INFORMS JOURNAL ON OPTIMIZATION
Vol. 00, No. 0, Xxxxx 0000, pp. 000–000
ISSN XXXX-XXXX | EISSN XXXX-XXXX | 00 | 0000 | 0001

INFORMS
DOI 10.1287/xxxx.0000.0000

A Manifold Proximal Linear Method for Sparse Spectral Clustering with Application to Single-Cell RNA Sequencing Data Analysis*

Zhongruo Wang*
Department of Mathematics, One Shields Ave, University of California, Davis, CA 95616 USA, zrnwang@ucdavis.edu

Bingyuan Liu*
Department of Statistics, 418 Thomas Building, Penn State University, University Park, PA 16802 USA, bul37@psu.edu

Shixiang Chen
Department of Industrial & Systems Engineering, Texas A&M University, 101 Bizzell St, College Station, TX 77843 USA, sxchen@tamu.edu

Shiqian Ma
Department of Mathematics, One Shields Ave, University of California, Davis, CA 95616 USA, sqma@ucdavis.edu

Lingzhou Xue
Department of Statistics, 318 Thomas Building, Penn State University, University Park, PA 16802 USA, lzxue@psu.edu

Hongyu Zhao
Department of Biostatistics, Yale University, 60 College Street, New Haven, CT, 06520 USA, hongyu.zhao@yale.edu

Spectral clustering is one of the fundamental unsupervised learning methods and is widely used in data analysis. Sparse spectral clustering (SSC) imposes sparsity to the spectral clustering and it improves the interpretability of the model. One widely adopted model for SSC in the literature is an optimization problem over the Stiefel manifold with nonsmooth and nonconvex objective. Such an optimization problem is very challenging to solve. Existing methods usually solve its convex relaxation or need to smooth its nonsmooth objective using certain smoothing techniques. Therefore, they were not targeting to solve the original formulation of SSC. In this paper, we propose a manifold proximal linear method (ManPL) that solves the original SSC formulation, without twisting the model. We also extend the algorithm to solve the multiple-kernel SSC problems, for which an alternating ManPL algorithm is proposed. Convergence and iteration complexity results of the proposed methods are established. We demonstrate the advantage of our proposed methods over existing methods via clustering of several datasets including UCI and single-cell RNA sequencing datasets.

Key words: Riemannian Optimization, Manifold Proximal Linear Method, Sparse Spectral Clustering, Single-Cell RNA Sequencing Data Analysis

1. Introduction

Clustering is a fundamental unsupervised learning problem with wide applications. The hierarchical clustering, K-means clustering and spectral clustering (SC) methods are widely used in practice (Friedman et al. 2001). It is known that interpretation of the dendrogram in hierarchical clustering can be difficult in practice, especially for large datasets. The K-means clustering, closely related to Lloyd's algorithm, does not guarantee to find the optimal solution and performs poorly for non-linearly separable or non-convex clusters. SC is a graph-based clustering method and it provides a promising alternative for identifying locally connected clusters (Chung and Graham 1997, Shi and Malik 2000, Ng et al. 2002).

Given the data matrix $X = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n] \in \mathbb{R}^{p\times n}$, where n is the number of data points and p is the feature dimension, SC constructs a symmetric affinity matrix $S = (s_{ij})_{n\times n}$, where $s_{ij} \geq 0$

* Zhongruo Wang and Bingyuan Liu contributed equally to this paper.

Corresponding authors: Shiqian Ma (sqma@ucdavis.edu) and Lingzhou Xue (lzxue@psu.edu)

measures the pairwise similarity between two samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ for $i,j=1,\ldots,n$. Denote diagonal matrix $D=\mathrm{Diag}\,(d_1,\ldots,d_n)$ with $d_i=\sum_{j=1}^{n} s_{ij}$. The main step of SC is to compute the following eigenvalue decomposition:

$$\min_{U\in\mathbb{R}^{n\times C}} \langle UU^\top, L\rangle,\ \text{s.t.},\ U^\top U = I_C, \tag{1.1}$$

where $L=I_n-D^{-1/2}SD^{-1/2}$ is the normalized Laplacian matrix, I_C denotes the $C\times C$ identity matrix, and C is the number of clusters. The rows of U can be regarded as an embedding of the data X from $\mathbb{R}^p$ to $\mathbb{R}^C$. The cluster assignment is then decided after using a standard clustering method such as the K-means clustering on the estimated embedding matrix $\hat{U}$ obtained by solving (1.1). Ideally, $\hat{U}$ should be a sparse matrix such that $\hat{U}_{ij}\neq 0$ if and only if sample i belongs to the j-th cluster. Therefore $\hat{U}\hat{U}^\top$ should be a block diagonal matrix which is also sparse. To this end, the sparse spectral clustering (SSC) (Lu et al. 2016, 2018, Park and Zhao 2018) is proposed to impose sparsity on $\hat{U}\hat{U}^\top$, which leads to the following optimization problem:

$$\min_{U\in\mathbb{R}^{n\times C}} \langle UU^\top, L\rangle + \lambda\|UU^\top\|_1,\ \text{s.t.},\ U^\top U = I_C, \tag{1.2}$$

where $\|Z\|_1=\sum_{ij}|Z_{ij}|$ is the entry-wise ℓ_1 norm of Z and it promotes the sparsity of Z, and $\lambda>0$ is a weighting parameter.

In practice, the performance of SSC is sensitive to a single measure of similarity between data points, and there are no clear criteria to choose an optimal similarity measure. Moreover, for some very complex data such as the single-cell RNA sequencing (scRNA-seq) data (Kiselev et al. 2019), one may benefit from considering multiple similarity matrices because they provide more information to the data. The next-generation sequencing technologies provide large detailed catalogs of the transcriptomes of massive cells to identify putative cell types. Clustering high-dimensional scRNA-seq data provides an informative step to disentangle the complex relationship between different cell types. For example, it is important to characterize the patterns of monoallelic gene expression across mammalian cell types (Deng et al. 2014), explore the mechanisms that control the progression of lung progenitors across distinct cell types (Treutlein et al. 2014), or study the

functionally distinct lineage in the bone marrow across mouse conventional dendritic cell types (Schlitzer et al. 2015). To this end, Park and Zhao (2018) suggest the following similarity matrices which lead to multiple-kernel SSC (MKSSC):

$$K_{\delta,m}(i,j) = \exp\left(\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\epsilon_{ij}^2}\right), \epsilon_{ij} = \frac{\delta(\mu_i + \mu_j)}{2}, \mu_i = \frac{\sum_{\ell \in \mathrm{KNN}(i)} \|\boldsymbol{x}_i - \boldsymbol{x}_\ell\|}{m},$$

where $\mathrm{KNN}(i)$ represents a set of sample indices that are the top m nearest neighbors of the sample $\boldsymbol{x}_i$. The parameters δ and m control the width of the neighborhoods. We use $S(\delta)$ and $S(m)$ to denote the sets of possible choices of δ and m, respectively. Then the total number of similarity matrices is equal to $T = |S(\delta)| \cdot |S(m)|$. We denote the normalized Laplacian matrices corresponding to these T similarity matrices as $L^{(\ell)}$, $\ell = 1, \ldots, T$. The MKSSC can be formulated as the following optimization problem:

$$\begin{aligned} \min_{U \in \mathbb{R}^{n \times C}, w \in \mathbb{R}^T} \quad & \bar{F}(U, w) \equiv \left\langle UU^\top, \sum_{\ell=1}^T w_\ell L^{(\ell)} \right\rangle + \lambda \|UU^\top\|_1 + \rho \sum_{\ell=1}^T w_\ell \log(w_\ell) \\ \text{s.t.} \quad & U^\top U = I_C, \sum_{\ell=1}^T w_\ell = 1, w_\ell \geq 0, \ell = 1, \ldots, T, \end{aligned} \tag{1.3}$$

where $w_\ell, \ell = 1, \ldots, T$ are unknown weightings of the kernels, and $\rho \sum_{\ell=1}^T w_\ell \log(w_\ell)$ serves as an entropy regularization term, and λ, ρ are two regularization parameters.

Note that both SSC (1.2) and MKSSC (1.3) are nonconvex and nonsmooth with Riemannian manifold constraints. Therefore, they are both numerically challenging to solve. In this paper, we propose a manifold proximal linear (ManPL) method for solving the SSC (1.2), and an alternating ManPL (AManPL) method for solving the MKSSC (1.3).

Our contributions lie in several folds.

(i) We propose the ManPL method for solving SSC (1.2) and the AManPL method for solving MKSSC (1.3). To the best of our knowledge, they are the first algorithms solving directly the original formulations of SSC (1.2) and MKSSC (1.3), without twisting the models.

(ii) We analyze the convergence and iteration complexity of both ManPL and AManPL. Though the two algorithms are closely related to the manifold proximal gradient (ManPG) algorithm

(Chen et al. 2020b), ManPL and AManPL deal with more complicated problems where the objective function involves a composition of a nonsmooth function and a smooth and nonconvex mapping. As a result, the analysis for ManPL and AManPL is different from that of ManPG and is new in the literature.

(iii) The subproblem in ManPL and AManPL is convex, but it is challenging to solve for large-scale problems. We propose a proximal point algorithm based on a semi-smooth Newton method for solving this subproblem.

(iv) We apply our proposed methods to clustering of scRNA-seq data. Our numerical experiments indicate that our AManPL algorithm is very suitable for clustering this type of data, and its advantage over existing methods is very clear.

The rest of this paper is organized as follows. We propose our ManPL method for solving the SSC (1.2) in Section 2. We provide details of the proximal point method and semi-smooth Newton method for solving the subproblem of ManPL in Section 3. We propose our AManPL method for solving the MKSSC (1.3) in Section 4. The numerical results using the proposed methods for solving clustering problems for UCI data, synthetic data, scRNA-seq data, and some unsupervised data are reported in Section 5. Finally, we draw some concluding remarks in Section 6. Proofs to the technical results are provided in the Appendix.

Notation. Throughout this paper, we use $\mathcal{M}$ to denote the Stiefel manifold. The smoothness, convexity, and Lipschitz continuity of a function f are always interpreted as the function is considered in the ambient Euclidean space. We use $\mathbb{S}^n_+$ to denote the set of $n \times n$ positive semidefinite matrices, and $\mathrm{Tr}(Z)$ to denote the trace of matrix Z.

2. A Manifold Proximal Linear Method for SSC

Since SSC (1.2) is both nonsmooth and nonconvex, it is numerically challenging to solve. In the literature, convex relaxations and smooth approximations of (1.2) have been suggested. In particular, Lu et al. (2016) proposed to replace $UU^\top$ with a positive semidefinite matrix P and solve the following convex relaxation:

$$\min_{P \in \mathbb{S}^n_+} \ \langle P, L\rangle + \lambda \|P\|_1, \ \text{s.t.}, \ 0 \preceq P \preceq I, \ \mathrm{Tr}(P) = C. \tag{2.1}$$

This convex problem (2.1) can be solved by classical optimization algorithms such as ADMM. Denote the solution of (2.1) by $\hat{P}$, the solution of (1.2) can be approximated by the top C eigenvectors of $\hat{P}$. In another work, Lu et al. (2018) proposed a nonconvex ADMM to solve the following smooth variant of (1.2):

$$\min_{U\in\mathbb{R}^{n\times C},P\in\mathbb{S}^n_+} \langle UU^\top, L\rangle + g_\sigma(P),\ \text{s.t.},\ P=UU^\top, U^\top U=I_C, \tag{2.2}$$

where $g_\sigma(\cdot)$ is a smooth function with smoothing parameter $\sigma>0$ that approximates the ℓ_1 regularizer $\lambda\|\cdot\|_1$. In Lu et al. (2018), the authors used the following smooth function:

$$g_\sigma(P) := \max_Z\ \langle P, Z\rangle - \frac{\sigma}{2}\|Z\|_F^2,\ \text{s.t.},\ \|Z\|_\infty\le\lambda, \tag{2.3}$$

where $\|Z\|_\infty = \max_{ij}|Z_{ij}|$. The nonconvex ADMM for solving (2.2) typically iterates as

$$U^{k+1} := \operatorname*{arg\,min}_{U\in\mathbb{R}^{n\times C}}\ \mathcal{L}(U, P^k; \Lambda^k),\ \text{s.t.},\ U^\top U = I_C, \tag{2.4a}$$

$$P^{k+1} := \operatorname*{arg\,min}_{P\in\mathbb{S}^n_+}\ \mathcal{L}(U^{k+1}, P; \Lambda^k), \tag{2.4b}$$

$$\Lambda^{k+1} := \Lambda^k - \mu(P^{k+1} - U^{k+1}{U^{k+1}}^\top), \tag{2.4c}$$

where the augmented Lagrangian function $\mathcal{L}$ is defined as

$$\mathcal{L}(U,P;\Lambda) := \langle UU^\top, L\rangle + g_\sigma(P) - \langle\Lambda, P-UU^\top\rangle + \frac{\mu}{2}\|P-UU^\top\|_F^2,$$

and $\mu>0$ is a penalty parameter. The two subproblems (2.4a) and (2.4b) are both relatively easy to solve. The reason to use the smooth function $g_\sigma(\cdot)$ to approximate $\lambda\|\cdot\|_1$ in (2.2) is for the purpose of convergence guarantee. In Lu et al. (2018), the authors proved that any limit point of the sequence generated by the nonconvex ADMM (2.4) is a stationary point of (2.2). This result relies on the fact that function g_σ is smooth. If one applies ADMM to the original SSC (1.2), then no convergence guarantee is known.

In this section, we introduce our ManPL algorithm that solves the original SSC (1.2) directly, and unlike (2.1) and (2.2), our ManPL algorithm does not twist the formulation. For the ease of presentation, we rewrite (1.2) as

$$\min_U\ F(U) \equiv f(U) + h(c(U)),\ \text{s.t.},\ U\in\mathcal{M}, \tag{2.5}$$

where $f(U) = \langle UU^\top, L\rangle$, $h(\cdot) = \lambda\|\cdot\|_1$, $c(U) = UU^\top$, $\mathcal{M} = \{U \in \mathbb{R}^{n\times C} \mid U^\top U = I_C\}$ is the Stiefel manifold. Moreover, note that f and c are smooth mappings, and h is nonsmooth but convex in the ambient Euclidean space. Therefore, (2.5) is a Riemannian optimization problem with nonsmooth and nonconvex objective function. Furthermore, throughout this paper, we use L_f, L_c, L_h to denote the Lipschitz constants of ∇f, ∇c, and h, respectively. Riemannian optimization has drawn much attention recently, due to its wide applications, including low rank matrix completion (Boumal and Absil 2011), phase retrieval (Bendory et al. 2018, Sun et al. 2018), phase synchronization (Boumal 2016, Liu et al. 2017), and dictionary learning (Cherian and Sra 2017, Sun et al. 2017). Several important classes of algorithms for Riemannian optimization with a smooth objective function were covered in the monograph (Absil et al. 2008). On the other hand, there has been very limited number of algorithms for Riemannian optimization with nonsmooth objective until very recently. The most natural idea for this class of optimization problems is the Riemannian subgradient method (RSGM) (Ferreira and Oliveira 1998, Grohs and Hosseini 2016, Hosseini and Uschmajew 2017). Recently, Li et al. (2021) studied the RSGM for Riemannian optimization with weakly convex objective. In particular, they showed that the number of iterations needed by RSGM for obtaining an ϵ-stationary point is $O(\epsilon^{-4})$. Motivated by the proximal gradient method for solving composite minimization in Euclidean space, Chen et al. (2020b) proposed a manifold proximal gradient method (ManPG) for solving the following Riemannian optimization problem:

$$\min_X \ f(X) + h(X), \ \text{s.t.}, \ X \in \mathcal{M}, \tag{2.6}$$

where $\mathcal{M}$ is the Stiefel manifold, f is a smooth function, and h is a nonsmooth and convex function. A typical iteration of ManPG for solving (2.6) is:

$$V^k := \arg\min_V \ \langle \nabla f(X^k), V\rangle + h(X^k + V) + \frac{1}{2t}\|V\|_F^2, \ \text{s.t.}, \ V \in \mathrm{T}_{X^k}\mathcal{M}, \tag{2.7a}$$

$$X^{k+1} := \mathrm{Retr}_{X^k}(\alpha_k V^k), \tag{2.7b}$$

where $\alpha_k > 0$ is a step size. Here $\mathrm{T}_U\mathcal{M}$ denotes the tangent space of $\mathcal{M}$ at U, and for the Stiefel manifold, it is known that $\mathrm{T}_U\mathcal{M} := \{V \in \mathbb{R}^{n\times C} \mid V^\top U + U^\top V = 0\}$. Moreover, Retr denotes the retraction operation, whose definition is given below.

DEFINITION 1. A retraction on a differentiable manifold $\mathcal{M}$ is a smooth mapping Retr from the tangent bundle $\mathrm{T}\mathcal{M}$ onto $\mathcal{M}$ satisfying the following two conditions (here Retr_X denotes the restriction of Retr onto $\mathrm{T}_X\mathcal{M}$)

- $\mathrm{Retr}_X(0) = X$, $\forall X \in \mathcal{M}$, where 0 denotes the zero element of $\mathrm{T}_X\mathcal{M}$.
- For $X \in \mathcal{M}$, it holds that

$$\lim_{\mathrm{T}_X\mathcal{M}\ni\xi\to 0} \frac{\|\mathrm{Retr}_X(\xi) - (X+\xi)\|_F}{\|\xi\|_F} = 0.$$

The retraction onto the Euclidean space is simply the identity mapping: $\mathrm{Retr}_X(\xi) = X + \xi$. Common retractions include the polar decomposition:

$$\mathrm{Retr}_X^{\mathrm{polar}}(\xi) = (X+\xi)(I_r + \xi^\top\xi)^{-1/2},$$

the QR decomposition:

$$\mathrm{Retr}_X^{\mathrm{QR}} = \mathbf{qf}(X+\xi),$$

where $\mathbf{qf}(A)$ is the Q factor of the QR factorization of A, and the Cayley transformation:

$$\mathrm{Retr}_X^{\mathrm{cayley}}(\xi) = \Big(I_n - \frac{1}{2}W(\xi)\Big)^{-1}\Big(I_n + \frac{1}{2}W(\xi)\Big)X,$$

where $W(\xi) = (I_n - \frac{1}{2}XX^\top)\xi X^\top - X\xi^\top(I_n - \frac{1}{2}XX^\top)$.

Comparing with (2.6), we note that (2.5) is more difficult to solve, because of the nonconvex term $c(U)$. In fact, ManPG cannot be used to solve the SSC (1.2) because of the existence of the nonconvex term $UU^\top$ composite with the ℓ_1 norm. As a result, a new algorithm is demanded for solving SSC (1.2). The iteration complexity of ManPG is proved to be $O(\epsilon^{-2})$ for obtaining an ϵ-stationary point of (2.6) (Chen et al. 2020b), which is better than the complexity of RSGM (Li et al. 2021). Variants of ManPG have been designed for different applications, such as alternating ManPG for sparse PCA and sparse CCA (Chen et al. 2020c), FISTA for sparse PCA (Huang and Wei 2019), manifold proximal point algorithm for robust subspace recovery and orthogonal dictionary learning (Chen et al. 2019, 2020a), and stochastic ManPG (Wang et al. 2020) for online sparse PCA. Moreover, ManPG has been extended to more general Riemannian proximal gradient

method (Huang and Wei 2021). Motivated by the success of ManPG and its variants, we propose a manifold proximal linear algorithm for solving SSC (2.5).

The proximal linear method has recently drawn great research attentions. It targets to solve the optimization problem in the form of (2.5) without the manifold constraint, i.e.,

$$\min_{x\in\mathbb{R}^n} \ f(x)+h(c(x)), \tag{2.8}$$

where $f:\mathbb{R}^n\to\mathbb{R}$ and $c:\mathbb{R}^n\to\mathbb{R}^m$ are smooth mappings, $h:\mathbb{R}^m\to\mathbb{R}$ is convex and nonsmooth. The proximal linear method for solving (2.8) iterates as follows:

$$x^{k+1}:=\arg\min_x \ \langle\nabla f(x^k),x-x^k\rangle+h(c(x^k)+J(x^k)(x-x^k))+\frac{1}{2t}\|x-x^k\|_2^2, \tag{2.9}$$

where $J(x)=\nabla c(x)$ is the Jacobian of c, and $t>0$ is a step size. Note that since h is convex, the update (2.9) is a convex problem. This method has been studied recently by Lewis and Wright (2016), Drusvyatskiy and Paquette (2019), Duchi and Ruan (2018) and applied to solving many important applications such as robust phase retrieval (Duchi and Ruan 2019), robust matrix recovery (Charisopoulos et al. 2019), and robust blind deconvolution (Charisopoulos et al. 2021).

Due to the nonconvex constraint $U\in\mathcal{M}$, solving (2.5) is more difficult than (2.8). Motivated by ManPG and the proximal linear method (2.9), we propose a ManPL algorithm for solving (2.5). A typical iteration of the ManPL algorithm for solving (2.5) is:

$$V^k:=\arg\min_V \ \langle\nabla f(U^k),V\rangle+h(c(U^k)+J(U^k)V)+\frac{1}{2t}\|V\|_F^2, \text{ s.t., } V\in\mathrm{T}_{U^k}\mathcal{M}, \tag{2.10a}$$

$$U^{k+1}:=\mathrm{Retr}_{U^k}(\alpha_k V^k). \tag{2.10b}$$

Similar to (2.7a), the equation (2.10a) computes the descent direction V by minimizing a convex function over the tangent space of $\mathcal{M}$. However, solving (2.10a) is more difficult than (2.7a) because of the non-trivial affine function, i.e., $c(U^k)+J(U^k)V$, composite with the nonsmooth function h. Moreover, the difference of (2.10a) and (2.9) is the constraint in (2.10a), which is needed in the Riemannian optimization setting. Fortunately, (2.10a) can still be solved efficiently by a proximal

point algorithm combined with a semi-smooth Newton method, which will be elaborated in Section 3. The retraction step (2.10b) brings the iterate back to the manifold $\mathcal{M}$.

The complete description of the ManPL for solving SSC (2.5) is given in Algorithm 1. The step (2.11) is a line search step to find the step size α_k such that there is a sufficient decrease on the function F.

Algorithm 1 The ManPL for SSC (2.5)

Input: initial point $U^0 \in \mathcal{M}$, parameters $\gamma \in (0,1)$, $t > 0$

for $k = 0, 1, \ldots$ **do**

Calculate V^k by solving (2.10a)

Let j_k be the smallest non-negative integer such that

$$F(\mathrm{Retr}_{U^k}(\gamma^{j_k} V^k)) \leq F(U^k) - \frac{\gamma^{j_k}}{2t}\|V^k\|_F^2 \tag{2.11}$$

Let $\alpha_k = \gamma^{j_k}$ and compute U^{k+1} by (2.10b)

end for

The main convergence and iteration complexity result of ManPL (Algorithm 1) is given in Theorem 1. Its proof is given in the appendix.

THEOREM 1. *Assume $F(U)$ is lower bounded by F^*. The limit point of the sequence $\{U^k\}$ generated by ManPL (Algorithm 1) is a stationary point of* (2.5)*. Moreover, ManPL returns an ϵ-stationary point of* (2.5) *in $O(\epsilon^{-2})$ iterations.*

3. A Semi-Smooth Newton-based Proximal Point Algorithm for the Subproblem

In this section, we introduce a proximal point algorithm (PPA) combined with a semi-smooth Newton method (SSN) for solving the subproblem (2.10a) in ManPL. The notion of semi-smoothness was originally introduced by (Mifflin 1977) for real valued functions and later extended to vector-valued mappings by (Qi and Sun 1993). 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 (Li et al. 2018, Yang et al. 2013), convex clustering (Wang et al. 2010), SDP (Zhao et al. 2010), and convex composite problems (Xiao et al. 2018).

For simplicity of the notation, we omit the index k in (2.10a), and denote $Z_1 := \nabla f(U^k)$, $Z_2 := c(U^k)$, $J = J(U^k)$, and operator $\mathcal{A}: V \to V^\top U^k + (U^k)^\top V$. Therefore, (2.10a) reduces to the following form:

$$\min_{V,Y} \ \frac{1}{2t}\|V + Z_1\|_F^2 + h(Z_2 + Y), \ \text{s.t.}, \ \mathcal{A}(V) = 0, \ Y = JV. \tag{3.1}$$

Note that we have introduced a variable Y to replace JV. The Lagrangian function for (3.1) is given by:

$$\mathcal{L}(V, Y; \Gamma_1, \Gamma_2) = \frac{1}{2t}\|V + Z_1\|_F^2 + h(Z_2 + Y) - \langle \Gamma_1, \mathcal{A}(V)\rangle - \langle \Gamma_2, JV - Y\rangle,$$

where Γ_1 and Γ_2 are the Lagrange multipliers associated to the two equality constraints. Therefore, (3.1) is equivalent to

$$\min_{V,Y}\{G(V, Y) := \max_{\Gamma_1,\Gamma_2} \mathcal{L}(V, Y; \Gamma_1, \Gamma_2)\}. \tag{3.2}$$

The minimization problem (3.2) can be solved by a PPA, which iterates as:

$$\begin{aligned}(V^{k+1}, Y^{k+1}) &:= \arg\min_{V,Y} G(V^k, Y^k) + \frac{1}{2\beta}\left(\|V - V^k\|_F^2 + \|Y - Y^k\|_F^2\right) \\ &:= \arg\min_{V,Y} \max_{\Gamma_1,\Gamma_2} \mathcal{L}(V, Y; \Gamma_1, \Gamma_2) + \frac{1}{2\beta}\left(\|V - V^k\|_F^2 + \|Y - Y^k\|_F^2\right),\end{aligned} \tag{3.3}$$

where $\beta > 0$ is a parameter. The problem (3.3) is equivalent to:

$$\max_{\Gamma_1,\Gamma_2} \min_{V,Y} \mathcal{L}(V, Y; \Gamma_1, \Gamma_2) + \frac{1}{2\beta}\left(\|V - V^k\|_F^2 + \|Y - Y^k\|_F^2\right). \tag{3.4}$$

Note that the minimization part of (3.4) is strongly convex and admits a closed-form solution given by:

$$V = \frac{t\beta}{t+\beta}\left(\mathcal{A}^*(\Gamma_1) + J^\top\Gamma_2 - \frac{1}{t}Z_1 + \frac{1}{\beta}V^k\right), \ Y = \text{Prox}_{\beta h}(Z_2 + Y^k - \beta\Gamma_2) - Z_2, \tag{3.5}$$

where Prox_g denotes the proximal mapping of function g, which is defined as:

$$\text{Prox}_g(Z) := \arg\min_X \ g(X) + \frac{1}{2}\|X - Z\|_F^2.$$

For simplicity of the notation, we define function $E(\Gamma_2) := Z_2 + Y^k - \beta\Gamma_2$. Substituting (3.5) to (3.4), and using the Moreau identity, we know that (3.4) is equivalent to:

$$\begin{aligned}\max_{\Gamma_1,\Gamma_2} \Theta(\Gamma_1,\Gamma_2) := &-\frac{1}{2}\frac{t\beta}{t+\beta}\left\|\frac{1}{t}Z_1 - \frac{1}{\beta}V^k - \mathcal{A}^*(\Gamma_1) - J^\top\Gamma_2\right\|_F^2 \\ &+ h(\mathrm{Prox}_{\beta h}E(\Gamma_2)) + \beta\|\mathrm{Prox}_{h^*/\beta}(E(\Gamma_2)/\beta)\|_F^2 + \langle\Gamma_2, Y^k\rangle - \frac{\beta}{2}\|\Gamma_2\|_F^2,\end{aligned} \tag{3.6}$$

where h^* denotes the conjugate function of h. Now by denoting

$$\Psi(\Gamma_2) := \max_{\Gamma_1} \Theta(\Gamma_1,\Gamma_2), \tag{3.7}$$

it is easy to verify that $\Psi(\Gamma_2)$ is strongly concave and continuously differentiable (Li et al. 2018), and its unique maximizer is found by solving the following nonsmooth system:

$$\nabla\Psi(\Gamma_2) = 0. \tag{3.8}$$

Solving (3.8) can be done by using SSN (Li et al. 2018, Xiao et al. 2018). After we obtain the solution to (3.8), the optimal Γ_1 can be found by solving the maximization problem in (3.7), which is an easy least-squares problem.

To summarize, the PPA for solving (2.10a) is given by (3.3), and its solution is given by (3.5). The required Γ_2 in (3.5) is obtained by solving (3.8) using SSN, and Γ_1 in (3.5) is obtained by solving the least-squares problem in (3.7). The convergence of the PPA and the SSN has been well studied in the literature (Li et al. 2018, Yang et al. 2013).

4. An Alternating ManPL Method for Multiple-Kernel SSC

In this section, we consider the multiple-kernel SSC (1.3). Park and Zhao (2018) consider to solve the following relaxation of (1.3) by letting $P = UU^\top$:

$$\begin{aligned}\min_{P,w} \quad &\left\langle P, \sum_{\ell=1}^T w_\ell L^{(\ell)} \right\rangle + \lambda\|P\|_1 + \rho\sum_{\ell=1}^T w_\ell \log(w_\ell) \\ \text{s.t.} \quad &\mathrm{Tr}(P) = C, 0 \preceq P \preceq I, \sum_{\ell=1}^T w_\ell = 1, w_\ell \geq 0, \ell = 1,\ldots,T.\end{aligned} \tag{4.1}$$

Note that this is still a nonconvex problem due to the bi-linear term in the objective function. Park and Zhao (2018) suggested to use an alternating minimization algorithm (AMA) to solve (4.1).

Note that this method is named MPSSC in (Park and Zhao 2018). In the k-th iteration of AMA, one first fixes w as w^k and solves the resulting problem with respect to P to obtain P^{k+1}, and then fixes P as P^{k+1} and solves the resulting problem with respect to w to obtain w^{k+1}. In particular, when w is fixed as w^k, problem (4.1) reduces to

$$\min_{P} \quad \left\langle P, \sum_{\ell=1}^{T} w_\ell^k L^{(\ell)} \right\rangle + \lambda \|P\|_1, \text{ s.t., } \operatorname{Tr}(P) = C, 0 \preceq P \preceq I, \tag{4.2}$$

which is a convex problem and can be solved via convex ADMM algorithm. When P is fixed as P^{k+1}, problem (4.1) reduces to

$$\min_{w} \quad c^\top w + \rho \sum_{\ell=1}^{T} w_\ell \log(w_\ell), \text{ s.t., } \sum_{\ell=1}^{T} w_\ell = 1, w_\ell \geq 0, \ell = 1, \ldots, T, \tag{4.3}$$

where $c_\ell = \langle P^{k+1}, L^{(\ell)} \rangle$, $\ell = 1, \ldots, T$. This is also a convex problem and it can be easily verified that (4.3) admits a closed-form solution given by

$$w_\ell = \frac{\exp(-c_\ell/\rho)}{\sum_{j=1}^{T} \exp(-c_j/\rho)}, \ \ell = 1, \ldots, T. \tag{4.4}$$

In summary, a typical iteration of the AMA algorithm proposed by (Park and Zhao 2018) is as follows:

$$\begin{cases} \text{update } P^{k+1} \text{ by solving (4.2)} \\ \text{update } w^{k+1} \text{ by solving (4.3).} \end{cases} \tag{4.5}$$

In our numerical experiments, we call this method AMA+CADMM, because (4.2) is solved by a convex ADMM.

Another approach to approximate (1.3) is to combine the idea of AMA (4.5) and the nonconvex ADMM for solving the smooth problem (2.2). In particular, one can solve the following smooth variant of (1.3):

$$\begin{aligned} \min_{U \in \mathbb{R}^{n \times C}, w \in \mathbb{R}^T} \quad & \left\langle UU^\top, \sum_{\ell=1}^{T} w_\ell L^{(\ell)} \right\rangle + g_\sigma(UU^\top) + \rho \sum_{\ell=1}^{T} w_\ell \log(w_\ell) \\ \text{s.t.} \quad & U^\top U = I_C, \sum_{\ell=1}^{T} w_\ell = 1, w_\ell \geq 0, \ell = 1, \ldots, T, \end{aligned} \tag{4.6}$$

where $g_\sigma(\cdot)$ is the smooth approximation to $\lambda\|\cdot\|_1$ defined in (2.3). When fixing w, (4.6) is in the same form as the smoothed SSC (2.2), so it can be solved by the nonconvex ADMM (2.4). When

fixing U, (4.6) is in the same form as (4.3), and admits a closed-form solution (4.4). In summary, the AMA+ADMM algorithm for solving (4.6) works as follows:

$$\begin{cases} \text{update } U^{k+1} \text{ by solving (4.6) with } w \text{ fixed as } w^k \text{ using nonconvex ADMM (2.4)} \\ \text{update } w^{k+1} \text{ by solving (4.6) with } U \text{ fixed as } U^{k+1} \text{ using (4.4).} \end{cases} \tag{4.7}$$

To differentiate with the AMA+CADMM algorithm (4.5), we call (4.7) the AMA+NADMM, because a nonconvex ADMM is used to solve (4.6) with w fixed as w^k.

By exploiting the structure of (1.3), we propose to solve (1.3) by an alternating ManPL algorithm (AManPL). More specifically, in the k-th iteration of AManPL, we first fix w as w^k, then (1.3) reduces to

$$\min_U \left\langle UU^\top, \sum_{\ell=1}^T w_\ell^k L^{(\ell)} \right\rangle + \lambda \|UU^\top\|_1, \text{ s.t., } U \in \mathcal{M}, \tag{4.8}$$

which is in the same form of (2.5) with L in (2.5) replaced by $\bar{L} := \sum_{\ell=1}^T w_\ell^k L^{(\ell)}$. Therefore, (4.8) can also be solved by ManPL. Here we adopt one step of ManPL, i.e., (2.10) to obtain U^{k+1}. More specifically, U^{k+1} is computed by the following two steps:

$$V^k := \arg\min_V \ \langle \nabla_U f(U^k, w^k), V\rangle + h(c(U^k) + J(U^k)V) + \frac{1}{2t}\|V\|_F^2, \text{ s.t., } V \in \mathrm{T}_{U^k}\mathcal{M}, \tag{4.9a}$$

$$U^{k+1} := \mathrm{Retr}_{U^k}(\alpha_k V^k), \tag{4.9b}$$

where $f(U,w) := \left\langle UU^\top, \sum_{\ell=1}^T w_\ell L^{(\ell)} \right\rangle$, $h(\cdot) := \lambda\|\cdot\|_1$, and $c(U) = UU^\top$. Note that (4.9a) can be solved by the same PPA+SSN algorithm discussed in Section 3. We then fix U in (1.3) as U^{k+1}, and then (1.3) reduces to

$$\min_w \ c^\top w + \rho \sum_{\ell=1}^T w_\ell \log(w_\ell), \text{ s.t., } \sum_{\ell=1}^T w_\ell = 1, w_\ell \geq 0, \ell = 1, \ldots, T, \tag{4.10}$$

where $c_\ell = \langle U^{k+1}{U^{k+1}}^\top, L^{(\ell)}\rangle$, $\ell = 1, \ldots, T$. We then obtain w^{k+1} by solving (4.10), which admits a closed-form solution given by (4.4). The AManPL is described in Algorithm 2.

We have the following convergence and iteration complexity result for AManPL for solving MKSSC (1.3). Its proof is given in the appendix.

Algorithm 2 The AManPL Method for Solving MKSSC (1.3)

Input: parameter $\gamma \in (0,1)$, initial point $U^0 \in \mathcal{M}$, let w^0 be the optimal solution to (4.10) for $c_\ell = \langle U^0 {U^0}^\top, L^{(\ell)} \rangle$

for $k = 0, 1, \ldots$ **do**

Calculate V^k by solving (4.9a)

Let j_k be the smallest non-negative integer such that

$$\bar{F}(\mathrm{Retr}_{U^k}(\beta^{j_k} V^k), w^k) \leq \bar{F}(U^k, w^k) - \frac{\gamma^{j_k}}{2t} \|V^k\|_F^2 \tag{4.11}$$

Let $\alpha_k = \gamma^{j_k}$ and compute U^{k+1} by (4.9b)

Update w_ℓ^{k+1} by (4.4) with $c_\ell = \langle U^{k+1} {U^{k+1}}^\top, L^{(\ell)} \rangle$, $\ell = 1, \ldots, T$

end for

THEOREM 2. *Assume $\bar{F}(U, w)$ in* (1.3) *is lower bounded by $\bar{F}^*$. The limit point of the sequence $\{U^k, w^k\}$ generated by AManPL (Algorithm 2) is a stationary point of problem* (1.3)*. Moreover, to obtain an ϵ-stationary point of problem* (1.3)*, the number of iterations needed by AManPL is $O(\epsilon^{-2})$.*

5. Numerical Experiments

In this section, we compare our proposed methods ManPL and AManPL with some existing methods for solving SSC and MKSSC. In particular, for SSC (1.2), we compare ManPL (Algorithm 1) with convex ADMM (Lu et al. 2016) (denoted by CADMM [1]) for solving (2.1) and nonconvex ADMM (Lu et al. 2018) (denoted by NADMM) for solving (2.2). We also include the spectral clustering (denoted by SC) in the comparison. For MKSSC (1.3), we compare AManPL (Algorithm 2) with MPSSC (i.e., AMA+CADMM [2]) (Park and Zhao 2018) and AMA+NADMM (4.7). All the algorithms were terminated when the absolute change of the objective value is smaller than 10^{-5}, which indicates that the algorithms were not making much progress. All the codes were run

[1] cdoes downloaded from `https://github.com/canyilu/LibADMM/blob/master/algorithms/sparsesc.m`

[2] codes downloaded from `https://github.com/ishspsy/project/tree/master/MPSSC`

in Matlab R2021a on a laptop with a 1.61 GHz Intel 6-Core i7 CPU and 16GB RAM. All reported CPU times are in seconds.

Formulation	Method
Convex SSC (2.1)	CADMM (Lu et al. 2016)
Smoothed SSC (2.2)	NADMM (2.4) (Lu et al. 2018)
Original SSC (1.2)	ManPL (Algorithm 1)
MKSSC (4.1)	AMA+CADMM (4.5) (Park and Zhao 2018)
Smoothed MKSSC (4.6)	AMA+NADMM (4.7)
Original MKSSC (1.3)	AManPL (Algorithm 2)

Table 1 Summary of different methods for solving SSC or MKSSC.

5.1. UCI Datasets

We first compare the clustering performance of different methods on three benchmark datasets in UCI machine learning repository (Dua and Graff 2017). For the parameters used in the models, we choose them in the following manner. For σ that is used in (2.2) and (4.6), we set it as $\sigma = 0.2$. For λ that is used in all six models (1.2), (1.3), (2.1), (2.2), (4.1) and (4.6), and ρ that is used in (1.3), (4.1) and (4.6), we choose them from the following sets:

$$\lambda \in \{5\times 10^{-3}, 10^{-3}, 5\times 10^{-4}, 10^{-4}, 10^{-5}\}, \quad \text{and} \quad \rho \in \{0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1\}. \tag{5.1}$$

We follow Park and Zhao (2018) to construct the similarity matrices and record the Normalized Mutual Information (NMI) scores (Strehl and Ghosh 2003) to measure the performance of the clustering. Given two clustering assignments I and J on a set of n data points with C_I and C_J clusters, repectively, the NMI score is defined as

$$\mathrm{NMI}(I,J) = \frac{\sum_{p=1}^{C_I}\sum_{q=1}^{C_J} |I_p \cap J_q| \log \frac{n|I_p \cap J_q|}{|I_p|\times|J_q|}}{\max\left(-\sum_{p=1}^{C_I} |I_p| \log \frac{|I_p|}{n}, -\sum_{q=1}^{C_J} |J_q| \log \frac{|J_q|}{n}\right)}, \tag{5.2}$$

where the numerator is the mutual information between I and J, and the denominator represents the entropy of the clustering assignments I and J. Note that higher NMI scores indicate better

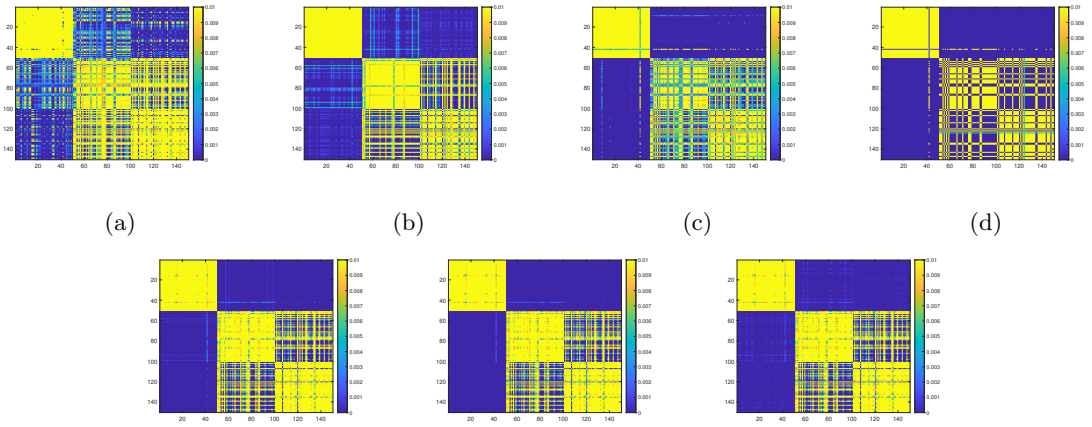
clustering performance. More specifically, based on the matrix $\hat{U}$ computed using the algorithm, we first perform K-means on $\hat{U}$ to get the resulting label I' for each data. Since the ground-truth assignments I of the clusters are available for the three datasets, we can then calculate the NMI score based on the resulting label I' and the ground-truth label I using (5.2).

Task	Method	Wine		Iris		Glass	
		NMI	CPU	NMI	CPU	NMI	CPU
SSC	CADMM	0.893	0.022	**0.742**	0.020	0.347	0.033
	NADMM	0.893	0.075	0.636	0.063	0.346	0.381
	ManPL	0.893	0.066	0.652	0.108	**0.407**	0.372
MKSSC	AMA+CADMM	**0.893**	0.965	0.804	0.784	0.353	1.842
	AMA+NADMM	**0.893**	1.230	**0.831**	0.357	0.357	4.761
	AManPL	0.882	0.578	0.804	0.658	**0.416**	2.630

Table 2 Comparison of NMI scores and CPU runtime for solving SSC or MKSSC on the UCI datasets.

The NMI scores are reported in Table 2. Note that we can compute the NMI scores because the ground-truth clustering assignments of these datasets are known. From Table 2 we see that for the Iris and Glass datasets, the MKSSC model always performs better than the SSC model in terms of NMI scores. For the Glass dataset, we see that our ManPL achieves better NMI score than CADMM and NADMM, and our AManPL achieves better NMI score than AMA+CADMM and AMA+NADMM. For the Wine dataset, we find that all algorithms perform similarly in terms of NMI scores, although AManPL achieves slightly worse NMI score. Moreover, we show the heatmap of $|UU^\top|$ for the Iris data set in Figure 1. For the SSC models (2.1), (2.2) and (1.2), we show the figures corresponding to $\lambda = 5 \times 10^{-3}$, and for the MKSSC models (4.1), (4.6) and (1.3), we show the figures corresponding to $\lambda = 10^{-4}, \rho = 2 \times 10^{-2}$. From Figure 1 we see that the figures generated by the SSC models, i.e., Figures 1 (b)-(d) give clearly better clustering results than the spectral clustering model (1.1), whose heatmap is given in Figure 1 (a). Furthermore, we also see that the

MKSSC models whose heatmaps are given in Figures 1 (e)-(g) give clearly better clustering results than the single view SSC models. These observations demonstrate the necessity of studying the multiple-kernel SSC models.



(a) (b) (c) (d)

(e) (f) (g)

Figure 1 **The heatmaps of $|UU^\top|$ on the Iris dataset estimated by (a) SC** (1.1)**, (b) CADMM for SSC** (2.1)**, (c) NADMM for SSC** (2.2)**, (d) ManPL for SSC** (1.2)**, (e) AMA+CADMM for MKSSC** (4.1)**, (f) AMA+NADMM for MKSSC** (4.6)**, and (g) AManPL for MKSSC** (1.3)**.**

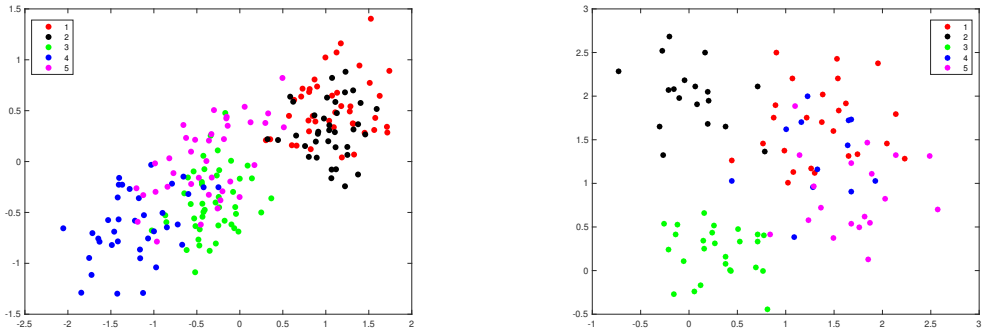
5.2. Synthetic Data

In this subsection, we follow Park and Zhao (2018) to evaluate the clustering performance of different methods on two synthetic datasets with $C = 5$ clusters.

- **Synthetic data 1.** We randomly generate C points in the 2-dimensional latent space spanning a circle as the centers of C clusters. For each cluster, we randomly generate the points by adding an independent noise to its center, and the entries of the noise follow a Gaussian distribution. The noise level is equal to the radius of the circle in the embedded space multiplied by a parameter σ. We project these 2-dimensional data to a p-dimensional space using a linear projection matrix and then add the heterogeneous noise to obtain the data matrix X.
- **Synthetic data 2.** We randomly generate a matrix $B' \in \mathbb{R}^{C\times d}$ with $d = 10$ by drawing its entries independently from Gaussian distributions, where different rows of B' specify heterogeneous variances. We randomly assign the cluster labels $z_1, \ldots, z_n \in [C]$. Let $B = \left[B', 0_{C\times(p-d)}\right]$ and

$Z = (Z_{ij})_{n\times C} = (1_{\{z_i=j\}})_{n\times C}$. We generate $X = ZB + W$, where W is a noise matrix with independent standard normally distributed entries. The noise level is equal to the radius of the circle in the embedded space multiplied by a parameter σ.

Figure 2 visualizes one realization of the simulated data for these two settings. From Figure 2 we see that different clusters mix together and the variability between clusters varies.



(a) Synthetic data 1 (b) Synthetic data 2

Figure 2 Illustration of one realization of the synthetic data.

The parameters λ and ρ are again chosen from (5.1). Note that there are 35 different combinations of (λ, ρ). For each pair of (λ, ρ), we run the three algorithms AMA+CADMM, AMA+NADMM and AManPL for 10 times, and then we report the average NMI scores and average CPU runtime over the 350 independent repetitions. The results are given in Table 3.

From Table 3 we see that AMA+NADMM and AManPL provided better NMI scores than AMA+CADMM in most cases, with the only exception of $(n, p) = (200, 500)$ in synthetic dataset 2 where AMA+CADMM is better in terms of the NMI score. We also observe that AMA+NADMM usually provides slightly better NMI score than AManPL, but AManPL is more efficient in some cases such as $n = 200$, $p = 250$, 300, and 500 in synthetic dataset 2. This suggest that AManPL has potential to be more efficient in large-scale problems.

We further conduct some numerical tests to test the sensitivity of the algorithms to the noise level σ. In particular, we repeat the tests above by varying σ in the following set of values:

$$\sigma \in \{0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5\}. \tag{5.3}$$

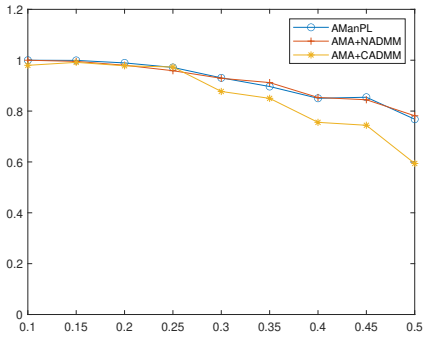
Method		AMA+CADMM		AMA+NADMM		AManPL	
n	p	NMI	CPU	NMI	CPU	NMI	CPU
Synthetic data 1 with $\sigma = 0.3$							
100	250	0.906	0.695	**0.937**	0.825	0.935	1.231
100	300	0.908	0.698	**0.939**	0.829	0.938	1.188
100	500	0.912	0.745	**0.944**	0.870	0.943	1.248
200	250	0.935	1.647	**0.941**	3.608	0.940	3.825
200	300	0.931	1.640	**0.937**	3.574	0.934	3.712
200	500	0.940	2.879	**0.944**	5.799	0.943	6.307
Synthetic data 2 with $\sigma = 0.2$							
100	250	0.727	0.609	**0.986**	1.049	0.973	1.445
100	300	0.742	0.612	**0.980**	1.079	0.968	1.460
100	500	0.906	0.570	**0.976**	1.049	0.961	1.292
200	250	0.959	1.224	**0.990**	4.047	0.984	3.924
200	300	0.972	1.247	**0.989**	4.233	0.980	3.893
200	500	**0.986**	1.265	0.984	4.741	0.970	3.796

Table 3 Comparison of NMI scores and CPU runtime for solving MKSSC for synthetic data 1 and 2.

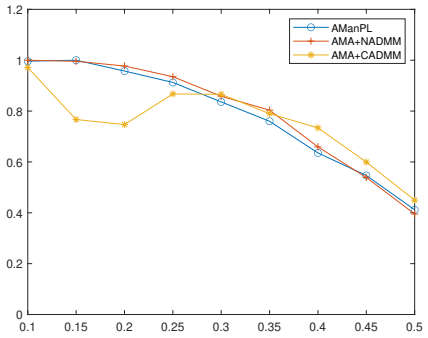
We report the NMI scores of the three algorithms for varying σ in Figure 3 for the two synthetic data sets both with $(n,p) = (100,300)$, $(n,p) = (200,300)$ and $(n,p) = (200,500)$. From Figure 3 we see that AMA+NADMM and AManPL have very similar performance when the noise level σ varies, and they are obviously better than AMA+CADMM for synthetic data 1. However, it appears that when the noise level σ is large, AMA+CADMM is better than the other two for synthetic data 2.

5.3. Single-Cell RNA Sequencing Data Analysis

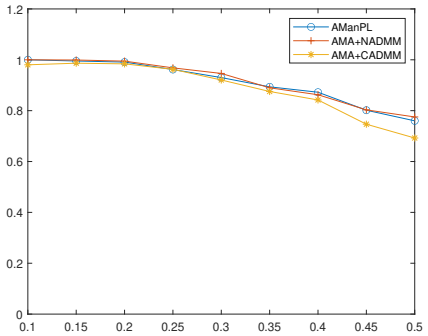
Clustering cells and identifying subgroups are important topics in high-dimensional scRNA-seq data analysis. The multiple kernel learning approach is vital as clustering scRNA-seq data is usually sensitive to the choice of the number of neighbors and scaling parameter. Recently, Park and



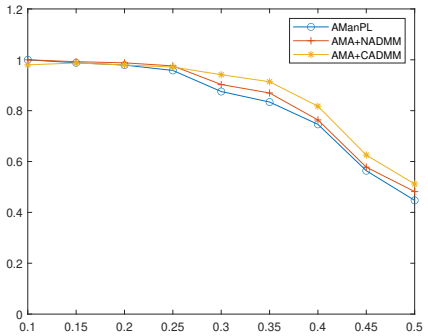
(a) Synthetic data 1, $(n, p) = (100, 300)$



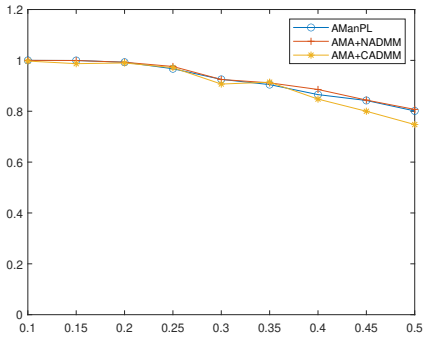
(b) Synthetic data 2, $(n, p) = (100, 300)$



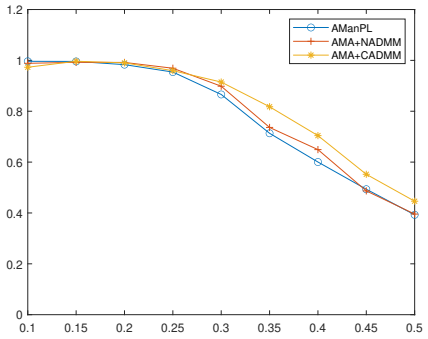
(c) Synthetic data 1, $(n, p) = (200, 300)$



(d) Synthetic data 2, $(n, p) = (200, 300)$



(e) Synthetic data 1, $(n, p) = (200, 500)$



(f) Synthetic data 2, $(n, p) = (200, 500)$

Figure 3 **Plots of NMI scores versus the noise level σ. The x-axis denotes σ and the y-axis denotes the NMI score.**

Zhao (2018) showed that AMA+CADMM for MKSSC provides a promising clustering result and outperforms several state-of-art methods such as SC, SSC, t-SNE (van der Maaten and Hinton 2008), and SIMLR (Wang et al. 2017). In what follows, we focus on the numerical comparison of

AMA+CADMM, AMA+NADMM and AManPL to cluster high-dimensional scRNA-seq data on seven real datasets used in Park and Zhao (2018). These seven real datasets represent several types of important dynamic processes such as cell differentiation, and they include the information about single cell types. We follow the procedure of Park and Zhao (2018) to specify multiple kernels for clustering scRNA-seq data. The parameters λ and ρ are again chosen from (5.1). For each dataset, we report the average NMI score and average CPU runtime of AMA+CADMM, AMA+NADMM and AManPL in Table 4. Note that we can compute the NMI scores because the ground-truth assignments of the clusters of these scRNA-seq data are known in the literature e.g., Park and Zhao (2018). Moreover, since the Tasic dataset is very large, in which $n = 1727$, $p = 5832$ and $C = 49$, we only run the algorithms with one particular choice of $(\lambda, \rho) = (10^{-3}, 10^{-1})$. From Table 4 we see that AManPL provides the best NMI score in most cases, with the first dataset being an exception for which AMA+NADMM gives the best NMI score. Furthermore, for the large-scale Tasic dataset, AManPL provides much better NMI score than the other two algorithms, and it is much more efficient than AMA+NADMM. The results in Table 4 demonstrate that our AManPL for solving (1.3) has great potential in analyzing the scRNA-seq data.

Method	AMA+CADMM		AMA+NADMM		AManPL		
Datasets	NMI	CPU	NMI	CPU	NMI	CPU	C
Deng et al. (2014)	0.679	1.761	**0.732**	3.041	0.707	2.788	7
Ting et al. (2014)	0.901	1.124	0.919	1.576	**0.941**	0.911	5
Treutlein et al. (2014)	0.423	0.572	0.655	0.873	**0.673**	0.672	5
Buettner et al. (2015)	0.505	2.071	0.403	8.879	**0.514**	2.168	3
Schlitzer et al. (2015)	0.368	3.201	0.378	14.769	**0.402**	4.918	3
Pollen et al. (2014)	0.662	4.864	0.816	24.020	**0.821**	6.640	11
Tasic et al. (2016)	0.123	1.083e+02	0.156	3.421e+03	**0.237**	4.692e+02	49

Table 4 Comparison of NMI scores and CPU runtime for solving MKSSC on real scRNA-seq datasets.

5.4. Unsupervised data

In practice, clustering usually needs to be performed for unsupervised data. Here we test the three algorithms AMA+CADMM, AMA+NADMM and AManPL on a microbiome dataset with no ground-truth assignment. The original data is from Morgan et al. (2015). After preprocessing, we have the mRNA expression of 170 genes (i.e., $p = 170$) for all 196 Inflammatory bowel disease (IBD) patients (i.e., $n = 196$). The spectral clustering on such microbiome mRNA data can help us understand the grouping effect of gene expressions for IBD patients. For this dataset, we do not know the ground-truth clusters, therefore NMI cannot be used. Instead, we choose to measure the performance of the clustering by using the Calinski-Harabasz (CH) score (Calinski and Harabasz 1974), which is widely used to measure the cluster quality when there is no ground-truth information available. According to the definition of the CH score, under the same value of proposed clustering number C, higher CH score usually corresponds to better clustering result. We report the CH scores of the three algorithms for clustering number $C = 2, 3, 4$, and 5 in Table 5. From Table 5 we see that AMA+NADMM gives the best CH scores when $C = 2$ and 4, and AManPL gives the best CH scores when $C = 3$ and 5. Moreover, both AMA+NADMM and AManPL produce much better CH scores than AMA+CADMM. We also see that the CH varies significantly for different C, which implies that it is very important to have a good estimation to C in practice.

C	AMA+CADMM	AMA+NADMM	AManPL
2	546.81	**795.18**	775.21
3	223.31	312.62	**316.14**
4	192.10	**193.48**	191.23
5	150.95	178.25	**180.69**

Table 5 Comparison of CH scores for solving MKSSC on an unsupervised dataset.

6. Conclusion

Motivated by the recent demands on analyzing the single cell RNA sequencing data, we considered the sparse spectral clustering and multiple-kernel sparse spectral clustering in this paper. The SSC

and MKSSC can be formulated as optimization problems over the Stiefel manifold with nonsmooth objective function. Existing methods usually solve their convex relaxations or their approximation with the nonsmooth function being approximated by a smooth function. In this paper, we proposed a manifold proximal linear method for solving SSC, and the alternating manifold proximal linear method for solving MKSSC. Convergence and iteration complexity of the proposed methods are analyzed. Numerical results on clustering the single cell RNA sequencing data demonstrated the practical potential of our proposed methods.

Acknowledgement

The authors would like to thank the Editor-in-Chief, the Associate Editor and anonymous referees for insightful and constructive comments that greatly improved the presentation of this paper. Bingyuan Liu and Lingzhou Xue were supported in part by NSF grants CCF-2007823, DMS-1953189, and DMS-1811552. Shixiang Chen was supported in part by NSF grant ECCS-1933878. Shiqian Ma was supported in part by NSF grants DMS-1953210 and CCF-2007797, and UC Davis CeDAR (Center for Data Science and Artificial Intelligence Research) Innovative Data Science Seed Funding Program. Hongyu Zhao was supported in part by NIH grants P50CA1965305 and P30CA016359.

References

Absil PA, Mahony R, Sepulchre R (2008) *Optimization Algorithms on Matrix Manifolds* (Princeton, NJ: Princeton University Press), ISBN 978-0-691-13298-3.

Bendory T, Eldar YC, Boumal N (2018) Non-convex phase retrieval from STFT measurements. *IEEE Transactions on Information Theory* 64(1):467–484.

Boumal N (2016) Nonconvex phase synchronization. *SIAM Journal on Optimization* 26(4):2355–2377.

Boumal N, Absil PA (2011) RTRMC: A Riemannian trust-region method for low-rank matrix completion. *Advances in Neural Information Processing Systems*, 406–414.

Boumal N, Absil PA, Cartis C (2018) Global rates of convergence for nonconvex optimization on manifolds. *IMA Journal of Numerical Analysis* 39(1):1–33.

Buettner F, Natarajan KN, Casale FP, Proserpio V, Scialdone A, Theis FJ, Teichmann SA, Marioni JC, Stegle O (2015) Computational analysis of cell-to-cell heterogeneity in single-cell RNA-sequencing data reveals hidden subpopulations of cells. *Nature Biotechnology* 33(2):155.

Calinski T, Harabasz J (1974) A dendrite method for cluster analysis. *Communications in Statistics* 3(1):1–27.

Charisopoulos V, Chen Y, Davis D, Diaz M, Ding L, Drusvyatskiy D (2019) Low-rank matrix recovery with composite optimization: Good conditioning and rapid convergence. *Foundations of Computational Mathematics (to appear)* .

Charisopoulos V, Davis D, Díaz M, Drusvyatskiy D (2021) Composite optimization for robust blind deconvolution. *Information and Inference: A Journal of the IMA* 10(2):333–396.

Chen S, Deng Z, Ma S, So AMC (2019) Manifold proximal point algorithms for dual principal component pursuit and orthogonal dictionary learning. *Proceedings of the 2019 Asilomar Conference on Signals, Systems, and Computers.*

Chen S, Deng Z, Ma S, So AMC (2020a) Manifold proximal point algorithms for dual principal component pursuit and orthogonal dictionary learning. *arXiv:2005.02356* .

Chen S, Ma S, So AMC, Zhang T (2020b) Proximal gradient method for nonsmooth optimization over the Stiefel manifold. *SIAM J. Optimization* 30(1):210–239.

Chen S, Ma S, Xue L, Zou H (2020c) An alternating manifold proximal gradient method for sparse principal component analysis and sparse canonical correlation analysis. *INFORMS Journal on Optimization* 2(3):192–208.

Cherian A, Sra S (2017) Riemannian dictionary learning and sparse coding for positive definite matrices. *IEEE Transactions on Neural Networks and Learning Systems* 28(12):2859–2871.

Chung FR, Graham FC (1997) *Spectral Graph Theory* (American Mathematical Society).

Deng Q, Ramsköld D, Reinius B, Sandberg R (2014) Single-cell RNA-seq reveals dynamic, random monoallelic gene expression in mammalian cells. *Science* 343(6167):193–196.

Drusvyatskiy D, Paquette C (2019) Efficiency of minimizing compositions of convex functions and smooth maps. *Mathematical Programming Series A* 178(1-2):503–558.

Dua D, Graff C (2017) UCI machine learning repository. URL `http://archive.ics.uci.edu/ml`.

Duchi JC, Ruan F (2018) Stochastic methods for composite and weakly convex optimization problems. *SIAM Journal on Optimization* 28(4):3229–3259.

Duchi JC, Ruan F (2019) Solving (most) of a set of quadratic equalities: Composite optimization for robust phase retrieval. *Information and Inference* 8:471–529.

Ferreira OP, Oliveira PR (1998) Subgradient algorithm on Riemannian manifolds. *Journal of Optimization Theory and Applications* 97(1):93–104.

Friedman J, Hastie T, Tibshirani R (2001) *The Elements of Statistical Learning* (Springer Series in Statistics New York).

Grohs P, Hosseini S (2016) ε-subgradient algorithms for locally lipschitz functions on Riemannian manifolds. *Advances in Computational Mathematics* 42(2):333–360.

Hosseini S, Pouryayevali M (2011) Generalized gradients and characterization of epi-lipschitz sets in Riemannian manifolds. *Nonlinear Analysis: Theory, Methods & Applications* 74(12):3884–3895.

Hosseini S, Uschmajew A (2017) A Riemannian gradient sampling algorithm for nonsmooth optimization on manifolds. *SIAM Journal on Optimization* 27(1):173–189.

Huang W, Wei K (2019) Extending FISTA to Riemannian optimization for sparse PCA. *arXiv preprint arXiv:1909.05485* .

Huang W, Wei K (2021) Riemannian proximal gradient methods. *Mathematical Programming, in press* .

Kiselev VY, Andrews TS, Hemberg M (2019) Challenges in unsupervised clustering of single-cell RNA-seq data. *Nature Reviews Genetics* 20:273–282.

Lewis AS, Wright SJ (2016) A proximal method for composite minimization. *Mathematical Programming* 158(1-2):501–546.

Li X, Chen S, Deng Z, Qu Q, Zhu Z, So AMC (2021) Weakly convex optimization over Stiefel manifold using Riemannian subgradient-type methods. *Accepted for publication in SIAM Journal on Optimization* .

Li X, Sun D, Toh KC (2018) A highly efficient semismooth Newton augmented Lagrangian method for solving Lasso problems. *SIAM J. Optimization* 28(1):433–458.

Liu H, Yue MC, So AMC (2017) On the estimation performance and convergence rate of the generalized power method for phase synchronization. *SIAM Journal on Optimization* 27(4):2426–2446.

Lu C, Feng J, Lin Z, Yan S (2018) Nonconvex sparse spectral clustering by alternating direction method of multipliers and its convergence analysis. *Thirty-Second AAAI Conference on Artificial Intelligence*.

Lu C, Yan S, Lin Z (2016) Convex sparse spectral clustering: Single-view to multi-view. *IEEE Transactions on Image Processing* 25(6):2833–2843.

Mifflin R (1977) Semismooth and semiconvex functions in constrained optimization. *SIAM Journal on Control and Optimization* 15(6):959–972.

Morgan XC, Kabakchiev B, Waldron L, Tyler AD, Tickle TL, Milgrom R, Stempak JM, Gevers D, Xavier RJ, Silverberg MS, Huttenhower C (2015) Associations between host gene expression, the mucosal microbiome, and clinical outcome in the pelvic pouch of patients with inflammatory bowel disease. *Genome Biology* 16:Article number: 67.

Ng AY, Jordan MI, Weiss Y (2002) On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems*, 849–856.

Park S, Zhao H (2018) Spectral clustering based on learning similarity matrix. *Bioinformatics* 34(12):2069–2076.

Pollen AA, Nowakowski TJ, Shuga J, Wang X, Leyrat AA, Lui JH, Li N, Szpankowski L, Fowler B, Chen P, et al. (2014) Low-coverage single-cell mRNA sequencing reveals cellular heterogeneity and activated signaling pathways in developing cerebral cortex. *Nature Biotechnology* 32(10):1053.

Qi L, Sun J (1993) A nonsmooth version of Newton's method. *Mathematical Programming* 58(1-3):353–367.

Schlitzer A, Sivakamasundari V, Chen J, Sumatoh HRB, Schreuder J, Lum J, Malleret B, Zhang S, Larbi A, Zolezzi F, et al. (2015) Identification of cDC1-and cDC2-committed DC progenitors reveals early lineage priming at the common DC progenitor stage in the bone marrow. *Nature Immunology* 16(7):718.

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

Strehl A, Ghosh J (2003) Cluster ensembles – a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research* 3:583–617.

Sun J, Qu Q, Wright J (2017) Complete dictionary recovery over the sphere I: Overview and the geometric picture. *IEEE Transactions on Information Theory* 63(2):853–884.

Sun J, Qu Q, Wright J (2018) A geometrical analysis of phase retrieval. *Foundations of Computational Mathematics* 18(5):1131–1198.

Tasic B, et al. (2016) Adult mouse cortical cell taxonomy revealed by single cell transcriptomics. *Nat. Neurosci.* 19:335–346.

Ting DT, Wittner BS, Ligorio M, Jordan NV, Shah AM, Miyamoto DT, Aceto N, Bersani F, Brannigan BW, Xega K, et al. (2014) Single-cell RNA sequencing identifies extracellular matrix gene expression by pancreatic circulating tumor cells. *Cell Reports* 8(6):1905–1918.

Treutlein B, Brownfield DG, Wu AR, Neff NF, Mantalas GL, Espinoza FH, Desai TJ, Krasnow MA, Quake SR (2014) Reconstructing lineage hierarchies of the distal lung epithelium using single-cell RNA-seq. *Nature* 509(7500):371.

van der Maaten L, Hinton G (2008) Visualizing data using t-SNE. *Journal of Machine Learning Research* 9(Nov):2579–2605.

Wang B, Ma S, Xue L (2020) Riemannian stochastic proximal gradient methods for nonsmooth optimization over the Stiefel manifold. *https://arxiv.org/pdf/2005.01209.pdf* .

Wang B, Ramazzotti D, De Sano L, Zhu J, Pierson E, Batzoglou S (2017) SIMLR: a tool for large-scale single-cell analysis by multi-kernel learning. *PROTEOMICS* 18:Article number: 1700232.

Wang C, Sun D, Toh KC (2010) Solving log-determinant optimization problems by a Newton-CG primal proximal point algorithm. *SIAM Journal on Optimization* 20(6):2994–3013.

Xiao X, Li Y, Wen Z, Zhang L (2018) A regularized semi-smooth Newton method with projection steps for composite convex programs. *Journal of Scientific Computing* 76:364–389.

Yang J, Sun D, Toh KC (2013) A proximal point algorithm for log-determinant optimization with group Lasso regularization. *SIAM Journal on Optimization* 23(2):857–893.

Yang WH, Zhang LH, Song R (2014) Optimality conditions for the nonlinear programming problems on Riemannian manifolds. *Pacific Journal of Optimization* 10(2):415–434.

Zhao XY, Sun D, Toh KC (2010) A Newton-CG augmented Lagrangian method for semidefinite programming. *SIAM Journal on Optimization* 20(4):1737–1765.