DG17] Dheeru Dua and Casey Graff. *UCI Machine Learning Repository*. 2017. URL: http://archive.ics.uci.edu/ml.

[DP18] Dmitriy Drusvyatskiy and Courtney Paquette. "Efficiency of minimizing compositions of convex functions and smooth maps". In: *Mathematical Programming* (2018), pp. 1–56.

[DR17] John C Duchi and Feng Ruan. "Solving (most) of a set of quadratic equalities: Composite optimization for robust phase retrieval". In: *arXiv preprint arXiv:1705.02356* (2017).

[FHT01] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*. Vol. 1. 10. Springer series in statistics New York, 2001.

[KAH19] Vladimir Yu Kiselev, Tallulah S Andrews, and Martin Hemberg. "Challenges in unsupervised clustering of single-cell RNA-seq data". In: *Nature Reviews Genetics* (2019), p. 1.

[Lu+18] Canyi Lu et al. “Nonconvex Sparse Spectral Clustering by Alternating Direction Method of Multipliers and Its Convergence Analysis”. In: *AAAI Conference on Artificial Intelligence (AAAI)*. 2018.

[LYL16] Canyi Lu, Shuicheng Yan, and Zhouchen Lin. “Convex Sparse Spectral Clustering: Single-view to Multi-view”. In: *IEEE Transactions on Image Processing (TIP)* 25.6 (2016), pp. 2833–2843.

[Mif77] Robert Mifflin. “Semismooth and semiconvex functions in constrained optimization”. In: *SIAM Journal on Control and Optimization* 15.6 (1977), pp. 959–972.

[NJW02] Andrew Y Ng, Michael I Jordan, and Yair Weiss. “On spectral clustering: Analysis and an algorithm”. In: *Advances in neural information processing systems*. 2002, pp. 849–856.

[Pol+14] Alex A Pollen et al. “Low-coverage single-cell mRNA sequencing reveals cellular heterogeneity and activated signaling pathways in developing cerebral cortex”. In: *Nature biotechnology* 32.10 (2014), p. 1053.

[PZ18] Seyoung Park and Hongyu Zhao. “Spectral clustering based on learning similarity matrix”. In: *Bioinformatics* 34.12 (2018), pp. 2069–2076.

[QS93] Liqun Qi and Jie Sun. “A nonsmooth version of Newton’s method”. In: *Mathematical programming* 58.1-3 (1993), pp. 353–367.

[Sch+15] Andreas Schlitzer et al. “Identification of cDC1-and cDC2-committed DC progenitors reveals early lineage priming at the common DC progenitor stage in the bone marrow”. In: *Nature immunology* 16.7 (2015), p. 718.

[Tin+14] David T Ting et al. “Single-cell RNA sequencing identifies extracellular matrix gene expression by pancreatic circulating tumor cells”. In: *Cell reports* 8.6 (2014), pp. 1905–1918.

[Tre+14] Barbara Treutlein et al. “Reconstructing lineage hierarchies of the distal lung epithelium using single-cell RNA-seq”. In: *Nature* 509.7500 (2014), p. 371.

[WST10] Chengjing Wang, Defeng Sun, and Kim-Chuan Toh. “Solving log-determinant optimization problems by a Newton-CG primal proximal point algorithm”. In: *SIAM Journal on Optimization* 20.6 (2010), pp. 2994–3013.

[Xia+18] Xiantao Xiao et al. “A regularized semi-smooth Newton method with projection steps for composite convex programs”. In: *Journal of Scientific Computing* (2018), pp. 1–26.

[YST13] Junfeng Yang, Defeng Sun, and Kim-Chuan Toh. "A proximal point algorithm for log-determinant optimization with group Lasso regularization". In: *SIAM Journal on Optimization* 23.2 (2013), pp. 857–893.

[ZHT06] Hui Zou, Trevor Hastie, and Robert Tibshirani. "Sparse principal component analysis". In: *Journal of computational and graphical statistics* 15.2 (2006), pp. 265–286.

[ZST10] Xin-Yuan Zhao, Defeng Sun, and Kim-Chuan Toh. "A Newton-CG augmented Lagrangian method for semidefinite programming". In: *SIAM Journal on Optimization* 20.4 (2010), pp. 1737–1765.