Journal of Machine Learning Research 22 (2021) 1-32 Submitted 10/18; Revised 11/20; Published 1/21

Convex Clustering: Model, Theoretical Guarantee and Efficient Algorithm

Defeng Sun DEFENG.SUN@POLYU.EDU.HK
Department of Applied Mathematics
The Hong Kong Polytechnic University
Hong Kong

Kim-Chuan Toh MATTOHKC@NUS.EDU.SG
Department of Mathematics and Institute of Operations Research and Analytics
National University of Singapore
10 Lower Kent Ridge Road, Singapore 119076

Yancheng Yuan* YANCHENG.YUAN@POLYU.EDU.HK
Department of Applied Mathematics
The Hong Kong Polytechnic University
Hong Kong

Editor: Inderjit Dhillon

Abstract

Clustering is a fundamental problem in unsupervised learning. Popular methods like K-means, may suffer from poor performance as they are prone to get stuck in its local minima. Recently, the sum-of-norms (SON) model (also known as the convex clustering model) has been proposed by Pelckmans et al. (2005), Lindsten et al. (2011) and Hocking et al. (2011). The perfect recovery properties of the convex clustering model with uniformly weighted all-pairwise-differences regularization have been proved by Zhu et al. (2014) and Panahi et al. (2017). However, no theoretical guarantee has been established for the general weighted convex clustering model, where better empirical results have been observed. In the numerical optimization aspect, although algorithms like the alternating direction method of multipliers (ADMM) and the alternating minimization algorithm (AMA) have been proposed to solve the convex clustering model (Chi and Lange, 2015), it still remains very challenging to solve large-scale problems. In this paper, we establish sufficient conditions for the perfect recovery guarantee of the general weighted convex clustering model, which include and improve existing theoretical results in (Zhu et al., 2014; Panahi et al., 2017) as special cases. In addition, we develop a semismooth Newton based augmented Lagrangian method for solving large-scale convex clustering problems. Extensive numerical experiments on both simulated and real data demonstrate that our algorithm is highly efficient and robust for solving large-scale problems. Moreover, the numerical results also show the superior performance and scalability of our algorithm comparing to the existing first-order methods. In particular, our algorithm is able to solve a convex clustering problem with 200,000 points in $\mathbb{R}^3$ in about 6 minutes.

Keywords: convex clustering, augmented Lagrangian method, semismooth Newton method, conjugate gradient method, unsupervised learning.

*. Corresponding author.

1. Introduction

Clustering is one of the most fundamental problems in unsupervised learning. Traditional clustering models, such as K-means and hierarchical clustering, may suffer from poor performance because of the non-convexity of the models and the difficulties in finding global optimal solutions for such models. The clustering results are generally highly dependent on the initializations and the results could differ significantly with different initializations. Moreover, these clustering models require the prior knowledge about the number of clusters which is not available in many real applications. Therefore, in practice, K-means is typically tried with different cluster numbers and the user will then decide on a suitable value based on his judgment on which computed result agrees best with his domain knowledge. Obviously, such a process could make the clustering results subjective.

In order to overcome the above issues, a new clustering model has been proposed (Pelckmans et al., 2005; Lindsten et al., 2011; Hocking et al., 2011) and demonstrated to be more robust compared to those traditional ones. Let $A \in \mathbb{R}^{d\times n} = [\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_n]$ be a given data matrix with n observations and d features. The convex clustering model for these n observations solves the following convex optimization problem:

$$\min_{X\in\mathbb{R}^{d\times n}} \frac{1}{2}\sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{a}_i\|^2 + \gamma \sum_{i<j} \|\mathbf{x}_i - \mathbf{x}_j\|_p, \tag{1}$$

where $\gamma > 0$ is a tuning parameter, and $\|\cdot\|_p$ denotes the vector p-norm. Here and below, $\|\cdot\|$ is used to denote the vector 2-norm or the Frobenius norm of a matrix. The p-norm above with $p \geq 1$ ensures the convexity of the model. Typically, p is chosen to be $1, 2$, or ∞. After solving (1) and obtaining the optimal solution $X^* = [\mathbf{x}_1^*, \ldots, \mathbf{x}_n^*]$, we assign $\mathbf{a}_i$ and $\mathbf{a}_j$ to the same cluster if and only if $\mathbf{x}_i^* = \mathbf{x}_j^*$. In other words, $\mathbf{x}_i^*$ is the centroid for observation $\mathbf{a}_i$. (Here we used the word "centroid" to mean the approximate one associated with $\mathbf{a}_i$ but not the final centroid of the cluster to which $\mathbf{a}_i$ belongs to. In practice, instead of requiring $\mathbf{x}_i^* = \mathbf{x}_j^*$, we assign $\mathbf{a}_i$ and $\mathbf{a}_j$ to the same cluster if $\|\mathbf{x}_i^* - \mathbf{x}_j^*\| \leq \epsilon$ for a given tolerance $\epsilon > 0$. In our numerical experiments, we usually set $\epsilon = 10^{-5}$. This has been also discussed in a recent preprint (Jiang and Vavasis, 2020).) The key idea behind the convex clustering model is that, if two observations $\mathbf{a}_i$ and $\mathbf{a}_j$ belong to the same cluster, then their corresponding centroids $\mathbf{x}_i^*$ and $\mathbf{x}_j^*$ should be the same. The first term in the model (1) is the fidelity term while the second term is the regularization term to penalize the differences between different centroids so as to enforce the property that centroids for observations in the same cluster should be identical.

The advantages of convex clustering lie mainly in two aspects. First, since the clustering model (1) is strongly convex, the optimal solution of the model for a given positive γ is unique and is more easily obtainable than traditional clustering algorithms like K-means. Second, instead of requiring the prior knowledge of the number of clusters, we can generate a clustering path via solving (1) for a sequence of positive values of γ. To handle cluster recovery for large-scale data sets, various researchers, e.g., Pelckmans et al. (2005); Lindsten et al. (2011); Hocking et al. (2011); Zhu et al. (2014); Tan and Witten (2015); Panahi et al.

(2017) have suggested the following weighted convex clustering model modified from (1):

$$\min_{X\in\mathbb{R}^{d\times n}} \frac{1}{2}\sum_{i=1}^{n}\|\mathbf{x}_i-\mathbf{a}_i\|^2+\gamma\sum_{i<j} w_{ij}\|\mathbf{x}_i-\mathbf{x}_j\|_p, \tag{2}$$

where $w_{ij}=w_{ji}\geq 0$ are given weights that are chosen based on the input data A. One can regard the original convex clustering model (1) as a special case, if we take $w_{ij}=1$ for all $i<j$ in the above weighted convex clustering model. To make the computational cost cheaper when evaluating the regularization terms, one would generally put a non-zero weight only for a pair of points which are nearby each other, and a typical choice of the weights is

$$w_{ij}=\begin{cases}\exp(-\phi\|\mathbf{a}_i-\mathbf{a}_j\|^2) & \text{if } (i,j)\in\mathcal{E},\\ 0 & \text{otherwise,}\end{cases}$$

where $\mathcal{E}=\cup_{i=1}^{n}\{(i,j)\mid \mathbf{a}_j \text{ is among } \mathbf{a}_i\text{'s } k\text{-nearest neighbors}, i<j\leq n\}$, and ϕ is a given positive constant.

The aforementioned advantages and the success of the convex clustering model (1) in recovering clusters in many examples with well selected values of γ have motivated researchers to provide theoretical guarantees on the cluster recovery property of (1). The first theoretical result on cluster recovery, which is established in (Zhu et al., 2014), is only valid for the case of two clusters. It showed that the model (1) can recover the two clusters perfectly if the data points are drawn from two cubes that are well separated. Tan and Witten (2015) further analyzed the statistical properties of (1). Recently, Panahi et al. (2017) provided theoretical recovery results in the general case of k clusters under relatively mild sufficient conditions, for the fully uniformly weighted convex clustering model (1).

In the practical aspect, various researchers have observed that better empirical performance can be achieved by (2) with well chosen weights when comparing to the original model (1) (Hocking et al., 2011; Lindsten et al., 2011; Chi and Lange, 2015). However, to the best of our knowledge, no theoretical recovery guarantee has been established for the general weighted convex clustering model (2). In this paper, we will propose mild sufficient conditions for (2) to attain perfect cluster recovery, which also include and improve the theoretical results in (Zhu et al., 2014; Panahi et al., 2017) as special cases. Our theoretical results thus definitively strengthened the theoretical foundation of convex clustering model. As expected, the conditions provided in the theoretical analysis are usually not checkable before one finds the right clusters and thus the range of parameter values for γ to achieve perfect recovery is an unknown priori. In practice, this difficulty is mitigated by choosing a sequence of values of γ to generate a clustering path.

The challenges for the convex model to obtain meaningful cluster recovery is then to solve it efficiently for a range of values of γ. Lindsten et al. (2011) used the off-the-shelf solver, CVX, to generate the solution path. However, Hocking et al. (2011) realized that CVX is competitive only for small-scale problems and it does not scale well when the number of data points increases. Thus the paper introduced three algorithms based on the subgradient methods for different regularizers corresponding to $p=1,2,\infty$. Recently, some new algorithms have been proposed to solve this problem. In particular, Chi and Lange (2015) adapted the ADMM and AMA to solve (1). However, as we will see in our numerical experiments, both algorithms may still encounter scalability issues, albeit less severe than

CVX. Furthermore, the efficiency of these two algorithms is sensitive to the parameter γ. This is not a favorable property since we need to solve (1) with γ in a relative large range to generate the clustering path. In (Panahi et al., 2017), the authors proposed a stochastic splitting algorithm for solving (1) in an attempt to resolve the aforementioned scalability issues. Although this stochastic approach scales well with the problem size (n in (1)), the convergence rate shown in (Panahi et al., 2017) is rather weak in that it requires at least $l \geq n^4/\varepsilon$ iterations to generate a solution X^l such that $\|X^l - X^*\|^2 \leq \varepsilon$ is satisfied with high probability. Moreover, because the error estimate is given in the sense of high probability, it is difficult to design an appropriate stopping condition for the algorithm in practice.

As the readers may observe, all the existing algorithms are purely first-order methods that do not use any second-order information underlying the convex clustering model. In contrast, here we design and analyse a deterministic second-order algorithm, the semismooth Newton based augmented Lagrangian method, to solve the convex clustering model. Our algorithm is motivated by the recent work (Li et al., 2018) in which the authors have proposed a semismooth Newton augmented Lagrangian method (ALM) to solve Lasso and fused Lasso problems, and the algorithm is demonstrated to be highly efficient for solving large, or even huge scale problems accurately. We are thus inspired to adapt this ALM framework for solving the weighted convex clustering model (2) in this paper.

Next we present a short summary of our main contributions in this paper.

1. We prove the perfect recovery guarantee of the general weighted convex clustering model (2) under mild sufficient conditions. Our results are not only applicable to the more practical weighted convex model but also improve the existing results when specialized to the fully uniformly weighted model (1). Moreover, our bounds for the tuning parameter γ are given explicitly in terms of the data points and their corresponding pairwise weights in the regularization term.

2. We propose a highly efficient and scalable algorithm, called the semismooth Newton based augmented Lagrangian method, to solve the convex clustering model, which is not only proven to be theoretically efficient but it is also demonstrated to be practically highly efficient and robust.

The remaining parts of this paper are organized as follows. We will summarize some related work in section 2. In section 3, we will introduce some preliminaries and notations which will be used in this paper. Theoretical results on the perfect recovery properties of the weighted convex clustering model will be presented in section 4. In section 5, we will introduce a highly efficient and robust optimization algorithm for solving the convex clustering model. After that, we will conduct numerical experiments to verify the theoretical results and evaluate the performance of our algorithm in section 6. Finally, we conclude the paper in section 7.

2. Related Work Based on Semidefinite Programming

In addition to the papers (Pelckmans et al., 2005; Lindsten et al., 2011; Hocking et al., 2011; Zhu et al., 2014; Tan and Witten, 2015; Panahi et al., 2017; Chi et al., 2018) on the convex clustering models (1) and (2), other convex models have been proposed to deal with the

non-convexity of the K-means clustering model. One such model is the convex relaxation of the K-means model via semidefinite programming (SDP) (Peng and Wei, 2007; Awasthi et al., 2015; Mixon et al., 2017).

For a given data matrix $A \in \mathbb{R}^{d\times n} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n]$, the classical K-means model solves the following non-convex optimization problem

$$\begin{array}{ll} \min & \sum_{t=1}^{k} \sum_{i\in I_t} \|\mathbf{a}_i - \frac{1}{|I_t|}\sum_{j\in I_t} \mathbf{a}_j\|^2 \\ \text{s.t.} & I_1, \ldots, I_k \text{ is a partition of } \{1, 2, \ldots, n\}. \end{array} \tag{3}$$

Now, if we define the $n \times n$ matrix D by $D_{ij} = \|\mathbf{a}_i - \mathbf{a}_j\|^2$, then by taking

$$X := \sum_{t=1}^{k} \frac{1}{|I_t|} \mathbf{1}_{I_t} \mathbf{1}_{I_t}^T,$$

where $\mathbf{1}_{I_t} \in \mathbb{R}^n$ is the indicator vector of the index set I_t, we can express the objective function in (3) as $\frac{1}{2}\mathrm{Tr}(DX)$. Based on this, Peng and Wei (2007) proposed the following SDP relaxation of the K-means model:

$$\min \Big\{ \mathrm{Tr}(DX) \mid \mathrm{Tr}(X) = k,\ X\mathbf{e} = \mathbf{e},\ X \geq 0,\ X \in \mathbb{S}_n^+ \Big\}, \tag{4}$$

where $X \geq 0$ means that all the elements in X are nonnegative, $\mathbb{S}_+^n$ is the cone of $n \times n$ symmetric and positive semidefinite matrices, and $\mathbf{e} \in \mathbb{R}^n$ is the column vector of all ones.

Recently, Mixon et al. (2017) proved that the K-means SDP relaxation approach can achieve perfect cluster recovery with high probability when the data A is sampled from the stochastic unit-ball model in $\mathbb{R}^d$, provided that the cluster centriods $\{\mathbf{a}^{(1)}, \ldots, \mathbf{a}^{(k)}\}$ satisfy the condition that $\min\{\|\mathbf{a}^{(\alpha)} - \mathbf{a}^{(\beta)}\| \mid 1 \leq \alpha < \beta \leq k\} > 2\sqrt{2}(1 + 1/\sqrt{d})$. However, the computational efficiency of SDP based relaxations highly depends on the efficiency of the available SDP solvers. While recent progress (Zhao et al., 2010; Yang et al., 2015; Sun et al., 2020) in solving large-scale SDPs allows one to solve the SDP relaxation problem for clustering 2–3 thousand points, it is however prohibitively expensive to solve the problem when n goes beyond 3000.

The work in (Chi and Lange, 2015) has implicitly demonstrated that it is generally much cheaper to solve the model (2) instead of the SDP relaxation model. However, based on our numerical experiments, the algorithms ADMM and AMA proposed in (Chi and Lange, 2015) for solving (2) only work efficiently when the number of data points is not too large (several thousands depending on the feature dimension of the data). Also, it is not easy for the proposed algorithms in (Chi and Lange, 2015) to achieve relatively high accuracy. This also explains why we need to design a new algorithm in this paper to overcome the aforementioned difficulties.

3. Preliminaries and Notation

In this section, we first introduce some preliminaries and notation which will be used later in this paper. For theoretical analysis, we adopt some definitions and notation from (Zhu et al., 2014; Panahi et al., 2017).

Definition 1 *For a given finite set $A = \{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n\} \subset \mathbb{R}^d$ and its partitioning $\mathcal{V} = \{V_1, V_2, \ldots, V_K\}$, where each V_i is a subset of A.*

(a) We say that a map ψ on A perfectly recovers $\mathcal{V}$ when $\psi(\mathbf{a}_i) = \psi(\mathbf{a}_j)$ is equivalent to $\mathbf{a}_i$ and $\mathbf{a}_j$ belonging to the same cluster. In other words, there exist distinct vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_K$ such that $\psi(\mathbf{a}_i) = \mathbf{v}_\alpha$ holds whenever $\mathbf{a}_i \in V_\alpha$.

(b) We call a partitioning $\mathcal{W} = \{W_1, W_2, \ldots, W_L\}$ of A a coarsening of $\mathcal{V}$ if each partition W_l is obtained by taking the union of a number of partitions in $\mathcal{V}$. Furthermore, $\mathcal{W}$ is called the trivial coarsening of $\mathcal{V}$ if $\mathcal{W} = \{A\}$. Otherwise, it is called a non-trivial coarsening.

Definition 2 *For any finite set $S \subset \mathbb{R}^d$, its diameter with respect to the q-norm for $q \geq 1$ is defined as*

$$D_q(S) := \max\{\|\mathbf{x} - \mathbf{y}\|_q \mid \mathbf{x}, \mathbf{y} \in S\}.$$

Moreover, we define its separation and centroid, respectively, as

$$d_q(S) := \min\{\|\mathbf{x} - \mathbf{y}\|_q \mid \mathbf{x}, \mathbf{y} \in S, \mathbf{x} \neq \mathbf{y}\}, \qquad c(S) = \frac{\sum_{\mathbf{x} \in S} \mathbf{x}}{|S|}.$$

For convenience, for any family of mutually disjoint finite sets $\mathcal{F} = \{F_i \subset \mathbb{R}^d\}$, we define $\mathcal{C}(\mathcal{F}) = \{c(F_i)\}$.

Later in this paper, we will establish the theoretical recovery guarantee based on the above definitions. Next, we introduce some preliminaries and notations for the design and analysis of the numerical optimization algorithms.

For a given simple undirected graph $\mathcal{G} = (\{1, \ldots, n\}, \mathcal{E})$ with n vertices and edges defined in $\mathcal{E}$, we define the symmetric adjacency matrix $G \in \mathbb{R}^{n \times n}$ with entries

$$G_{ji} = G_{ij} = \begin{cases} 1 & if\ (i,j) \in \mathcal{E}, \\ 0 & \text{otherwise.} \end{cases}$$

Based on an enumeration of the index pairs in $\mathcal{E}$ (say in the lexicographic order), which we denote by $l(i,j)$ for the pair (i,j), we define the node-arc incidence matrix $\mathcal{J} \in \mathbb{R}^{n \times |\mathcal{E}|}$ as

$$\mathcal{J}_k^{l(i,j)} = \begin{cases} 1 & \text{if } k = i, \\ -1 & \text{if } k = j, \\ 0 & \text{otherwise,} \end{cases} \tag{5}$$

where $\mathcal{J}_k^{l(i,j)}$ is the k-th entry of the $l(i,j)$-th column of $\mathcal{J}_k$.

Proposition 3 *With the matrices G and $\mathcal{J}$ defined above, we have the following results*

$$\mathcal{J}\mathcal{J}^T = \mathrm{diag}(G\mathbf{e}) - G =: L_G, \tag{6}$$

where $\mathbf{e} \in \mathbb{R}^n$ is the column vector of all ones, and L_G is the Laplacian matrix associated with the adjacency matrix G.

Now, for given variables $X \in \mathbb{R}^{d\times n}$, $Z \in \mathbb{R}^{d\times|\mathcal{E}|}$ and the graph $\mathcal{G}$, we define the linear map $\mathcal{B} : \mathbb{R}^{d\times n} \to \mathbb{R}^{d\times|\mathcal{E}|}$ and its adjoint $\mathcal{B}^* : \mathbb{R}^{d\times|\mathcal{E}|} \to \mathbb{R}^{d\times n}$, respectively, by

$$\mathcal{B}(X) = [(\mathbf{x}_i - \mathbf{x}_j)]_{(i,j)\in\mathcal{E}} = X\mathcal{J}, \tag{7}$$

$$\mathcal{B}^*(Z) = Z\mathcal{J}^T. \tag{8}$$

Thus, by Proposition 3, we have

$$\mathcal{B}^*(\mathcal{B}(X)) = X\mathcal{J}\mathcal{J}^T = XL_G. \tag{9}$$

Let $p : \mathcal{X} \to (-\infty, +\infty]$ be a given proper closed convex function. The proximal mapping $\mathrm{Prox}_{tp}(\cdot)$ is defined by

$$\mathrm{Prox}_{tp}(x) = \arg\min_{u\in\mathcal{X}}\{tp(u) + \frac{1}{2}\|u - x\|^2\}, \quad x \in \mathcal{X}, \tag{10}$$

where $t > 0$ is a constant. In this paper, we will often make use of the following Moreau identity (See (Bauschke and Combettes, 2011)[Theorem 14.3(ii)])

$$\mathrm{Prox}_{tp}(x) + t\mathrm{Prox}_{p^*/t}(x/t) = x,$$

where p^* is the conjugate function of p. In particular, if $p(x) = \rho\|x\|_2$, then it is not difficult to show that, $p^*(y)$ is the indicator function defined as follows:

$$p^*(y) = \begin{cases} 0 & \text{if } \|y\|_2 \le \rho, \\ +\infty & \text{if } \|y\|_2 > \rho \end{cases}$$

and the proximal mapping $\mathrm{Prox}_{p^*}(\cdot)$ is given by

$$\mathrm{Prox}_{p^*}(y) = \begin{cases} y & \text{if } \|y\|_2 \le \rho, \\ \rho y/\|y\|_2 & \text{if } \|y\|_2 > \rho. \end{cases}$$

It is well known that proximal mappings are important for designing optimization algorithms and they have been well studied. The proximal mappings for many commonly used functions have closed form formulas. Here, we summarize those that are related to this paper in Table 1. In the table, $\delta_C(\cdot)$ denotes the indicator function of a given closed convex set C, which is defined as

$$\delta_C(x) = \begin{cases} 0 & \text{if } x \in C, \\ +\infty & \text{if } x \notin C. \end{cases}$$

We denote the projection onto C as Π_C.

4. Theoretical Guarantee of Convex Clustering Models

The empirical success of the convex clustering model (1) has strongly motivated researchers to investigate its theoretical recovery guarantee. The perfect recovery guarantee for convex clustering model (1), where all pairwise differences are considered with equal weights, have been proved by Zhu et al. (2014) for the 2-clusters case and later by Panahi et al. (2017) for

Table 1: Proximal maps for selected functions

$p(\cdot)$	$\mathrm{Prox}_{tp}(\mathbf{x})$	Comment
$\|\cdot\|_1$	$\left[1-\frac{t}{\|\mathbf{x}_l\|}\right]_+\mathbf{x}_l$	Elementwise soft-thresholding
$\|\cdot\|_2$	$\left[1-\frac{t}{\|\mathbf{x}\|_2}\right]_+\mathbf{x}$	Blockwise soft-thresholding
$\|\cdot\|_\infty$	$\mathbf{x}-\Pi_{t\mathcal{S}}(\mathbf{x})$	$\mathcal{S}$ is the unit ℓ_1-ball
$\delta_C(\cdot)$	$\Pi_C(\mathbf{x})$	C is a closed convex set

the k-clusters case. Tan and Witten (2015) analyzed the statistical properties of model (1) and Radchenko and Mukherjee (2017) analyzed the statistical properties of model (1) with the ℓ_1-regularization terms. In practice, many researchers (e.g. Tan and Witten (2015); Chi and Lange (2015)) have suggested the use of the model (2), which is not only to be more computationally attractive but also to obtain more robust clustering results. However, so far no theoretical guarantee has been provided for the convex clustering model with general weights. In this section, we first review the nice theoretical results proved by Zhu et al. (2014) and Panahi et al. (2017) for (1), and then we will present our new theoretical guarantee for the more challenging case of the general weighted convex clustering model (2).

4.1 Theoretical Recovery Guarantee of Convex Clustering Model (1)

The first theoretical result by Zhu et al. (2014) guarantees the perfect recovery of (1) for the two-clusters case when the data in each cluster are contained in a cube and the two cubes are sufficiently well separated. More recently, much stronger theoretical results have been established by Panahi et al. (2017) wherein the authors proved the theoretical recovery guarantee of the fully uniformly weighted model (1) for the general case of k-clusters.

Theorem 4 (Panahi et al. (2017)) *Consider a finite set $A=\{\mathbf{a}_i\in\mathbb{R}^d \mid i=1,2,\ldots,n\}$ of vectors and its partitioning $\mathcal{V}=\{V_1,V_2,\ldots,V_K\}$. For the SON model in (1), denote its optimal solution by $\{\bar{\mathbf{x}}_i\}$ and define the map $\phi(\mathbf{a}_i)=\bar{\mathbf{x}}_i$, $i=1,\ldots,n$.*

(i) If γ is chosen such that

$$\max_{V\in\mathcal{V}}\frac{D_2(V)}{|V|} \;\le\; \gamma \le \frac{d_2(\mathcal{C}(\mathcal{V}))}{2n\sqrt{K}},$$

then the map ϕ perfectly recovers $\mathcal{V}$.

(ii) If γ satisfies the following inequalities,

$$\max_{V\in\mathcal{V}}\frac{D_2(V)}{|V|} \le \gamma \le \max_{V\in\mathcal{V}}\frac{\|c(A)-c(V)\|_2}{|A|-|V|},$$

then the map ϕ perfectly recovers a non-trivial coarsening of $\mathcal{V}$.

It was shown in (Panahi et al., 2017) that one can treat the theoretical result in (Zhu et al., 2014) as a special case of Theorem 4.

We shall see in the next subsection that we can improve the upper bound in part (i) of Theorem 4 to $\gamma\le\frac{d_2(\mathcal{C}(\mathcal{V}))}{2n}$, as a special case of our new theoretical results.

4.2 Theoretical Recovery Guarantee of the Weighted Convex Clustering Model (2)

Although the convex clustering model (1) with the fully uniformly weighted regularization has the nice theoretical recovery guarantee, it is usually computationally too expensive to solve since the number of terms in the regularization grows quadratically with the number of data points. In order to reduce the computational burden, in practice many researchers have proposed to use the partially weighted convex clustering model (2). Moreover, they have observed better empirical performance of (2) with well chosen weights, comparing to the original model (1) (Hocking et al., 2011; Lindsten et al., 2011; Chi and Lange, 2015). However, to the best of our knowledge, so far no theoretical recovery results have been established for the general weighted convex clustering model (2). Here we will prove that under rather mild conditions, perfect recovery can be guaranteed for the weighted model (2). In addition, our theoretical results subsume the known results for the fully uniformly weighted model (1) as special cases.

Next, we will establish the main theoretical results for (2). Our results and part of the proof have been inspired by the ideas used in (Panahi et al., 2017). For convenience, we define the index sets

$$I_\alpha := \{i \mid \mathbf{a}_i \in V_\alpha\}, \ \text{ for } \alpha = 1, 2, \ldots, K.$$

Let $n_\alpha = |I_\alpha|$,

$$\mathbf{a}^{(\alpha)} = \frac{1}{n_\alpha} \sum_{i \in I_\alpha} \mathbf{a}_i, \quad w^{(\alpha,\beta)} = \sum_{i \in I_\alpha} \sum_{j \in I_\beta} w_{ij}, \quad \forall\, \alpha, \beta = 1, \ldots, K$$

$$w_i^{(\beta)} = \sum_{j \in I_\beta} w_{ij}, \quad \forall\, i = 1, \ldots, n, \ \beta = 1, \ldots, K.$$

Here we will interpret $w_i^{(\beta)}$ as the coupling between point $\mathbf{a}_i$ and the β-th cluster, and $w^{(\alpha,\beta)}$ as the coupling between the α-th cluster and the β-th cluster. For $p \geq 1$, we define

$$h(\mathbf{v}) := \|\mathbf{v}\|_p = \Big(\sum_{i=1}^{d} |v_i|^p \Big)^{\frac{1}{p}}, \quad \mathbf{v} = (v_1, v_2, \ldots, v_d) \in \mathbb{R}^d,$$

and note that the subdifferential of $h(\mathbf{v})$ is given by

$$\partial h(\mathbf{v}) = \begin{cases} \{\mathbf{y} \in \mathbb{R}^d \mid \|\mathbf{y}\|_q \leq 1, \langle \mathbf{y}, \mathbf{v} \rangle = \|\mathbf{v}\|_p\} & \text{if } \mathbf{v} \neq 0, \\ \{\mathbf{y} \in \mathbb{R}^d \mid \|\mathbf{y}\|_q \leq 1\} & \text{if } \mathbf{v} = 0, \end{cases}$$

where $q \geq 1$ is the conjugate index of p such that $\frac{1}{p} + \frac{1}{q} = 1$. Observe that, for any $\mathbf{y} \in \partial h(\mathbf{v})$, we have $\|\mathbf{y}\|_q \leq 1$.

Theorem 5 *Consider the input data $A = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n] \in \mathbb{R}^{d \times n}$ and its partitioning $\mathcal{V} = \{V_1, V_2, \ldots, V_K\}$. Assume that all the centroids $\{\mathbf{a}^{(1)}, \mathbf{a}^{(2)}, \ldots, \mathbf{a}^{(K)}\}$ are distinct. Let $q \geq 1$ be the conjugate index of p such that $\frac{1}{p} + \frac{1}{q} = 1$. Denote the optimal solution of (2) by $\{\mathbf{x}_i^*\}$ and define the map $\phi(\mathbf{a}_i) = \mathbf{x}_i^*$ for $i = 1, \ldots, n$.*

1. *Let*

$$\mu_{ij}^{(\alpha)} := \sum_{\beta=1,\beta\neq\alpha}^{K} \Big| w_i^{(\beta)} - w_j^{(\beta)} \Big|, \quad i,j \in I_\alpha,\ \alpha = 1,2,\ldots,K.$$

Assume that $w_{ij} > 0$ *and* $n_\alpha w_{ij} > \mu_{ij}^{(\alpha)}$ *for all* $i,j \in I_\alpha$, $\alpha = 1,\ldots,K$. *Let*

$$\begin{array}{l} \gamma_{\min} \ := \ \max_{1\le\alpha\le K} \max_{i,j\in I_\alpha} \left\{ \frac{\|\mathbf{a}_i - \mathbf{a}_j\|_q}{n_\alpha w_{ij} - \mu_{ij}^{(\alpha)}} \right\}, \\ \gamma_{\max} := \min_{1\le\alpha<\beta\le K} \left\{ \frac{\|\mathbf{a}^{(\alpha)} - \mathbf{a}^{(\beta)}\|_q}{\frac{1}{n_\alpha}\sum_{1\le l\le K, l\neq\alpha} w^{(\alpha,l)} + \frac{1}{n_\beta}\sum_{1\le l\le K, l\neq\beta} w^{(\beta,l)}} \right\}. \end{array} \tag{11}$$

If $\gamma_{\min} < \gamma_{\max}$ *and* γ *is chosen such that* $\gamma \in [\gamma_{\min}, \gamma_{\max})$, *then the map* ϕ *perfectly recovers* $\mathcal{V}$.

2. *If* γ *is chosen such that*

$$\gamma_{\min} \ \le \ \gamma \ < \max_{1\le\alpha\le K} \frac{n_\alpha \|\mathbf{c} - \mathbf{a}^{(\alpha)}\|_q}{\sum_{1\le\beta\le K,\beta\neq\alpha} w^{(\alpha,\beta)}},$$

where $\mathbf{c} = \frac{1}{n}\sum_{i=1}^n \mathbf{a}_i$, *then the map* ϕ *perfectly recovers a non-trivial coarsening of* $\mathcal{V}$.

Proof First we introduce the following centroid optimization problem corresponding to (2):

$$\min\Big\{ \frac{1}{2}\sum_{\alpha=1}^K n_\alpha \|\mathbf{x}^{(\alpha)} - \mathbf{a}^{(\alpha)}\|^2 + \gamma \sum_{\alpha=1}^K \sum_{\beta=\alpha+1}^K w^{(\alpha,\beta)} \|\mathbf{x}^{(\alpha)} - \mathbf{x}^{(\beta)}\|_p \mid \mathbf{x}^{(1)},\ldots,\mathbf{x}^{(K)} \in \mathbb{R}^d \Big\}. \tag{12}$$

Denote the optimal solution of (12) by $\{\bar{\mathbf{x}}^{(\alpha)} \mid \alpha = 1,2,\ldots,K\}$. The proof will rely on the relationships between (2) and (12).

(1a) First we show that, if $\gamma < \gamma_{\max}$, then $\bar{\mathbf{x}}^{(\alpha)} \neq \bar{\mathbf{x}}^{(\beta)}$ for all $\alpha \neq \beta$. From the optimality condition of (12), we have that

$$n_\alpha(\bar{\mathbf{x}}^{(\alpha)} - \mathbf{a}^{(\alpha)}) + \gamma \sum_{\beta=1,\beta\neq\alpha}^K w^{(\alpha,\beta)} \bar{\mathbf{z}}^{(\alpha,\beta)} \ = \ 0, \quad \forall\, \alpha = 1,\ldots,K, \tag{13}$$

where $\bar{\mathbf{z}}^{(\alpha,\beta)} \in \partial h(\bar{\mathbf{x}}^{(\alpha)} - \bar{\mathbf{x}}^{(\beta)})$ and $\bar{\mathbf{z}}^{(\alpha,\beta)} = -\bar{\mathbf{z}}^{(\beta,\alpha)}$ for $\alpha \neq \beta$. From (13), we get for $\alpha \neq \beta$,

$$\begin{array}{rll} & \bar{\mathbf{x}}^{(\alpha)} - \bar{\mathbf{x}}^{(\beta)} & = \mathbf{a}^{(\alpha)} - \mathbf{a}^{(\beta)} - \frac{\gamma}{n_\alpha}\sum_{l=1,l\neq\alpha}^K w^{(\alpha,l)}\bar{\mathbf{z}}^{(\alpha,l)} + \frac{\gamma}{n_\beta}\sum_{l=1,l\neq\beta}^K w^{(\beta,l)}\bar{\mathbf{z}}^{(\beta,l)} \\ \Rightarrow & \|\bar{\mathbf{x}}^{(\alpha)} - \bar{\mathbf{x}}^{(\beta)}\|_q & \ge \|\mathbf{a}^{(\alpha)} - \mathbf{a}^{(\beta)}\|_q - \frac{\gamma}{n_\alpha}\sum_{l=1,l\neq\alpha}^K w^{(\alpha,l)}\|\bar{\mathbf{z}}^{(\alpha,l)}\|_q - \frac{\gamma}{n_\beta}\sum_{l=1,l\neq\beta}^K w^{(\beta,l)}\|\bar{\mathbf{z}}^{(\beta,l)}\|_q \\ & & \ge \|\mathbf{a}^{(\alpha)} - \mathbf{a}^{(\beta)}\|_q - \gamma\Big(\frac{1}{n_\alpha}\sum_{l=1,l\neq\alpha}^K w^{(\alpha,l)} + \frac{1}{n_\beta}\sum_{l=1,l\neq\beta}^K w^{(\beta,l)}\Big) \\ & & \ge \|\mathbf{a}^{(\alpha)} - \mathbf{a}^{(\beta)}\|_q \left(1 - \frac{\gamma}{\gamma_{\max}}\right) \ > \ 0. \end{array}$$

Thus $\bar{\mathbf{x}}^{(\alpha)} \neq \bar{\mathbf{x}}^{(\beta)}$ for all $\alpha \neq \beta$.

(1b) Next we prove that if $\gamma \geq \gamma_{\min}$, then

$$\mathbf{x}_i^* = \bar{\mathbf{x}}^{(\alpha)}, \quad \forall\, i \in I_\alpha, \quad \alpha = 1, \ldots, K$$

is the unique optimal solution of (2).
To do so, we start with the optimality condition for (2), which is given as follows:

$$\mathbf{x}_i - \mathbf{a}_i + \gamma \sum_{j=1, j\neq i}^{n} w_{ij}\mathbf{z}_{ij} = 0, \;\; i = 1, 2, \ldots, n, \tag{14}$$

where $\mathbf{z}_{ij} \in \partial h(\mathbf{x}_i - \mathbf{x}_j)$. Consider

$$\mathbf{z}_{ij}^* \;=\; \begin{cases} \bar{\mathbf{z}}^{(\alpha,\beta)} & \text{if } i \in I_\alpha,\, j \in I_\beta,\, 1 \leq \alpha, \beta \leq K,\, \alpha \neq \beta, \\ \frac{1}{n_\alpha w_{ij}}\left[\frac{1}{\gamma}(\mathbf{a}_i - \mathbf{a}_j) - (\mathbf{p}_i^{(\alpha)} - \mathbf{p}_j^{(\alpha)})\right] & \text{if } i, j \in I_\alpha,\, i \neq j,\, \alpha = 1, \ldots, K, \end{cases}$$

where

$$\mathbf{p}_i^{(\alpha)} \;=\; \sum_{\beta=1, \beta\neq\alpha}^{K} \left[w_i^{(\beta)} - \frac{1}{n_\alpha} w^{(\alpha,\beta)}\right] \bar{\mathbf{z}}^{(\alpha,\beta)}.$$

We can readily prove that

$$\|\mathbf{p}_i^{(\alpha)} - \mathbf{p}_j^{(\alpha)}\|_q \;\leq\; \mu_{ij}^{(\alpha)}$$

and

$$\begin{aligned} \sum_{j\in I_\alpha} \mathbf{p}_j^{(\alpha)} \;&=\; \sum_{j\in I_\alpha} \left(\sum_{\beta=1, \beta\neq\alpha}^{K} \left[w_j^{(\beta)} - \frac{1}{n_\alpha} w^{(\alpha,\beta)}\right] \bar{\mathbf{z}}^{(\alpha,\beta)} \right) \\ &=\; \sum_{\beta=1, \beta\neq\alpha}^{K} \left(\sum_{j\in I_\alpha} \left[w_j^{(\beta)} - \frac{1}{n_\alpha} w^{(\alpha,\beta)}\right] \right) \bar{\mathbf{z}}^{(\alpha,\beta)} \;=\; \mathbf{0}. \end{aligned}$$

For convenience, we set $\mathbf{z}_{ii}^* = 0$ for $i = 1, 2, \ldots, n$. Now, we show that $\mathbf{z}_{ij}^* \in \partial h(\mathbf{x}_i^* - \mathbf{x}_j^*)$. If $i \in I_\alpha$ and $j \in I_\beta$ for $\alpha \neq \beta$, then we have that

$$\mathbf{z}_{ij}^* = \bar{\mathbf{z}}^{(\alpha,\beta)} \in \partial h(\bar{\mathbf{x}}^{(\alpha)} - \bar{\mathbf{x}}^{(\beta)}) = \partial h(\mathbf{x}_i^* - \mathbf{x}_j^*).$$

It remains to show that $\|\mathbf{z}_{ij}^*\|_q \leq 1$ for all $i, j \in I_\alpha, \alpha = 1, 2, \ldots, K$. By direct calculations, we have that for $\gamma \geq \gamma_{\min}$,

$$\begin{aligned} \|\mathbf{z}_{ij}^*\|_q \;&=\; \frac{1}{n_\alpha w_{ij}} \left\| \frac{1}{\gamma}(\mathbf{a}_i - \mathbf{a}_j) - (\mathbf{p}_i^{(\alpha)} - \mathbf{p}_j^{(\alpha)}) \right\|_q \;\leq\; \frac{1}{\gamma n_\alpha w_{ij}} \|\mathbf{a}_i - \mathbf{a}_j\|_q + \frac{1}{n_\alpha w_{ij}} \mu_{ij}^{(\alpha)} \\ &\leq\; \frac{1}{n_\alpha w_{ij}} (n_\alpha w_{ij} - \mu_{ij}^{(\alpha)}) + \frac{1}{n_\alpha w_{ij}} \mu_{ij}^{(\alpha)} \;=\; 1, \end{aligned}$$

which implies that $\mathbf{z}_{ij}^* \in \partial h(\mathbf{x}_i^* - \mathbf{x}_j^*) = \partial h(\mathbf{0})$ for all $i, j \in I_\alpha$.

Finally, we show that the optimality condition (14) holds for $(\mathbf{x}_1^*, \ldots, \mathbf{x}_n^*)$. We have that for $i \in I_\alpha$,

$$\begin{aligned}
\mathbf{x}_i^* - \mathbf{a}_i + \gamma \sum_{j=1, j\neq i}^{n} w_{ij}\mathbf{z}_{ij}^* &= \bar{\mathbf{x}}^{(\alpha)} - \mathbf{a}_i + \gamma \sum_{\beta=1}^{K} \sum_{j\in I_\beta} w_{ij}\mathbf{z}_{ij}^* \\
&= \bar{\mathbf{x}}^{(\alpha)} - \mathbf{a}^{(\alpha)} + \gamma \sum_{\beta=1, \beta\neq\alpha}^{K} \Big(\sum_{j\in I_\beta} w_{ij} \Big) \bar{\mathbf{z}}^{(\alpha,\beta)} + \mathbf{a}^{(\alpha)} - \mathbf{a}_i + \gamma \sum_{j\in I_\alpha} w_{ij}\mathbf{z}_{ij}^* \\
&= \gamma \sum_{\beta=1, \beta\neq\alpha}^{K} \Big[w_i^{(\beta)} - \frac{1}{n_\alpha} w^{(\alpha,\beta)} \Big] \bar{\mathbf{z}}^{(\alpha,\beta)} + \mathbf{a}^{(\alpha)} - \mathbf{a}_i + \gamma \sum_{j\in I_\alpha} w_{ij}\mathbf{z}_{ij}^* \\
&= \gamma \mathbf{p}_i^{(\alpha)} + \mathbf{a}^{(\alpha)} - \mathbf{a}_i + \frac{\gamma}{n_\alpha} \sum_{j\in I_\alpha} \Big[\frac{1}{\gamma}(\mathbf{a}_i - \mathbf{a}_j) - (\mathbf{p}_i^{(\alpha)} - \mathbf{p}_j^{(\alpha)}) \Big] \\
&= 0.
\end{aligned}$$

Thus $(\mathbf{x}_1^*, \ldots, \mathbf{x}_n^*)$ is the optimal solution of (2).

From (1a) and (1b), we see that if $\gamma \in [\gamma_{\min}, \gamma_{\max})$, then $\phi(\mathbf{a}_i) = \mathbf{x}_i^* = \bar{\mathbf{x}}^{(\alpha)}$ for all $i \in I_\alpha$, $\alpha = 1, \ldots, K$, and $\bar{\mathbf{x}}^\alpha \neq \bar{\mathbf{x}}^\beta$ for all $\alpha \neq \beta$. Thus the mapping ϕ perfectly recovers the clusters in $\mathcal{V}$.

(2) Suppose on the contrary that $\bar{\mathbf{x}}^{(1)} = \bar{\mathbf{x}}^{(2)} = \cdots = \bar{\mathbf{x}}^{(K)} =: \bar{\mathbf{x}}$. Then from the optimality condition (13), we have

$$n_\alpha(\bar{\mathbf{x}} - \mathbf{a}^{(\alpha)}) + \gamma \sum_{\beta=1, \beta\neq\alpha}^{K} w^{(\alpha,\beta)} \bar{\mathbf{z}}^{(\alpha,\beta)} = 0, \quad \forall\, \alpha = 1, \ldots, K,$$

where $\bar{\mathbf{z}}^{(\alpha,\beta)} \in \partial h(0)$ and $\bar{\mathbf{z}}^{(\alpha,\beta)} = -\bar{\mathbf{z}}^{(\beta,\alpha)}$. By summing up the above equation over α, we get

$$0 = \Big(\sum_{\alpha=1}^{K} n_\alpha \Big) \bar{\mathbf{x}} - \sum_{\alpha=1}^{K} n_\alpha \mathbf{a}^{(\alpha)} + \gamma \sum_{\beta=2}^{K} \sum_{\alpha=1}^{\beta-1} w^{(\alpha,\beta)} (\bar{\mathbf{z}}^{(\alpha,\beta)} + \bar{\mathbf{z}}^{(\beta,\alpha)}) = n\bar{\mathbf{x}} - \sum_{i=1}^{n} \mathbf{a}_i.$$

Thus $\bar{\mathbf{x}} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{a}_i = \mathbf{c}$. Hence the optimality condition (13) gives

$$n_\alpha \|\mathbf{c} - \mathbf{a}^{(\alpha)}\|_q \leq \gamma \sum_{\beta=1, \beta\neq\alpha}^{K} w^{(\alpha,\beta)}, \ \ \forall\, \alpha \in \{1, 2, \ldots, K\}.$$

This implies that

$$\gamma \geq \max_{1\leq\alpha\leq K} \frac{n_\alpha \|\mathbf{c} - \mathbf{a}^{(\alpha)}\|_q}{\sum_{\beta=1, \beta\neq\alpha}^{K} w^{(\alpha,\beta)}},$$

which is a contradiction. Thus $\{\bar{\mathbf{x}}^{(1)}, \ldots, \bar{\mathbf{x}}^{(K)}\}$ must have a distinct pair. ■

The above theorem has established the theoretical recovery guarantee for the general weighted convex clustering model (2). Later, we will demonstrate that the sufficient conditions that γ must satisfy is practically meaningful in the numerical experiments section. Next, we explain the derived sufficient conditions.

Intuitively, we can get meaningful clustering results when the given data set has the properties that the data points within the same cluster are "tight" (in other words, the diameter should be small) and the centroids for different clusters are well separated. Indeed, the conditions we have established are consistent with the intuition we just mentioned. On the one hand, the lower bound $\gamma_{\min}$, which is defined in (11), characterizes the maximum weighted distance between the data points in the same cluster. On the other hand, the upper bound $\gamma_{\max}$ characterizes the minimum weighted distance between different centroids. Thus based on our discussion, we can expect perfect recovery to be practically possible for the weighted convex clustering model if the upper bound is larger than the lower bound.

Remark 6 *(a) Note that in Theorem 5, the assumption that $w_{ij} > 0$ is only needed for all the pairs (i, j) belonging to the same cluster I_α for all $1 \leq \alpha \leq K$. Thus the weights w_{ij} can be chosen to be zero if $\mathbf{a}_i$ and $\mathbf{a}_j$ belong to different clusters. As a result, the number of pairwise differences in the regularization term can be much fewer than the total of $n(n-1)/2$ terms. For example, if all the K clusters have the same size of n/K, then theoretically the number of edges in $\mathcal{E}$ is roughly equal to $n^2/(2K)$. This implies that we can gain substantial computational efficiency when dealing with the sparse weighted regularization term, especially if K is large (say more than 10). On the other hand, the theoretical gain in the computational efficiency is less significant if K is small (say less than 5).*

(b) We should mention that in practice, one does not know the index sets I_α, $\alpha = 1, \ldots, K$, of the true clusters. Thus one can only decide whether to put an edge (i, j) in $\mathcal{E}$ based on the proximity of the points $\mathbf{a}_i$ and $\mathbf{a}_j$. In our paper, we use the k-nearest neighbors scheme to construct $\mathcal{E}$. In that case, the number of pairwise differences in the regularizaton term is at most kn, and k is a constant that is typically chosen to be much smaller than n. Thus in practice, there is a significant gain in the computational efficiency by using the sparse convex clustering model compared to the full model (1).

(c) The quantity $\mu_{ij}^{(\alpha)} = \sum_{\beta=1,\beta\neq\alpha}^{K} |w_i^{(\beta)} - w_j^{(\beta)}|$, for $i, j \in I_\alpha$, measures the total difference in the couplings between $\mathbf{a}_i$ and $\mathbf{a}_j$ with the β-th cluster for all $\beta \neq \alpha$.

Next, we show that the results in Theorem 4 are special cases of our results. Therefore, we also include the result in (Zhu et al., 2014) as a special case.

Corollary 7 *In (2), if we take $w_{ij} = 1$ for all $1 \leq i < j \leq n$, then the results in Theorem 5 reduce to the following.*

(i) If

$$\max_{1\leq\alpha\leq K} \frac{D_q(V_\alpha)}{|V_\alpha|} \leq \gamma < \min_{1\leq\alpha,\beta\leq K,\alpha\neq\beta} \Big\{ \frac{\|\mathbf{a}^{(\alpha)} - \mathbf{a}^{(\beta)}\|_q}{2n - n_\alpha - n_\beta} \Big\},$$

then the map ϕ perfectly recovers $\mathcal{V}$.

(ii) If

$$\max_{1\le\alpha\le K}\frac{D_q(V_\alpha)}{|V_\alpha|}\ \le\ \gamma\ \le\ \max_{V\in\mathcal{V}}\frac{\|c(A)-c(V)\|_q}{|A|-|V|},$$

then the map ϕ perfectly recovers a non-trivial coarsening of $\mathcal{V}$.

Proof The results for this corollary follow directly from Theorem 5 by noting that $D_q(V_\alpha) = \max_{i,j\in I_\alpha}\|\mathbf{a}_i-\mathbf{a}_j\|_q/n_\alpha$, and using the following facts for the special case:

(1) $\mu_{ij}^{(\alpha)} = \sum_{\beta=1,\beta\neq\alpha}^K |w_i^{(\beta)} - w_j^{(\beta)}| = \sum_{\beta=1,\beta\neq\alpha}^K |n_\beta - n_\beta| = 0$, for all $i,j\in I_\alpha$, $1\le\alpha\le K$.

(2) $\frac{1}{n_\alpha}\sum_{\beta=1,\beta\neq\alpha}^K w^{(\alpha,\beta)} = \frac{1}{n_\alpha}\sum_{\beta=1,\beta\neq\alpha}^K n_\alpha n_\beta = n - n_\alpha$, for all $1\le\alpha\le K$.

We omit the details here. ■

If we compare the upper bound we obtained for γ in part (i) of Corollary 7 to that obtained in Theorem 4 of (Panahi et al., 2017) for the case $p=2$ (and hence $q=2$), we can see that our upper bound is more relax in the sense that

$$\min_{1\le\alpha,\beta\le K,\alpha\neq\beta}\left\{\frac{\|\mathbf{a}^{(\alpha)}-\mathbf{a}^{(\beta)}\|_2}{2n-n_\alpha-n_\beta}\right\}\ >\ \min_{1\le\alpha,\beta\le K,\alpha\neq\beta}\left\{\frac{\|\mathbf{a}^{(\alpha)}-\mathbf{a}^{(\beta)}\|_2}{2n}\right\} = \frac{d_2(\mathcal{C}(\mathcal{V}))}{2n}\ \ge\ \frac{d_2(\mathcal{C}(\mathcal{V}))}{2n\sqrt{K}}.$$

5. A Semismooth Newton-CG Augmented Lagrangian Method for Solving (2)

In this section, we introduce a fast convergent ALM for solving the weighted convex clustering model (2)[1]. For simplicity, we will only focus on designing a highly efficient algorithm to solve (2) with $p=2$. The other cases can be done in a similar way. In particular, the same algorithmic design and implementation can be applied to the case $p=1$ or $p=\infty$ without much difficulty.

5.1 Duality and Optimality Conditions

From now on, we will focus on the following weighted convex clustering model with the vector 2-norm regularization:

$$\min_{X\in\mathbb{R}^{d\times n}}\frac{1}{2}\sum_{i=1}^n\|\mathbf{x}_i-\mathbf{a}_i\|^2+\gamma\sum_{i<j}w_{ij}\|\mathbf{x}_i-\mathbf{x}_j\|_2.$$

By ignoring the terms with $w_{ij}=0$, we consider the following problem:

$$\min_{X\in\mathbb{R}^{d\times n}}\frac{1}{2}\sum_{i=1}^n\|\mathbf{x}_i-\mathbf{a}_i\|^2+\gamma\sum_{(i,j)\in\mathcal{E}}w_{ij}\|\mathbf{x}_i-\mathbf{x}_j\|_2, \tag{15}$$

1. Part of the results described in this section has been published in the ICML 2018 paper (Yuan et al., 2018).

where $\mathcal{E} := \{(i,j) \mid w_{ij} > 0\}$.
Now, we present the dual problem of (15) and its Karush-Kuhn-Tucker (KKT) conditions. First, we write (15) equivalently in the following compact form

$$(P) \quad \min_{X,U} \Big\{ \frac{1}{2}\|X - A\|^2 + p(U) \mid \mathcal{B}(X) - U = 0 \Big\},$$

where $p(U) = \gamma \sum_{(i,j)\in\mathcal{E}} w_{ij}\|U^{l(i,j)}\|$ and $\mathcal{B}$ is the linear map defined in (7). Here $U^{l(i,j)}$ denotes the $l(i,j)$-th column of $U \in \mathbb{R}^{d\times|\mathcal{E}|}$. The dual problem for (P) is given by

$$(D) \quad \max_{V,Z} \Big\{ \langle A, V\rangle - \frac{1}{2}\|V\|^2 \ \mid \ \mathcal{B}^*(Z) - V = 0, Z \in \Omega \Big\},$$

where $\Omega = \{Z \in \mathbb{R}^{d\times|\mathcal{E}|} \ \mid \ \|Z^{l(i,j)}\| \le \gamma w_{ij}, (i,j) \in \mathcal{E}\}$. The KKT conditions for (P) and (D) are given by

$$(KKT) \quad \begin{cases} V + X - A &= 0, \\ U - \mathrm{Prox}_p(U + Z) &= 0, \\ \mathcal{B}(X) - U &= 0, \\ \mathcal{B}^*(Z) - V &= 0. \end{cases}$$

5.2 A Semismooth Newton-CG Augmented Lagrangian Method for Solving (P)

In this section, we will design an inexact ALM for solving the primal problem (P) but it will also solve (D) as a byproduct.

We begin by defining the following Lagrangian function for (P):

$$l(X,U;Z) \ = \ \frac{1}{2}\|X - A\|^2 + p(U) + \langle Z, \mathcal{B}(X) - U\rangle. \tag{16}$$

For a given parameter $\sigma > 0$, the augmented Lagrangian function associated with (P) is given by

$$\mathcal{L}_\sigma(X,U;Z) \ = \ l(X,U;Z) + \frac{\sigma}{2}\|\mathcal{B}(X) - U\|^2.$$

The algorithm for solving (P) is described in **Algorithm** 1. To ensure the convergence of the inexact ALM in **Algorithm** 1, we need the following stopping criterion for solving the subproblem (17) in each iteration:

$$(A) \quad \mathrm{dist}(0, \partial\Phi_k(X^{k+1}, U^{k+1})) \le \epsilon_k / \max\{1, \sqrt{\sigma_k}\}, \tag{18}$$

where $\{\epsilon_k\}$ is a given summable sequence of nonnegative numbers.

Since a semismooth Newton-CG method will be used to solve the subproblems involved in the above ALM method, we call our algorithm a semismooth Newton-CG augmented Lagrangian method (Ssnal in short).

Algorithm 1 SSNAL for (P)

Initialization: Choose $(X^0, U^0) \in \mathbb{R}^{d\times n} \times \mathbb{R}^{d\times|\mathcal{E}|}$, $Z^0 \in \mathbb{R}^{d\times|\mathcal{E}|}$, $\sigma_0 > 0$ and a summable nonnegative sequence $\{\epsilon_k\}$.

repeat

Step 1. Compute

$$(X^{k+1}, U^{k+1}) \approx \arg\min\{\Phi_k(X,U) = \mathcal{L}_{\sigma_k}(X,U;Z^k) \ \mid \ X \in \mathbb{R}^{d\times n},\ U \in \mathbb{R}^{d\times|\mathcal{E}|}\} \quad (17)$$

to satisfy the condition (A) with the tolerance ϵ_k.

Step 2. Compute

$$Z^{k+1} = Z^k + \sigma_k(\mathcal{B}(X^{k+1}) - U^{k+1}).$$

Step 3. Update $\sigma_{k+1} \uparrow \sigma_\infty \le \infty$.

until Stopping criterion is satisfied.

5.3 Solving the Subproblem (17)

The inexact ALM is a well studied algorithmic framework for solving convex composite optimization problems. The key challenge in making the ALM numerically efficient is in solving the subproblem (17) in each iteration efficiently to the required accuracy. Next, we will design a semismooth Newton-CG algorithm to solve (17). We will establish its superlinear (quadratic) convergence rate and develop sophisticated numerical techniques to solve the associated semismooth Newton equations very efficiently by exploiting the underlying second-order structured sparsity in the subproblems.

For a given σ and $\tilde{Z}$, the subproblem (17) in each iteration has the following form:

$$\min_{X\in\mathbb{R}^{d\times n}, U\in\mathbb{R}^{d\times|\mathcal{E}|}} \Phi(X,U) := \mathcal{L}_\sigma(X,U;\tilde{Z}). \quad (19)$$

Since $\Phi(\cdot,\cdot)$ is a strongly convex function, the level set $\{(X,U)|\Phi(X,U) \le \alpha\}$ is a closed and bounded convex set for any $\alpha \in \mathbb{R}$ and problem (19) admits a unique optimal solution which we denote as $(\bar{X}, \bar{U})$. Now, for any X, denote

$$\begin{aligned}\phi(X) &:= \inf_U \Phi(X,U) = \tfrac{1}{2}\|X-A\|^2 + \inf_U\Big\{p(U) + \tfrac{\sigma}{2}\|U - \mathcal{B}(X) - \sigma^{-1}\tilde{Z}\|^2\Big\} - \tfrac{1}{2\sigma}\|\tilde{Z}\|^2 \\ &= \tfrac{1}{2}\|X-A\|^2 + p(\mathrm{Prox}_{p/\sigma}(\mathcal{B}(X) + \sigma^{-1}\tilde{Z})) + \tfrac{1}{2\sigma}\|\mathrm{Prox}_{\sigma p^*}(\sigma\mathcal{B}(X) + \tilde{Z})\|^2 - \tfrac{1}{2\sigma}\|\tilde{Z}\|^2.\end{aligned}$$

Therefore, we can compute $(\bar{X}, \bar{U}) = \arg\min \Phi(X,U)$ by first computing

$$\bar{X} = \arg\min_X \phi(X),$$

and then compute $\bar{U} = \mathrm{Prox}_{p/\sigma}(\mathcal{B}(\bar{X}) + \sigma^{-1}\tilde{Z})$. Since $\phi(\cdot)$ is strongly convex and continuously differentiable on $\mathbb{R}^{d\times n}$ with

$$\nabla\phi(X) = X - A + \mathcal{B}^*(\mathrm{Prox}_{\sigma p^*}(\sigma\mathcal{B}(X) + \tilde{Z})), \quad (20)$$

we know that $\bar{X}$ can be obtained by solving the following nonlinear equation

$$\nabla\phi(X) = 0. \quad (21)$$

It is well known that for solving smooth nonlinear equations, the Newton's method is usually the first choice if it can be implemented efficiently. However, the usually required smoothness condition on $\nabla\phi(\cdot)$ is not satisfied in our problem. This motivates us to develop a semismooth Newton method to solve the nonsmooth equation (21). Before we present our semismooth Newton method, we introduce the following definition of semismoothness, adopted from (Mifflin, 1977; Kummer, 1988; Qi and Sun, 1993), which will be useful for analysis.

Definition 8 *(Semismoothness). For a given open set $\mathcal{O} \subseteq \mathbb{R}^n$, let $F : \mathcal{O} \to \mathbb{R}^m$ be a locally Lipschitz continuous function and $\mathcal{G} : \mathcal{O} \rightrightarrows \mathbb{R}^{m\times n}$ be a nonempty compact valued upper-semicontinuous multifunction. F is said to be semismooth at $x \in \mathcal{O}$ with respect to the multifunction $\mathcal{G}$ if F is directionally differentiable at x and for any $V \in \mathcal{G}(x+\Delta x)$ with $\Delta x \to 0$,*

$$F(x+\Delta x) - F(x) - V\Delta x = o(\|\Delta x\|).$$

F is said to be strongly semismooth at $x \in \mathcal{O}$ with respect to $\mathcal{G}$ if it is semismooth at x with respect to $\mathcal{G}$ and

$$F(x+\Delta x) - F(x) - V\Delta x = O(\|\Delta x\|^2).$$

F is said to be a semismooth (respectively, strongly semismooth) function on $\mathcal{O}$ with respect to $\mathcal{G}$ if it is semismooth (respectively, strongly semismooth) everywhere in $\mathcal{O}$ with respect to $\mathcal{G}$.

The following lemma shows that the proximal mapping of the 2-norm is strongly semismooth with respect to its Clarke generalized Jacobian (See Clarke (1983) [Definition 2.6.1] for the definition of the Clarke generalized Jacobian).

Lemma 9 (Zhang et al. (2020), Lemma 2.1) *For any $t > 0$, the proximal mapping $\mathrm{Prox}_{t\|\cdot\|_2}$ is strongly semismooth with respect to the Clarke generalized Jacobian $\partial\mathrm{Prox}_{t\|\cdot\|_2}(\cdot)$.*

Next we derive the generalized Jacobian of the locally Lipschitz continuous function $\nabla\phi(\cdot)$. For any given $X \in \mathbb{R}^{d\times n}$, the following set-valued map is well defined:

$$\begin{aligned}\hat{\partial}^2\phi(X) &:= \{\mathcal{I} + \sigma\mathcal{B}^*\mathcal{V}\mathcal{B} \mid \mathcal{V} \in \partial\mathrm{Prox}_{\sigma p^*}(\tilde{Z} + \sigma\mathcal{B}X)\} \\ &= \{\mathcal{I} + \sigma\mathcal{B}^*(\mathcal{I}-\mathcal{P})\mathcal{B} \mid \mathcal{P} \in \partial\mathrm{Prox}_{p/\sigma}(\tfrac{1}{\sigma}\tilde{Z} + \mathcal{B}X)\}, \end{aligned} \tag{22}$$

where $\partial\mathrm{Prox}_{\sigma p^*}(\tilde{Z}+\sigma\mathcal{B}X)$ and $\partial\mathrm{Prox}_{p/\sigma}(\frac{1}{\sigma}\tilde{Z}+\mathcal{B}(X))$ are the Clarke generalized Jacobians of the Lipschitz continuous mappings $\mathrm{Prox}_{\sigma p^*}(\cdot)$ and $\mathrm{Prox}_{p/\sigma}(\cdot)$ at $\tilde{Z}+\sigma\mathcal{B}X$ and $\frac{1}{\sigma}\tilde{Z}+\mathcal{B}X$, respectively. Note that from (Clarke, 1983) [p.75] and (Hiriart-Urruty et al., 1984) [Example 2.5], we have that

$$\partial^2\phi(X)(d) = \hat{\partial}^2\phi(X)(d), \quad \forall d \in \mathbb{R}^{d\times n},$$

where $\partial^2\phi(X)$ is the generalized Hessian of ϕ at X. Since $\mathcal{I}-\mathcal{P} = \mathcal{V} \in \partial\mathrm{Prox}_{\sigma p^*}(\cdot)$ is symmetric and positive semdefinite, the elements in $\hat{\partial}^2\phi(X)$ are positive definite, which guarantees that (23) in **Algorithm** 2 is well defined.

Now, we can present our semismooth Newton-CG (Ssncg) method for solving (21) and we could expect to get a fast superlinear or even quadratic convergence rate.

Algorithm 2 SSNCG for (21)

Initialization: Given $X^0 \in \mathbb{R}^{d\times n}$, $\mu \in (0, 1/2)$, $\tau \in (0, 1]$, and $\bar{\eta}, \delta \in (0, 1)$. For $j = 0, 1, \ldots$

repeat

Step 1. Pick an element $\mathcal{H}_j$ in $\hat{\partial}^2\phi(X^j)$ that is defined in (22). Apply the conjugate gradient (CG) method to find an approximate solution $d^j \in \mathbb{R}^{d\times n}$ to

$$\mathcal{H}_j(d) \approx -\nabla\phi(X^j) \tag{23}$$

such that $\|\mathcal{H}_j(d^j) + \nabla\phi(X^j)\| \le \min(\bar{\eta}, \|\nabla\phi(X^j)\|^{1+\tau})$.

Step 2. (Armijo line search (Nocedal and Wright, 2006)[sec. 3.1]) Set $\alpha_j = \delta^{m_j}$, where m_j is the first nonnegative integer m for which

$$\phi(X^j + \delta^m d^j) \le \phi(X^j) + \mu\delta^m\langle\nabla\phi(X^j), d^j\rangle.$$

Step 3. Set $X^{j+1} = X^j + \alpha_j d^j$.

until Stopping criterion based on $\|\nabla\phi(X^{j+1})\|$ is satisfied.

To close this section, we discuss an implementable stopping criterion for (18) in the algorithm SSNAL. Note that the algorithm SSNCG is applied to solve the optimization problem in Step 1 of SSNAL via

$$X^{k+1} = \arg\min_X \phi_k(X) \quad \text{and} \quad U^{k+1} = \mathrm{Prox}_{p/\sigma}(\mathcal{B}(X^{k+1} + \sigma^{-1}Z^k)),$$

where $\phi_k(\cdot) := \inf_U \Phi_k(X, U)$. Thus we have

$$(\nabla\phi_k(X^{k+1}), 0) \in \partial\Phi_k(X^{k+1}, U^{k+1}).$$

Therefore, the stopping criterion (A) could be achieved by the following implementable stopping criterion:

$$\text{(A')} \quad \|\nabla\phi_k(X^{k+1})\| \le \epsilon_k/\max(1, \sqrt{\sigma_k}), \quad \textstyle\sum_{k=0}^{\infty}\epsilon_k < \infty. \tag{24}$$

5.4 Using the Conjugate Gradient Method to Solve (23)

In this section, we will discuss how to solve the very large (of dimension $dn \times dn$) symmetric positive definite linear system (23) to compute the Newton direction efficiently. As the matrix representation of the coefficient linear operator $\mathcal{H}_j$ in (23) is expensive to compute and factorize, we will adopt the conjugate gradient (CG) method to solve it. It is well known that the convergence rate of the CG method depends critically on the condition number of the coefficient matrix. Fortunately, for our linear system (23), the coefficient linear operator typically has a moderate condition number since it satisfies the following condition:

$$I \preceq \mathcal{H}_j \preceq I + \sigma\mathcal{B}^*\mathcal{B} \preceq (1 + \sigma\lambda_{\max}(L_G))I,$$

where $\lambda_{\max}(L_G)$ denotes the maximum eigenvalue of the Laplacian matrix L_G of the graph $\mathcal{G}$, and the notation "$A \preceq B$" means that $B - A$ is symmetric positive semidefinite. It is

known from Anderson and Morley (1985) that $\lambda_{\max}(G)$ is at most 2 times the maximum degree of the graph. In the numerical experiments, the maximum degree of the graph is roughly equal to the number of k nearest neighbors. In those cases, the condition number of $\mathcal{H}_j$ is bounded independent of dn, and provided that σ is not too large, we can expect the CG method to converge rapidly even when n and/or d are large.

The computational cost for each CG step is highly dependent on the cost for computing the matrix-vector product $\mathcal{H}_j(\tilde{d})$ for any given $\tilde{d} \in \mathbb{R}^{d\times n}$. Thus we need to analyze how this product can be computed efficiently. Let $D := \mathcal{B}X^j + \sigma^{-1}\tilde{Z}$. For $(i,j) \in \mathcal{E}$, define

$$\alpha_{ij} = \begin{cases} \frac{\sigma^{-1}\gamma w_{ij}}{\|D^{l(i,j)}\|} & \text{if } \|D^{l(i,j)}\| > 0, \\ \infty & \text{otherwise.} \end{cases}$$

Note that for the given $D \in \mathbb{R}^{d\times|\mathcal{E}|}$, the cost for computing α is $O(d|\mathcal{E}|)$ arithmetic operations. For later convenience, denote

$$\widehat{\mathcal{E}} = \{(i,j) \in \mathcal{E} \mid \alpha_{ij} < 1\}.$$

Now we choose $\mathcal{P} \in \partial \mathrm{Prox}_{p/\sigma}(D)$ explicitly. We can take $\mathcal{P} : \mathbb{R}^{d\times|\mathcal{E}|} \to \mathbb{R}^{d\times|\mathcal{E}|}$ that is defined by

$$(\mathcal{P}(U))^{l(i,j)} = \begin{cases} \alpha_{ij}\frac{\langle D^{l(i,j)}, U^{l(i,j)}\rangle}{\|D^{l(i,j)}\|^2} D^{l(i,j)} + (1-\alpha_{ij})U^{l(i,j)} & \text{if } (i,j) \in \widehat{\mathcal{E}}, \\ 0 & \text{otherwise.} \end{cases}$$

Thus to compute $\mathcal{H}_j(X) = X + \sigma\mathcal{B}^*\mathcal{B}(X) - \sigma\mathcal{B}^*\mathcal{P}\mathcal{B}(X) = X(I_n + \sigma L_G) - \sigma\mathcal{B}^*\mathcal{P}\mathcal{B}(X)$ efficiently for a given $X \in \mathbb{R}^{d\times n}$, we need the efficient computation of $\mathcal{B}^*\mathcal{P}\mathcal{B}(X)$ by using the following proposition.

Proposition 10 *Let $X \in \mathbb{R}^{d\times n}$ be given.*
(a) Consider the symmetric matrix $M \in \mathbb{R}^{n\times n}$ defined by $M_{ij} = 1 - \alpha_{ij}$ if $(i,j) \in \widehat{\mathcal{E}}$ and $M_{ij} = 0$ otherwise. Let $Y = [M_{ij}(\mathbf{x}_i - \mathbf{x}_j)]_{(i,j)\in\mathcal{E}} = X\mathcal{M}$, where $\mathcal{M}$ is defined similarly as in (5) for the matrix M. Then we have

$$\mathcal{B}^*(Y) = XL_M,$$

where L_M is the Laplacian matrix associated with M. The cost of computing the result $\mathcal{B}^*(Y)$ is $O(d|\widehat{\mathcal{E}}|)$ arithmetic operations.

(b) Define $\rho \in \mathbb{R}^{|\mathcal{E}|}$ by

$$\rho_{l(i,j)} := \begin{cases} \frac{\alpha_{ij}}{\|D^{l(i,j)}\|^2}\langle D^{l(i,j)}, \mathbf{x}_i - \mathbf{x}_j\rangle, & \text{if } (i,j) \in \widehat{\mathcal{E}}, \\ 0, & \text{otherwise.} \end{cases}$$

For the given $D \in \mathbb{R}^{d\times|\mathcal{E}|}$, the cost for computing ρ is $O(d|\widehat{\mathcal{E}}|)$ arithmetic operations. Let $W^{l(i,j)} = \rho_{l(i,j)}D^{l(i,j)}$. Then,

$$\mathcal{B}^*(W) = W\mathcal{J}^T = D\mathrm{diag}(\rho)\mathcal{J}^T.$$

(c) The computing cost for $\mathcal{B}^*\mathcal{P}\mathcal{B}(X) = \mathcal{B}^*(Y) + \mathcal{B}^*(W)$ in total is $O(d|\widehat{\mathcal{E}}|)$.

With the above proposition, we can readily see that $\mathcal{H}_j(X)$ can be computed in $O(d|\mathcal{E}|)+O(d|\widehat{\mathcal{E}}|)$ operations, where the first term comes from computing $X(I+\sigma L_G)$ and the second term comes from computing $\sigma\mathcal{B}\mathcal{P}\mathcal{B}^*(X)$ based on Proposition 10.

Besides the algorithmic aspect, the next remark shows that the second-order information gathered in the semismooth Newton method can possiblely capture data points which are near to the boundary of a cluster if we wisely choose the weights w_{ij}. We believe this is a very useful result since boundary points detection is a challenging problem in practice, especially in the high dimensional setting where locating boundary points is challenging even if we know the labels of all the data points.

Remark 11 *If we choose the weights based on the k-nearest neighbors, for example, set*

$$w_{ij} = \begin{cases} \exp(-\phi\|\mathbf{a}_i - \mathbf{a}_j\|^2) & if\ (i,j) \in \mathcal{E}, \\ 0 & otherwise, \end{cases}$$

where $\mathcal{E} = \cup_{i=1}^n\{(i,j) \mid \mathbf{a}_j$ *is among* $\mathbf{a}_i$*'s k-nearest neighbors,*$i < j \le n\}$*. Then, if* γ *is properly chosen, after we get the optimal solution,* $\alpha_{ij} < 1$ *means that* $\mathbf{a}_j$ *is among* $\mathbf{a}_i$*'s k-nearest neighbors but they do not belong to the same cluster.*[2] *Naturally we expect there will only be a small number of such occurrences. Hence,* $|\widehat{\mathcal{E}}|$ *is expected to be much smaller than* $|\mathcal{E}|$*. On the other hand, for* $\alpha_{ij} \ge 1$*, it means that data points* $\mathbf{a}_i$ *and* $\mathbf{a}_j$ *are in the same cluster. This result implies that after we have solved the optimization problem (2) with a properly selected* γ*,* $\alpha_{ij} < 1$ *indicates that point i is near to the boundary of its cluster. Also, we can expect most of the columns of the matrix* $\mathcal{P}(\mathcal{B}(X))$ *to be zero since its number of non-zero columns is at most* $|\widehat{\mathcal{E}}|$*. We call such a property inherited from the generalized Hessian of* $\phi(\cdot)$ *at* X *as the* **second-order sparsity***. This also explains why we are able to compute* $\mathcal{B}^*\mathcal{P}\mathcal{B}(X)$ *at a very low cost.*

5.5 Convergence Results

In this section, we will establish the convergence results for both Ssnal and Ssncg under mild assumptions. First, we present the following global convergence result of our proposed Algorithm Ssnal for solving (P).

Theorem 12 *Let* $\{(X^k, U^k, Z^k)\}$ *be the sequence generated by Algorithm 1 with stopping criterion* (A)*. Then the sequence* $\{X^k\}$ *converges to the unique optimal solution of* (P)*, and* $\|\mathcal{B}(X^k) - U^k\|$ *converges to* 0*. In addition,* $\{Z^k\}$ *is converges to an optimal solution* $Z^* \in \Omega$ *of* (D)*.*

The above convergence theorem can be obtained from (Rockafellar, 1976a,b) without much difficulties. Next, we state the convergence property for the semismooth Newton algorithm Ssncg, which is used to solve the subproblems in Algorithm 1.

Theorem 13 *Let the sequence* $\{X^j\}$ *be generated by Algorithm* Ssncg*. Then* $\{X^j\}$ *converges to the unique solution* $\bar{X}$ *of the problem in (21), and for j sufficiently large,*

$$\|X^{j+1} - \bar{X}\| = O(\|X^j - \bar{X}\|^{1+\tau}),$$

2. Since $\alpha_{ij} < 1$ indicates $\|D^{l(i,j)}\| > \sigma^{-1}\gamma w_{ij}$. With the property that $\|D^{l(i,j)}\| \le \sigma^{-1}\|Z^{l(i,j)}\| + \|\mathbf{x}_i - \mathbf{x}_j\|$ and $\|Z^{l(i,j)}\| \le \gamma w_{ij}$, we can conclude that $\|\mathbf{x}_i - \mathbf{x}_j\| > 0$, which means $\mathbf{a}_i$ and $\mathbf{a}_j$ are not in the same cluster.

where $\tau \in (0, 1]$ is a given constant in the algorithm, which is typically chosen to be 0.5.

Proof From Lemma 9, we know that $\mathrm{Prox}_{t\|\cdot\|_2}$ is strongly semismooth for any $t > 0$, together with the Moreau identity $\mathrm{Prox}_{tp}(x) + t\mathrm{Prox}_{p^*/t}(x/t) = x$, we know that

$$\nabla\phi(X) = X - A + \mathcal{B}^*(\mathrm{Prox}_{\sigma p^*}(\sigma\mathcal{B}(X) + \tilde{Z})),$$

is strongly semismooth. By (Zhao et al., 2010) [Proposition 3.3], we know that d^j obtained in Ssncg is a descent direction, which guarantees that the Algorithm Ssncg is well defined. From (Zhao et al., 2010) [Theorem 3.4, 3.5], we can get the desired convergence results. ■

5.6 Generating an initial point

In our implementation, we use the inexact alternating direction method of multipliers (iadmm) developed in (Chen et al., 2017) to generate an initial point to warm-start Ssnal. Note that with the global convergence result stated in Theorem 12, the performance of Ssnal does not sensitively depend on the initial points, but it is still helpful if we can choose a good one.

Algorithm 3 iadmm for (P)

Initialization: Choose $\sigma > 0$, $(X^0, U^0, Z^0) \in \mathbb{R}^{d\times n} \times \mathbb{R}^{d\times|\mathcal{E}|} \times \mathbb{R}^{d\times|\mathcal{E}|}$, and a summable nonnegative sequence $\{\epsilon_k\}$. For $k = 0, 1, \ldots,$

repeat

Step 1. Let $R^k = A + \sigma\mathcal{B}^*(U^k - \sigma^{-1}Z^k)$. Compute

$$\begin{aligned} X^{k+1} &\approx \arg\min_X\{\mathcal{L}_\sigma(X, U^k; Z^k)\}, \\ U^{k+1} &= \arg\min_U\{\mathcal{L}_\sigma(X^{k+1}, U; Z^k)\}, \end{aligned}$$

where X^{k+1} is an inexact solution satisfying the accuracy requirement that $\|(I_n + \sigma\mathcal{B}^*\mathcal{B})X^{k+1} - R^k\| \le \epsilon_k$.

Step 2. Compute

$$Z^{k+1} = Z^k + \tau\sigma_k(\mathcal{B}(X^{k+1}) - U^{k+1}),$$

where $\tau \in (0, \frac{1+\sqrt{5}}{2})$ is typically chosen to be 1.618.

until the stopping criterion is satisfied.

Observe that in Step 1, X^{k+1} is a computed solution for the following large linear system of equations:

$$(I_n + \sigma\mathcal{B}^*\mathcal{B})X = R^k \quad \Longleftrightarrow \quad (I_n + \sigma L_G)X^T = (R^k)^T.$$

To compute X^{k+1}, we can adopt a direct approach if the sparse Cholesky factorization of $I_n + \sigma L_G$ (which only needs to be done once) can be computed at a moderate cost; otherwise we can adopt an iterative approach by applying the conjugate gradient method to solve the above fairly well-conditioned linear system.

6. Numerical Experiments

In this section, we will first demonstrate that the sufficient conditions we derived for perfect recovery in Theorem 5 is practical via a simulated example. Then, we will show the superior performance of our proposed algorithm SSNAL on both simulated and real data sets, comparing to the popular algorithms such as ADMM and AMA which are proposed in (Chi and Lange, 2015). In particular, we will focus on the efficiency, scalability, and robustness of our algorithm for different values of γ. Also, we will show the performance of our algorithm on large data sets and unbalanced data. Previous numerical demonstration on the scalability and performance of (2) on large data sets is limited. The problem sizes of the instances tested in (Chi and Lange, 2015) and other related papers are at most several hundreds ($n \leq 500$ in (Chi and Lange, 2015), $n \leq 600$ in (Panahi et al., 2017)), which are not large enough to conclusively demonstrate the scalability of the algorithms. In this paper, we will present numerical results for n up to $\mathbf{200,000}$. We will also analyze the sensitivity of the efficiency of SSNAL and AMA, with respect to different choices of the hyper-parameters in (2), such as k (the number of nearest neighbors) and γ.

We focus on solving (2) with $p = 2$ since the rotational invariance of the 2-norm makes it a robust choice in practice. Also, this case is more challenging than $p = 1$ or $p = \infty$.[3] As the results reported in (Chi and Lange, 2015) have been regarded as the benchmark for the convex clustering model (2), we will compare our algorithm with the open source software CVXCLUSTR[4] in (Chi and Lange, 2015), which is an R package with key functions written in C. We write our code in MATLAB without any dedicated C functions. All our computational results are obtained from a desktop having 16 cores with 32 Intel Xeon E5-2650 processors at 2.6 GHz and 64 GB memory.

In our implementation, we stop our algorithm based on the following relative KKT residual:

$$\max\{\eta_P, \eta_D, \eta\} \leq \epsilon,$$

where

$$\eta_P = \frac{\|\mathcal{B}X - U\|}{1 + \|U\|},\ \eta_D = \frac{\sum_{(i,j)\in\mathcal{E}} \max\{0, \|Z^{l(i,j)}\|_2 - \gamma w_{ij}\}}{1 + \|A\|},$$

$$\eta = \frac{\|\mathcal{B}^*(Z) + X - A\| + \|U - \mathrm{Prox}_p(U + Z)\|}{1 + \|A\| + \|U\|},$$

and $\epsilon > 0$ is a given tolerance. In our experiments, we set $\epsilon = 10^{-6}$ unless specified otherwise. Since the numerical results reported in (Chi and Lange, 2015) have demonstrated the superior performance of AMA over ADMM, we will mainly compare our proposed algorithm with AMA. We note that CVXCLUSTR does not use the relative KKT residual as its stopping criterion but used the duality gap in AMA and $\max\{\eta_P, \eta_D\} \leq \epsilon$ in ADMM. To make a fair comparison, we first solve (2) using SSNAL with a given tolerance ϵ, and denote the primal objective value obtained as P_{Ssnal}. Then, we run AMA in CVXCLUSTR and stop it as soon as the computed primal objective function value (P_{AMA}) is close enough

3. Our algorithm can be generalized to solve (2) with $p = 1$ and $p = \infty$ without much difficulty.
4. https://cran.r-project.org/web/packages/cvxclustr/index.html

to P_{Ssnal}, i.e.,

$$P_{\text{AMA}} - P_{\text{Ssnal}} \leq 10^{-6} P_{\text{Ssnal}}. \tag{25}$$

We note that since (2) is an unconstrained problem, the quality of the computed solutions can directly be compared based on the objective function values. We also stop AMA if the maximum of 10^5 iterations is reached.

When we generate the clustering path for the first parameter value of γ, we first run the iADMM introduced in Algorithm 3 for 100 iterations to generate an initial point, then we use Ssnal to solve (2). After that, we use the previously computed optimal solution for the γ as the initial point to warm-start Ssnal for solving the problem corresponding to the next γ. The same strategy is used in cvxclustr.

6.1 Numerical Verification of Theorem 5

In this section, we demonstrate that the theoretical results obtained in Theorem 5 are practically meaningful by conducting numerical experiments on a simulated data set with five clusters. We generate the five clusters randomly via a 2D Gaussian kernel. Each of the cluster has 100 data points, as shown in Figure 1.

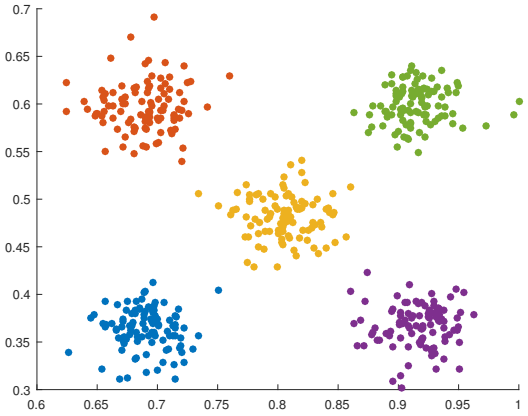

Figure 1: Visualization of the generated data.

Since we know the cluster assignment for each data point, we can construct the corresponding centroid problem given in (12). Then, we can solve the weighted convex clustering model (2) and the corresponding centroid problem (12) separately to compare the results. In our experiments, we choose the weight w_{ij} as follows

$$w_{ij} = \begin{cases} \exp(-0.5\|\mathbf{a}_i - \mathbf{a}_j\|^2) & \text{if } (i,j) \in \mathcal{E}, \\ 0 & \text{otherwise,} \end{cases}$$

where $\mathcal{E} = \cup_{i=1}^{n}\{(i,j) \mid \mathbf{a}_j \text{ is among } \mathbf{a}_i\text{'s 30-nearest neighbors}, i < j \leq n\} \cup_{\alpha=1}^{5} \{(i,j) \mid i,j \in I_\alpha, i < j\}$.

First, we solve (2) and (12) separately to find their optimal solutions, denoted as $X^* = [\mathbf{x}_1^*, \mathbf{x}_2^*, \ldots, \mathbf{x}_n^*]$ and $\bar{X} = [\bar{\mathbf{x}}^{(1)}, \bar{\mathbf{x}}^{(2)}, \ldots, \bar{\mathbf{x}}^{(K)}]$, respectively. Then, we can construct the new solution $\hat{X}$ for (2) based on $\bar{X}$ as

$$\hat{\mathbf{x}}_i = \bar{\mathbf{x}}^{(\alpha)} \;\; \forall\, i \in I_\alpha, \quad \alpha = 1, \ldots, 5.$$

We also compute the theoretical lower bound $\gamma_{\min}$ and upper bound $\gamma_{\max}$ based on the formula given in Theorem 5, and they are given by

$$\gamma_{\min} = 1.56 \times 10^{-3}, \quad \gamma_{\max} = 0.485.$$

Based on the computed results shown in the left panel of Figure 2, we can observe the phenomenon that for very small γ, X^* and $\hat{X}$ are different. However, when γ becomes larger, X^* and $\hat{X}$ coincide with each other in that $\|X^* - \hat{X}\|$ is almost 0 (up to the accuracy level we solve the problems (2) and (12)). In fact, we see that for γ larger than the theoretical lower bound $\gamma_{\min}$ but less than $\gamma_{\max}$, we have perfect recovery of the clusters by solving (2), and when γ is slightly smaller than $\gamma_{\min}$, we lose the perfect recovery property.

Furthermore, from our results in Theorem 5, we know that when γ is smaller than $\gamma_{\max}$ but larger than $\gamma_{\min}$, we should recover the correct number of clusters. This is indeed observed in the result shown in the right panel of Figure 2 where we track the number of clusters for different values of γ. Moreover, when γ is about two times larger than $\gamma_{\max}$, we get a coarsening of the clusters. The results shown above demonstrate that the theoretical results we have established in Theorem 5 are meaningful in practice.

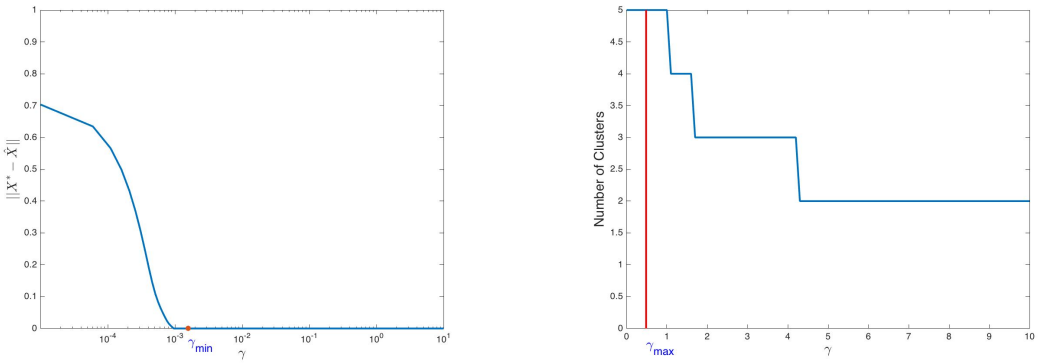

Figure 2: Left panel: $\|X^* - \hat{X}\|$ vs γ; Right panel: number of clusters vs γ.

Remark 14 *If we consider the full convex clustering model (1) with equal weights for the above example, then based on the results in part (i) of Corollary 7, we can get the following lower bound and upper bound for γ:*

$$\hat{\gamma}_{\min} = 1.53 \times 10^{-3}, \quad \hat{\gamma}_{\max} = 2 \times 10^{-4},$$

which is actually not feasible. This interesting result also demonstrates the importance of the weighted convex clustering model and the new theoretical bounds obtained in this paper.

Next, we show the numerical performance of our proposed optimization algorithm for solving (2).

6.2 Simulated data

In this section, we show the performance of our algorithm Ssnal on three simulated data sets: Two Half-Moon, Unbalanced Gaussian (Rezaei and Fränti, 2016) and semi-spherical

shells data. We compare our proposed SSNAL with the AMA in (Chi and Lange, 2015) on different problem scales. The numerical results in Table 2 and Table 3 show the superior performance of SSNAL over AMA. We also visualize some selected recovery results for Two Half-moon and Unbalanced Gaussian in Figure 3 .

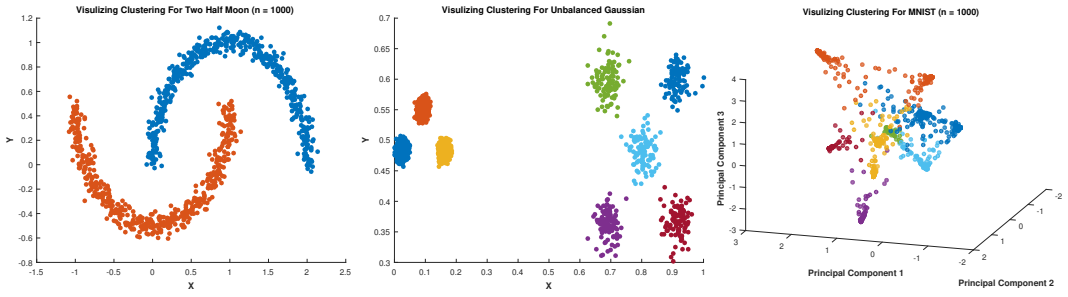

Figure 3: Selected recovery results by model (2) with 2-norm. Left: Two Half-Moon data with $n = 1000$, $k = 20$, $\gamma = 5$. Middle: Unbalanced Gaussian data with $n = 6500$, $k = 10$, $\gamma = 1$. Right: a subset of MNIST with $n = 1000$, $k = 10$ and $\gamma = 1$.

Two Half-Moon data

The simulated data of two interlocking half-moons in $\mathbb{R}^2$ is one of the most popular test examples in clustering. Here we compare the computational time between our proposed SSNAL and AMA on this data set with different problem scales. We note that AMA could not satisfy the stopping criteria (25) within 100000 iterations when n is large. In the experiments, we choose $k = 10$, $\phi = 0.5$ (for the weights w_{ij}) and $\gamma \in [0.2 : 0.2 : 10]$ (in MATLAB notation) to generate the clustering path. After generating the clustering path with SSNAL, we repeat the experiments using the same pre-stored primal objective values and stop the AMA using the criterion (25). We report the average time for solving each problem (50 in total) in Table 2. Observe that the SSNAL can be more than 50 times faster than AMA.

We also compare the recovery performance between the convex clustering model (2) and K-means (3). We choose the popular rand index (Hubert and Arabie, 1985) as the metric to evaluate the performance of these two clustering algorithms. Roughly speaking, the rand index is a quantitative measurement for the similarity of two partitions of a given index set. In the left panel of Figure 4, we can see that comparing to the K-means model, the convex clustering model is able to achieve a much better Rand Index, even when the number of clusters is not correctly identified.

Unbalanced Gaussian and semi-spherical shells data

Next, we show the performance of SSNAL and AMA on the Unbalanced Gaussian data points in $\mathbb{R}^2$ (Rezaei and Fränti, 2016). In this experiment, we solve (2) with $k = 10$, $\phi = 0.5$ and $\gamma \in [0.2 : 0.2 : 2]$. For this data set, we have scaled it so that each entry is in the interval

Table 2: Computation time (in seconds) comparison on the Two Half-Moon data. (— means that the maximum number of 100,000 iterations is reached)

n	200	500	1000	2000	5000	10000
AMA	0.41	4.43	28.27	78.36	—	—
Ssnal	**0.11**	**0.19**	**0.49**	**0.91**	**3.82**	**9.15**

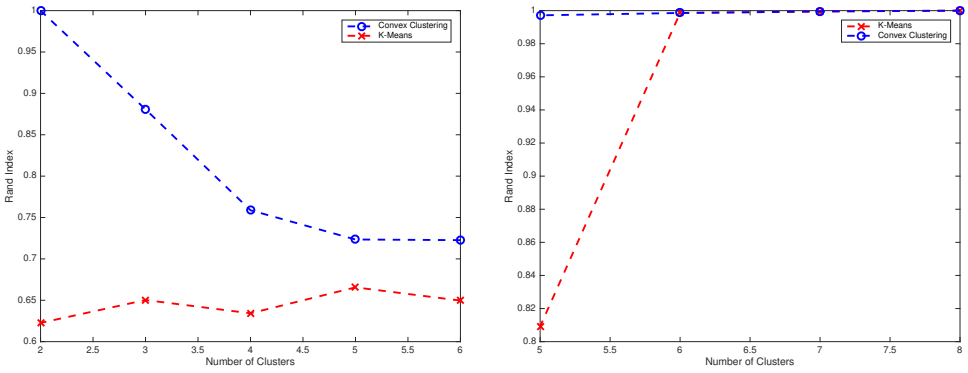

Figure 4: Clustering performance (in terms of the Rand Index) of the convex clustering and K-means models on the Two Half Moon data set (left panel) and the Unbalanced Gaussian data set (right panel).

$[0, 1]$. We can see from Figure 3 that the convex clustering model (2) can recover the cluster assignments perfectly with well chosen parameters.

In the experiments, we find that AMA has difficulties in reaching the stopping criterion (25). We summarize some selected results in Table 3, wherein we report the computation times and iteration counts for both AMA and Ssncg. Note that we report the number of Ssncg iterations because each of these iterations constitute the main cost for Ssnal. In the right panel of Figure 4, we show the recovery performance of the convex clustering model and K-means on this data set.

Table 3: Numerical results on the unbalanced Gaussian data.

γ	0.2	0.4	0.6	0.8	1.0
t_{AMA}	264.54	256.21	260.06	262.16	263.27
t_{Ssnal}	**1.15**	**0.57**	**0.65**	**0.64**	**0.83**
Iter_{AMA}	100000	97560	97333	100000	100000
$\text{Iter}_{\text{Ssncg}}$	**23**	**21**	**24**	**24**	**27**

In order to test the performance of our Ssnal on large data set, we also generate a data set with $\mathbf{200,000}$ points in $\mathbb{R}^3$ such that 50% of the points are uniformly distributed in a semi-spherical shell whose inner and outer surfaces have radius equals to 1.0 and 1.4,

respectively. The other 50% of the points are uniformly distributed in a concentric semi-spherical shell whose inner and outer surfaces have radius equals to 1.6 and 2.0, respectively. Figure 5 depicts the recovery result when we use only 6,000 points. For the data set with $\mathbf{n = 200,000}$, our algorithm takes only **374 seconds** to solve the model (2) when we choose $\gamma = 50$, $\phi = 0.5$ and $k = 10$. In solving the problem, our algorithm used 32 Ssncg iterations and the average number of CG steps needed to solve the large linear system (23) is 79.3 only. Thus, we can see that our algorithm can be very efficient in solving the convex clustering model (2) even when the data set is large. Note that we did not run AMA as it will take too much time to solve the problem.

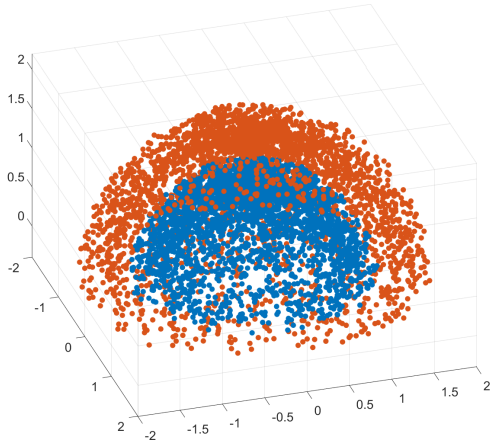

Figure 5: Clustering result by model (2) for a semi-spherical shells data set with $6,000$ points.

6.3 Real data

In this section, we compare the performance of our proposed Ssnal with AMA on some real data sets, namely, MNIST, Fisher Iris, WINE, Yale Face B(10 Train subset). For real data sets, a preprocessing step is sometimes necessary to transform the data to the one whose features are meaningful for clustering. Thus, for a subset of MNIST (we selected a subset because AMA cannot handle the whole data set), we first apply the preprocessing method described in (Mixon et al., 2017). Then we apply the model (2) on the preprocessed data. The comparison results between Ssnal and AMA on the real data sets are presented in Table 4. One can observe that Ssnal can be much more efficient than AMA.

6.4 Sensitivity with different γ

In order to generate a clustering path for a given data set, we need to solve (2) for a sequence of $\gamma > 0$. So the stability of the performance of the optimization algorithm with different γ is very important. In our experiments, we have found that the performance of AMA is rather sensitive to the value of γ in that the time taken to solve problems with different values of γ can vary widely. However, Ssnal is much more stable. In Figure 6, we show

Table 4: Computation time comparison on real data. (*) means that the maximum of 100000 iterations is reached for all instances.

Data set	d	n	AMA(s)	Ssnal(s)
MNIST	10	1,000	79.48	**1.47**
MNIST	10	10,000	1753.8*	**69.3**
Fisher Iris	4	150	0.58	**0.16**
WINE	13	178	2.62	**0.19**
Yale Face B	1024	760	211.36	**35.13**

the comparison between Ssnal and AMA on both the Two Half-Moon and MNIST data sets with $\gamma \in [0.2 : 0.2 : 10]$.

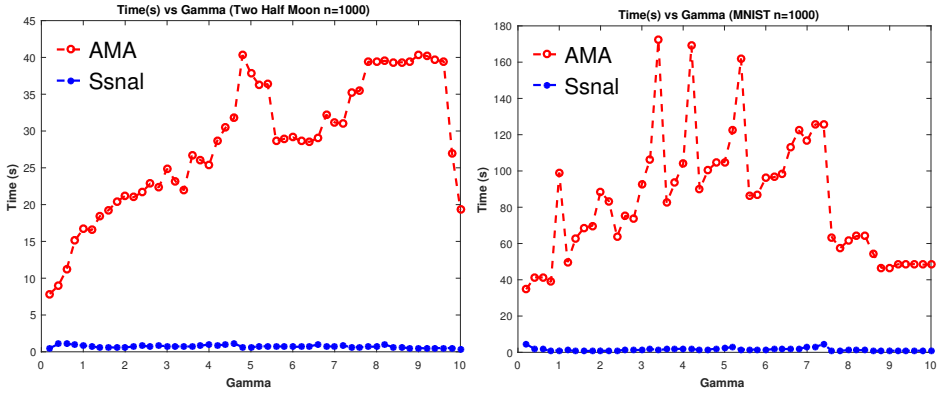

Figure 6: Time comparison between Ssnal and AMA on both Two Half-Moon and MNIST data with $\gamma \in [0.2 : 0.2 : 10]$.

6.5 Scalability of our proposed algorithm

In this section, we demonstrate the scalability of our algorithm Ssnal. Before we show the numerical results, we give some insights as to why our algorithm could be scalable. Recall that the most computationally expensive step in our framework is in using the semismooth Newton-CG method to solve (21). However, if we look inside the algorithm, we can see that the key step is to use the CG method to solve (23) efficiently to get the Newton direction. According to our complexity analysis in Section 5.4, the computational cost for one step of the CG method is $O(d|\mathcal{E}| + d|\widehat{\mathcal{E}}|)$. By the specific choice of $\mathcal{E}$, $|\mathcal{E}|$ and $|\widehat{\mathcal{E}}|$ should only grow slowly with n. The low computational cost for the matrix-vector product in our CG method, the rapid convergence of the CG method, and the fast convergence of the Ssncg are the key reasons behind why our algorithm can be scalable and efficient.

In our experiments, we set $\phi = 0.5$, $k = 10$ (the number of nearest neighbors). Then we solve (2) with $\gamma \in [0.4 : 0.4 : 20]$. After generating the clustering path, we compute the average time for solving a single instance of (2) for each problem scale. Another factor

related to the scalability is the number of neighbors k used to generate $\mathcal{E}$ in (2). So, we also show the performance of Ssnal with different values of k. For each $k \in [5:5:50]$, we generate the clustering path for the Two Half-Moon data with $n = 2000$. Then we report the average time for solving a single instance of (2) for each k. We summarize our numerical results in Figure 7. We can observe that the computation time grows almost linearly with n and k.

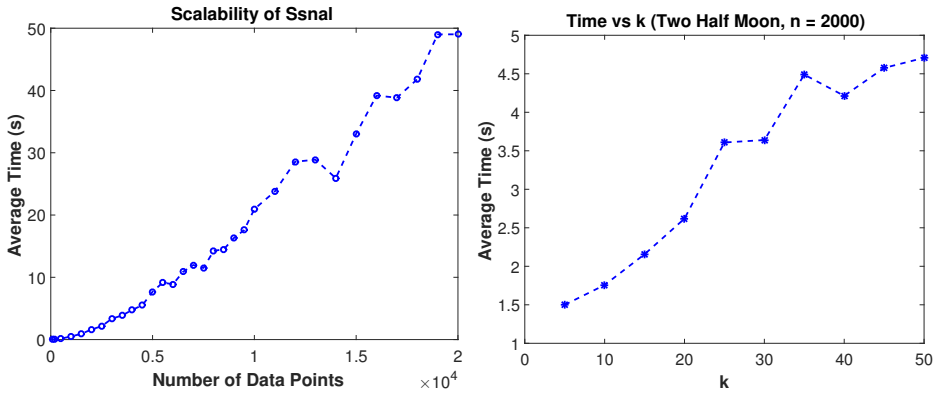

Figure 7: Numerical results to demonstrate the scalability of our proposed algorithm Ssnal with respect to n and k.

Comparing to the numerical results reported in Chi and Lange (2015) and Panahi et al. (2017) with $n \leq 500$ and $n \leq 600$, respectively, in our experiments, we apply our algorithm on the Half-Moon data with n ranging from 100 to 20000. Together with the semi-spherical shells with $\mathbf{200,000}$ data points, our results have convincingly demonstrated the scalability of Ssnal.

7. Conclusion

In this paper, we established the theoretical recovery guarantee for the general weighted convex clustering model, which includes many popular setting as special cases. The theoretical results we obtained serve to provide a more solid foundation for the convex clustering model. We have also proposed a highly efficient and scalable semismooth Newton based augmented Lagrangian method to solve the convex clustering model (2). To the best of our knowledge, this is the first optimization algorithm for convex clustering model which uses the second-order generalized Hessian information. Extensive numerical results shown in the paper have demonstrated the scalability and superior performance of our proposed algorithm Ssnal comparing to the state-of-the-art first-order methods such as AMA and ADMM. The convergence results for our algorithm are also provided.

As a possible future work, we plan to design a distributed and parallel version of Ssnal with the aim to handle huge scale data sets. From the modeling perspective, we will also work on generalizing our algorithm to handle kernel based convex clustering models. More robust strategy for cluster assignments will be also a possible future topic for us to consider.

Acknowledgments

We thank the three referees for the helpful suggestions to improve this paper. Defeng Sun is supported in part by Hong Kong Research Grant Council under grant PolyU 153014/18P. Kim-Chuan Toh is supported by the Academic Research Fund of the Ministry of Education, Singapore, under grant R-146-000-257-112.

References

W. N. Anderson and T. D. Morley. Eigenvalues of the Laplacian of a graph. *Linear and Multilinear Algebra*, 18:141–145, 1985.

P. Awasthi, A. S. Bandeira, M. Charikar, R. Krishnaswamy, S. Villar, and R. Ward. Relax, no need to round: Integrality of clustering formulations. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science*, pages 191–200, 2015.

H. H. Bauschke and P. L. Combettes. *Convex analysis and monotone operator theory in Hilbert spaces*, volume 408. Springer, 2011.

L. Chen, D. F. Sun, and K. C. Toh. An efficient inexact symmetric Gauss–Seidel based majorized ADMM for high-dimensional convex composite conic programming. *Mathematical Programming*, 161:237–270, 2017.

E. C. Chi and K. Lange. Splitting methods for convex clustering. *J. Computational and Graphical Statistics*, 24(4):994–1013, 2015.

E. C. Chi, B. R. Gaines, W. W. Sun, H. Zhou, and J. Yang. Provable convex co-clustering of tensors. *arXiv preprint arXiv:1803.06518*, 2018.

F. Clarke. *Optimization and Nonsmooth Analysis*. John Wiley and Sons, New York, 1983.

J. B. Hiriart-Urruty, J. J. Strodiot, and V. H. Nguyen. Generalized Hessian matrix and second-order optimality conditions for problems with $C^{1,1}$ data. *Appl. Math. Optim.*, 11: 43–56, 1984.

T. D. Hocking, A. Joulin, F. Bach, and J. P. Vert. Clusterpath an algorithm for clustering using convex fusion penalties. In *28th International Conference on Machine Learning*, 2011.

L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, 1985.

T. Jiang and S. Vavasis. On identifying clusters from sum-of-norms clustering computation. *arXiv preprint arXiv:2006.11355*, 2020.

B. Kummer. Newton's method for non-differentiable functions. *Advances in Mathematical Optimization*, 45:114–125, 1988.

X. D. Li, D. F. Sun, and K. C. Toh. A highly efficient semismooth Newton augmented Lagrangian method for solving lasso problems. *SIAM J. Optimization*, 28:433–458, 2018.

F. Lindsten, H. Ohlsson, and L. Ljung. Clustering using sum-of-norms regularization: With application to particle filter output computation. In *Statistical Signal Processing Workshop (SSP)*, pages 201–204. IEEE, 2011.

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

D. G Mixon, S. Villar, and R. Ward. Clustering subgaussian mixtures by semidefinite programming. *Information and Inference: A Journal of the IMA*, 6(4):389–415, 2017.

J. Nocedal and S. Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

A. Panahi, D. Dubhashi, F. D. Johansson, and C. Bhattacharyya. Clustering by sum of norms: Stochastic incremental algorithm, convergence and cluster recovery. In *34th International Conference on Machine Learning*, volume 70, pages 2769–2777. PMLR, 2017.

K. Pelckmans, J. De Brabanter, J. Suykens, and B. De Moor. Convex clustering shrinkage. In *PASCAL Workshop on Statistics and Optimization of Clustering Workshop*, 2005.

J. Peng and Y. Wei. Approximating k-means-type clustering via semidefinite programming. *SIAM Journal on Optimization*, 18(1):186–205, 2007.

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

P. Radchenko and G. Mukherjee. Convex clustering via l_1 fusion penalization. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(5):1527–1546, 2017.

M. Rezaei and P. Fränti. Set-matching methods for external cluster validity. *IEEE Trans. on Knowledge and Data Engineering*, 28(8):2173–2186, 2016.

R. T. Rockafellar. Augmented Lagrangians and applications of the proximal point algorithm in convex programming. *Mathematics of Operations Research*, 1(2):97–116, 1976a.

R. T. Rockafellar. Monotone operators and the proximal point algorithm. *SIAM J. Control and Optimization*, 14(5):877–898, 1976b.

D. F. Sun, K. C. Toh, Y. C. Yuan, and X. Y. Zhao. SDPNAL+: a Matlab software for semidefinite programming with bound constraints (version 1.0). *Optimization Methods and Software*, 35(1):87–115, 2020.

K. M. Tan and D. Witten. Statistical properties of convex clustering. *Electronic J. Statistics*, 9(2):2324, 2015.

L. Q. Yang, D. F. Sun, and K. C. Toh. SDPNAL+: a majorized semismooth Newton-CG augmented Lagrangian method for semidefinite programming with nonnegative constraints. *Mathematical Programming Computation*, 7(3):331–366, 2015.

Y. C. Yuan, D. F. Sun, and K. C. Toh. An efficient semismooth Newton based algorithm for convex clustering. In *35th International Conference on Machine Learning*. PMLR 80, 2018.

Y. J. Zhang, N. Zhang, D. F. Sun, and K. C. Toh. An efficient Hessian based algorithm for solving large-scale sparse group lasso problems. *Mathematical Programming*, 179(1): 223–263, 2020.

X. Y. Zhao, D. F. Sun, and K. C. Toh. A Newton-CG augmented Lagrangian method for semidefinite programming. *SIAM Journal on Optimization*, 20(4):1737–1765, 2010.

C. Zhu, H. Xu, C. L. Leng, and S. C. Yan. Convex optimization procedure for clustering: Theoretical revisit. In *Advances in Neural Information Processing Systems 27*, pages 1619–1627, 2014.