Fast Multiplier Methods to Optimize Non-exhaustive, Overlapping Clustering

Yangyang Hou* Joyce Jiyoung Whang† David F. Gleich* Inderjit S. Dhillon‡

Abstract

Clustering is one of the most fundamental and important tasks in data mining. Traditional clustering algorithms, such as K-means, assign every data point to exactly one cluster. However, in real-world datasets, the clusters may overlap with each other. Furthermore, often, there are outliers that should not belong to any cluster. We recently proposed the NEO-K-Means (Non-Exhaustive, Overlapping K-Means) objective as a way to address both issues in an integrated fashion. Optimizing this discrete objective is NP-hard, and even though there is a convex relaxation of the objective, straightforward convex optimization approaches are too expensive for large datasets. A practical alternative is to use a low-rank factorization of the solution matrix in the convex formulation. The resulting optimization problem is non-convex, and we can locally optimize the objective function using an augmented Lagrangian method. In this paper, we consider two fast multiplier methods to accelerate the convergence of an augmented Lagrangian scheme: a proximal method of multipliers and an alternating direction method of multipliers (ADMM). For the proximal augmented Lagrangian or proximal method of multipliers, we show a convergence result for the non-convex case with bound-constrained subproblems. These methods are up to 13 times faster—with no change in quality—compared with a standard augmented Lagrangian method on problems with over 10,000 variables and bring runtimes down from over an hour to around 5 minutes.

1 Introduction

Traditional clustering algorithms, such as k-means, produce a disjoint, exhaustive clustering, i.e., the clusters are pairwise disjoint, and every data point is assigned to some cluster. However, in real-world datasets, the clusters may overlap with each other, and there are often outliers that should not belong to any cluster. We recently proposed the NEO-K-Means (Non-Exhaustive, Overlapping K-Means) objective as a generalization of the k-means clustering objective that allows us to simultaneously identify overlapping clusters as well as outliers [24]. Hence, it produces a non-exhaustive clustering. Curiously, both operations appear to be necessary because the outliers induce non-obvious effects when the clusters are allowed to overlap. It has been shown that the NEO-K-Means objective is effective in finding ground-truth clusters in data clustering problems. Furthermore, by considering a weighted and kernelized version of the NEO-K-Means objective, we can also tackle the problem of finding overlapping communities in social and information networks.

There are currently two practical methods to optimize the non-convex NEO-K-Means objective for large problems: the iterative NEO-K-Means algorithm [24] that generalizes Lloyd's algorithm [14] and an augmented Lagrangian algorithm to optimize a non-convex, low-rank semidefinite programming (SDP) relaxation of the NEO-K-Means objective [11]. The iterative algorithm is fast, but it tends to get stuck into regions where the more sophisticated optimization methods can make further progress. The augmented Lagrangian method for the non-convex objective, when started from the output of the iterative algorithm, is able to make further progress on optimizing the objective function. In addition, the augmented Lagrangian method tends to achieve better F_1 performance on identifying ground-truth clusters and produce better overlapping communities in real-world networks than the simple iterative algorithm [11]. In this paper, our goal is to improve upon the augmented Lagrangian method to optimize the low-rank SDP for the NEO-K-Means objective more quickly.

The optimization problem that results from the low-rank strategy on the convex SDP is a non-convex, quadratically constrained, bound-constrained problem. We consider two *multiplier methods* for this problem. The first method adds a proximal regularizer to the augmented Lagrangian method. This general strategy is called either the proximal augmented Lagrangian method (e.g., [12]) or the proximal method of multipliers [21]. The second method is an alternating direction method of multipliers (ADMM) strategy for our objective function. Both strategies, when specialized on the NEO-K-Means problem, have the potential to accelerate our solution process.

There is an extensive literature on both strategies for convex optimization [5, 10, 21] and there are a variety of convergence theories in the non-convex case [16, 19, 13]. However, we were unable to identify any existing convergence guarantees for these methods that mapped to our specific instantiations with bound-

*Department of Computer Science, Purdue University. Email: {*hou*13, *dgleich*}@purdue.edu

†Department of Computer Engineering, Sungkyunkwan University. Email: jjwhang@skku.edu

‡Department of Computer Science, The University of Texas at Austin. Email: inderjit@cs.utexas.edu

constrained subproblems. Towards that end, we specialize a general convergence result about the proximal augmented Lagrangian or proximal method of multipliers due to Pennanen [19] to our algorithm. The resulting theorem is a general convergence result about the proximal augmented Lagrangian method for non-convex problems with bound-constrained subproblems (Theorem 5.1). The proof involves adapting a few details from Pennanen to our case.

We evaluate the resulting methods on real-world problems where the existing augmented Lagrangian takes over an hour of computation time. The proximal augmented Lagrangian strategy tends to run about $3-6$ times faster, and the ADMM strategy tends to run about $4-13$ times faster bringing the runtimes of these methods down into range of 5 to 10 minutes. The iterative method, in contrast, runs in seconds – so there is still a considerable gap between the approaches. That said, the optimization based approaches have runtimes that are reasonable for a pipeline-style analysis and cases where the data collection itself is highly time-consuming as would be common in many datasets from the biological and physical sciences.

In summary:

- We propose two algorithms to optimize the non-convex problem for non-exhaustive, overlapping clustering: a proximal augmented Lagrangian method and an ADMM method.
- We specialize a general convergence result about the proximal method of multipliers for non-convex problems to the bound-constrained proximal augmented Lagrangian method to have a sound convergence theory.
- We show that these new methods reduce the runtime for problems where the classical augmented Lagrangian method takes over an hour to the range of 5 to 10 minutes with no change in quality.

The rest of the paper is organized as follows. In Section 2, we review the NEO-K-Means objective and its low-rank SDP formulation, and in Section 3, we formally describe the classical augmented Lagrangian method. In Section 4, we present our two multiplier methods: the proximal augmented Lagrangian method, and an ADMM for the NEO-K-Means low-rank SDP. For the proximal augmented Lagrangian method, we present the convergence analysis in Section 5. In Section 6, we discuss simplified ADMM variants. Finally, we present experimental results in Section 7, and discuss future work in Section 8.

2 The NEO-K-Means Objective

The goal of non-exhaustive, overlapping clustering is to find a set of cohesive clusters such that clusters are allowed to overlap with each other and outliers are not assigned to any cluster. That is, given a set of data points $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$, we find a set of clusters $\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_k$ such that $\mathcal{C}_1 \cup \mathcal{C}_2 \cup ... \cup \mathcal{C}_k \subseteq \mathcal{X}$ and $\mathcal{C}_i \cap \mathcal{C}_j \neq \emptyset$ for some $i \neq j$.

To find such clusters, we proposed the NEO-K-Means objective function in [24]. The NEO-K-Means objective is an intuitive variation of the classical k-means where two parameters α and β are introduced to control the amount of overlap and non-exhaustiveness, respectively. We also found that optimizing a weighted and kernelized NEO-K-Means objective is equivalent to optimizing normalized cuts for overlapping community detection [24].

Let us define an assignment matrix $\boldsymbol{U} = [u_{ij}]_{n \times k}$ such that $u_{ij} = 1$ if a data point $\mathbf{x}_i$ belongs to $\mathcal{C}_j$; and $u_{ij} = 0$ otherwise. Let $\mathbb{I}\{exp\}$ denote the indicator function such that $\mathbb{I}\{exp\} = 1$ if exp is true; 0 otherwise. Given a positive weight for each data point w_i, and a nonlinear mapping ϕ, the weighted kernel NEO-K-Means objective function is defined as follows:

$$\begin{aligned} \underset{\boldsymbol{U}}{\text{minimize}} \quad & \textstyle\sum_{c=1}^{k} \sum_{i=1}^{n} u_{ic} w_i \|\phi(\mathbf{x}_i) - \mathbf{m}_c\|^2 \\ & \text{where } \mathbf{m}_c = \frac{\sum_{i=1}^{n} u_{ic} w_i \phi(\mathbf{x}_i)}{\sum_{i=1}^{n} u_{ic} w_i} \\ \text{subject to} \quad & \text{trace}(\boldsymbol{U}^T \boldsymbol{U}) = (1+\alpha)n, \\ & \textstyle\sum_{i=1}^{n} \mathbb{I}\{(\boldsymbol{U}\mathbf{1})_i = 0\} \leq \beta n. \end{aligned} \tag{2.1}$$

This objective function implies that $(1+\alpha)n$ assignments are made while minimizing the sum of the squared distances between a data point and its cluster center. Also notice that at most βn data points are allowed to have no membership in any cluster. If $\alpha = 0$ and $\beta = 0$, then this objective is equivalent to the classical weighted kernel k-means objective. Some guidelines about how to select α and β have been described in [24]. To optimize the objective function (2.1), a simple iterative algorithm has also been proposed in [24]. However, the simple iterative algorithm tends to get stuck at a local optimum that can be far away from the global optimum, like the standard k-means algorithm [14].

The following optimization problem is a non-convex relaxation of the NEO-K-Means problem that was developed in our previous work [11]. We call it the low-rank SDP based on its derivation as a low-rank heuristic for solving large-scale SDPs. We introduce a bit of notation to state the problem. Let $\boldsymbol{K}$ be a standard kernel matrix ($K_{ij} = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$), let $\boldsymbol{W}$ denote a diagonal weight matrix such that $W_{ii} = w_i$ indicates the weight of data point i, and let $\mathbf{d}$ denote a vector of length n where $d_i = w_i K_{ii}$. In terms of the solution variables, let $\mathbf{f}$ be a length n vector where f_i is a real-valued count of the number of clusters data point i is assigned to, and let $\mathbf{g}$ be a length n vector where g_i is close to 0 if i

should not be assigned to any cluster and g_i is close to 1 if i should be assigned to a cluster. The solution matrix $\boldsymbol{Y}$ represents a relaxed, normalized assignment matrix where Y_{ij} indicates that data point i should be in cluster j with columns normalized by the cluster size. The low-rank SDP optimization problem for (2.1) is then

$$
\begin{array}{lll}
\underset{\boldsymbol{Y},\mathbf{f},\mathbf{g},\mathbf{s},r}{\text{minimize}} & \mathbf{f}^T\mathbf{d} - \text{trace}(\boldsymbol{Y}^T\boldsymbol{K}\boldsymbol{Y}) & \\
\text{subject to} & k = \text{trace}(\boldsymbol{Y}^T\boldsymbol{W}^{-1}\boldsymbol{Y}) & (a) \\
& 0 = \boldsymbol{Y}\boldsymbol{Y}^T\mathbf{e} - \boldsymbol{W}\mathbf{f} & (b) \\
& 0 = \mathbf{e}^T\mathbf{f} - (1+\alpha)n & (c) \\
& 0 = \mathbf{f} - \mathbf{g} - \mathbf{s} & (d) \\
& 0 = \mathbf{e}^T\mathbf{g} - (1-\beta)n - r & (e) \\
& Y_{i,j} \geq 0, \mathbf{s} \geq 0, r \geq 0 & \\
& 0 \leq \mathbf{f} \leq k\mathbf{e}, 0 \leq \mathbf{g} \leq 1 &
\end{array}
\tag{2.2}
$$

where $\mathbf{s}$ and r are slack variables to convert the inequality constraints into equality constraints. The objective function is derived following a standard kernelized conversion. Constraint (a) gives the normalization condition on the variable $\boldsymbol{Y}$ to normalize for cluster-size; constraint (b) requires that the number of assignments listed in $\mathbf{f}$ corresponds to the number in the solution matrix $\boldsymbol{Y}$; constraint (c) bounds the total number of assignments as $(1+\alpha)n$; constraint (d) is equivalent to $\mathbf{f} \geq \mathbf{g}$; and constraint (e) enforces the number of assigned data points to be at least $(1-\beta)n$; the remaining bound constraints enforce simple non-negativity and upper-bounds on the number of cluster assignments. We will discuss how to solve the low-rank SDP problem in the next section.

3 The Augmented Lagrangian Method for the NEO-K-Means Low-Rank SDP

To solve the low-rank SDP problem (2.2), the classical augmented Lagrangian method (ALM) has been used in [11]. The augmented Lagrangian technique is an iterative process where each iteration is done by minimizing an augmented Lagrangian problem that includes a current estimate of the Lagrange multipliers for the constraints as well as a quadratic penalty term that enforces the feasibility of the solution. We introduce it here because we will draw heavily on the notation for our subsequent results.

Let $\boldsymbol{\lambda} = [\lambda_1; \lambda_2; \lambda_3]$ denote the Lagrange multipliers for the three scalar constraints $(a), (c), (e)$. For the vector constraints (b) and (d), let $\boldsymbol{\mu}$ and $\boldsymbol{\gamma}$ denote the corresponding Lagrange multipliers, respectively. Let σ be a positive penalty parameter. Then, the augmented Lagrangian for (2.2) is:

$$
\begin{aligned}
&\mathcal{L}_A(Y, \mathbf{f}, \mathbf{g}, \mathbf{s}, r; \boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\gamma}, \sigma) = \\
&\underbrace{\mathbf{f}^T\mathbf{d} - \text{trace}(\boldsymbol{Y}^T\boldsymbol{K}\boldsymbol{Y})}_{\text{the objective}} \\
&- \lambda_1(\text{trace}(\boldsymbol{Y}^T\boldsymbol{W}^{-1}\boldsymbol{Y}) - k) \\
&\quad + \frac{\sigma}{2}(\text{trace}(\boldsymbol{Y}^T\boldsymbol{W}^{-1}\boldsymbol{Y}) - k)^2 \\
&- \boldsymbol{\mu}^T(\boldsymbol{Y}\boldsymbol{Y}^T\mathbf{e} - \boldsymbol{W}\mathbf{f}) \\
&\quad + \frac{\sigma}{2}(\boldsymbol{Y}\boldsymbol{Y}^T\mathbf{e} - \boldsymbol{W}\mathbf{f})^T(\boldsymbol{Y}\boldsymbol{Y}^T\mathbf{e} - \boldsymbol{W}\mathbf{f}) \\
&- \lambda_2(\mathbf{e}^T\mathbf{f} - (1+\alpha)n) \\
&\quad + \frac{\sigma}{2}(\mathbf{e}^T\mathbf{f} - (1+\alpha)n)^2 \\
&- \boldsymbol{\gamma}^T(\mathbf{f} - \mathbf{g} - \mathbf{s}) \\
&\quad + \frac{\sigma}{2}(\mathbf{f} - \mathbf{g} - \mathbf{s})^T(\mathbf{f} - \mathbf{g} - \mathbf{s}) \\
&- \lambda_3(\mathbf{e}^T\mathbf{g} - (1-\beta)n - r) \\
&\quad + \frac{\sigma}{2}(\mathbf{e}^T\mathbf{g} - (1-\beta)n - r)^2
\end{aligned}
$$

At each iteration of the augmented Lagrangian framework, the following subproblem is solved:

$$
\begin{array}{lll}
\text{minimize} & \mathcal{L}_{\mathcal{A}}(\boldsymbol{Y}, \mathbf{f}, \mathbf{g}, \mathbf{s}, r; \boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\gamma}, \sigma) \\
\text{subject to} & Y_{i,j} \geq 0, \mathbf{s} \geq 0, r \geq 0, \\
& 0 \leq \mathbf{f} \leq k\mathbf{e}, 0 \leq \mathbf{g} \leq 1.
\end{array}
\tag{3.3}
$$

To minimize the subproblem with respect to the variables $\boldsymbol{Y}$, $\mathbf{f}$, $\mathbf{g}$, $\mathbf{s}$, and r, we can use a limited-memory BFGS with bound constraints algorithm [6]. In [11], it has been shown that this technique produces reasonable solutions for the NEO-K-Means objective. In particular, when the clustering performance is evaluated on real-world datasets, this technique has been shown to be effective in finding the ground-truth clusters. Furthermore, by optimizing the weighted kernel NEO-K-Means, this technique is also able to find cohesive overlapping communities in real-world networks. The empirical success of the augmented Lagrangian framework motivates us to investigate developing faster solvers for the NEO-K-Means low-rank SDP problem, which will be discussed in the next section.

4 Fast Multiplier Methods for the NEO-K-Means Low-Rank SDP

There is a resurgence of interest in proximal point methods and alternating methods for convex and nearly convex objectives in machine learning due to their fast convergence rate. Here we propose two variants of the classical augmented Lagrangian approach on problem (2.2) that can utilize some of these techniques for improved speed.

4.1 Proximal Augmented Lagrangian (PALM).

The proximal augmented Lagrangian method differs

from the classical augmented Lagrangian method only in an additional proximal regularization term for primal updates. This can be considered as a type of simultaneous primal-dual proximal-point step that helps to regularize the subproblems solved at each step. This idea leads to the following iterates:

$$\mathbf{x}^{k+1} = \underset{\mathbf{x}\in\mathcal{C}}{\operatorname{argmin}} \mathcal{L}_{\mathcal{A}}(\mathbf{x}; \boldsymbol{\lambda}^k, \boldsymbol{\mu}^k, \boldsymbol{\gamma}^k, \sigma) + \frac{1}{2\tau}\|\mathbf{x} - \mathbf{x}^k\|^2$$

where $\mathbf{x}$ represents $[\mathbf{y}; \mathbf{f}; \mathbf{g}; \mathbf{s}; r]$ for simplicity with $\mathbf{y} = \boldsymbol{Y}(:)$ vectorized by column. Then we update the multipliers $\boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\gamma}$ as in the classical augmented Lagrangian. We may also need to update the penalty parameter σ and the proximal parameter τ respectively.

We use a limited-memory BFGS with bound constraints to solve the new subproblem with respect to the variable $\mathbf{x}$. If we let $\tau = \sigma$, this special case is called proximal method of multipliers, first introduced in [21]. The proximal method of multipliers has better theoretical convergence guarantees for convex optimization problems (compared with the augmented Lagrangian) [21]. In this non-convex setting, we believe it is likely to help to improve conditioning of the Hessian's in the subproblems and thus reduce the solution time for each subproblem. And this is indeed what we find.

4.2 Alternating Direction Method of Multipliers (ADMM). There are four sets of variables in problem (2.2) ($\boldsymbol{Y}$, $\mathbf{f}$, $\mathbf{g}$ and slack variables). We can use this structure to break the augmented Lagrangian function into smaller subproblems for each set of variables. Some of these subproblems are then easier to solve. For example, updating variable $\mathbf{f}$ alone is a simple convex problem, thus it is very efficient to have a globally optimal solution. The alternating direction method of multipliers approach of updating block variables $\boldsymbol{Y}$, $\mathbf{f}$, $\mathbf{g}$, $\mathbf{s}$ and r respectively, utilizes this property, which leads to the following iterates:

$$\begin{aligned}
\boldsymbol{Y}^{k+1} &= \underset{\boldsymbol{Y}}{\operatorname{argmin}} \mathcal{L}_{\mathcal{A}}(\boldsymbol{Y}, \mathbf{f}^k, \mathbf{g}^k, \mathbf{s}^k, r^k; \\
&\qquad \boldsymbol{\lambda}^k, \boldsymbol{\mu}^k, \boldsymbol{\gamma}^k, \sigma) \\
\mathbf{f}^{k+1} &= \underset{\mathbf{f}}{\operatorname{argmin}} \mathcal{L}_{\mathcal{A}}(\boldsymbol{Y}^{k+1}, \mathbf{f}, \mathbf{g}^k, \mathbf{s}^k, r^k; \\
&\qquad \boldsymbol{\lambda}^k, \boldsymbol{\mu}^k, \boldsymbol{\gamma}^k, \sigma) \\
\mathbf{g}^{k+1} &= \underset{\mathbf{g}}{\operatorname{argmin}} \mathcal{L}_{\mathcal{A}}(\boldsymbol{Y}^{k+1}, \mathbf{f}^{k+1}, \mathbf{g}, \mathbf{s}^k, r^k; \\
&\qquad \boldsymbol{\lambda}^k, \boldsymbol{\mu}^k, \boldsymbol{\gamma}^k, \sigma) \\
\mathbf{s}^{k+1} &= \underset{\mathbf{s}}{\operatorname{argmin}} \mathcal{L}_{\mathcal{A}}(\boldsymbol{Y}^{k+1}, \mathbf{f}^{k+1}, \mathbf{g}^{k+1}, \mathbf{s}, r^k; \\
&\qquad \boldsymbol{\lambda}^k, \boldsymbol{\mu}^k, \boldsymbol{\gamma}^k, \sigma) \\
r^{k+1} &= \underset{r}{\operatorname{argmin}} \mathcal{L}_{\mathcal{A}}(\boldsymbol{Y}^{k+1}, \mathbf{f}^{k+1}, \mathbf{g}^{k+1}, \mathbf{s}^{k+1}, r; \\
&\qquad \boldsymbol{\lambda}^k, \boldsymbol{\mu}^k, \boldsymbol{\gamma}^k, \sigma)
\end{aligned}$$

then the multipliers $\boldsymbol{\lambda}$, $\boldsymbol{\mu}$, $\boldsymbol{\gamma}$ and the penalty parameter σ are updated accordingly.

We expect that this strategy will aid convergence because it decouples the update of $\boldsymbol{Y}$ from the update of $\mathbf{f}$. In the problem with all variables, the interaction of these terms has the strongest non-convex interaction. We now detail how we solve each of the subproblems.

Update $\boldsymbol{Y}$. We use a limited-memory BFGS with bound constraints to solve the subproblem with respect to the variables $\boldsymbol{Y}$ since it is non-convex.

Update f and g. The update for $\mathbf{f}$ and $\mathbf{g}$ respectively both have the following general form:

$$\begin{array}{ll} \underset{\mathbf{x}}{\text{minimize}} & f(\mathbf{x}) = \mathbf{x}^T\mathbf{a} + \dfrac{\sigma}{2}\mathbf{x}^T\boldsymbol{D}\mathbf{x} + \dfrac{\sigma}{2}(\mathbf{e}^T\mathbf{x})^2 \\ \text{subject to} & 0 \le \mathbf{x} \le b\mathbf{e} \end{array} \tag{4.4}$$

where $\mathbf{e}$ is the vector of all 1s and $\boldsymbol{D}$ is a positive diagonal matrix. To solve this, we use ideas similar to [18, S6.2.5]. Let $\tau = \mathbf{e}^T\mathbf{x}$, thus $0 \le \tau \le bn$. We solve this problem by finding roots of the following function $F(\tau)$:

$$F(\tau) = \tau - \mathbf{e}^T P[-\tfrac{1}{\sigma}\boldsymbol{D}^{-1}(\mathbf{a} + \sigma\tau\mathbf{e}); 0, b]$$

where the function $P[\mathbf{x}; 0, b]$ projects the point $\mathbf{x}$ onto the rectangular box $[0, b]$. (To find these roots, bisection suffices because $F(0) \le 0$ and $F(bn) \ge 0$.) This is a globally optimal solution by the following lemma.

LEMMA 4.1. *$\mathbf{x}^* = P[-\frac{1}{\sigma}\boldsymbol{D}^{-1}(\mathbf{a} + \sigma\tau^*\mathbf{e}); 0, b]$, where τ^* is the root of $F(\tau)$, satisfies the first order KKT conditions: $\mathbf{x}^* - P[\mathbf{x}^* - \nabla f(\mathbf{x}^*); 0, b] = 0$ (this form is given in equation 17.51 of [17]).*

Proof. We have three cases: $x_i^* = 0$; $x_i^* = b$; and $0 < x_i^* < b$ for any i.

For $x_i^* = 0$, which means $a_i + \sigma\tau \ge 0$, we have

$$\begin{aligned} & x_i^* - P[x_i^* - (a_i + \sigma d_i x_i^* + \sigma\tau); 0, b] \\ = \;& -P[-a_i - \sigma\tau; 0, b] = 0. \end{aligned}$$

For $x_i^* = b$, which means $-(a_i + \sigma\tau)/(\sigma d_i) \ge b$, we have

$$\begin{aligned} & x_i^* - P[x_i^* - (a_i + \sigma d_i x_i^* + \sigma\tau); 0, b] \\ = \;& b - P[b - (a_i + \sigma d_i b + \sigma\tau); 0, b] = b - b = 0. \end{aligned}$$

For $0 < x_i^* < b$, which means $x_i^* = -(a_i + \sigma\tau)/(\sigma d_i)$, we have

$$\begin{aligned} & x_i^* - P[x_i^* - (a_i + \sigma d_i x_i^* + \sigma\tau); 0, b] \\ = \;& x_i^* - P[x_i^*; 0, b] = x_i^* - x_i^* = 0. \end{aligned}$$

■

Update s and r. These updates just require solving one variable quadratic optimization with simple bound constraints; the result is a simple update procedure.

5 Convergence Analysis of the Proximal Augmented Lagrangian

We use both the proximal augmented Lagrangian and the ADMM strategy on the problem without any convexity. For these cases, local convergence is the best we can achieve. We now establish a general convergence result for the proximal augmented Lagrangian with bound constraints. We observed empirical convergence of the ADMM method, but currently lack any theoretical guarantees.

From Pennanen [19], we know that the proximal method of multipliers is locally convergent for a general class of problems with sufficient assumptions. We will show that our proximal method of multipliers algorithm applied to (2.2) can be handled by their approach and we extend their analysis to our case. Because we are imitating the analysis from Pennanen for a specific new problem, we decided to explicitly mimic the original language to highlight the changes in the derivation. Thus, there is a high degree of textual overlap between the following results and [19].

First, we state some notation and one useful fact. The indication function $\delta_{\mathcal{C}}$ of a set $\mathcal{C}$ in Hilbert Space $\mathcal{H}$ has value 0 for $x \in \mathcal{C}$ and $+\infty$ otherwise. The subdifferential of $\delta_{\mathcal{C}}$ is the normal cone operator of $\mathcal{C}$: $N_{\mathcal{C}}(\mathbf{x}) = \{\mathbf{v} \in \mathcal{H} | \langle \mathbf{v}, \mathbf{y} - \mathbf{x} \rangle \leq 0, \forall \mathbf{y} \in \mathcal{C}\}$.

PROPOSITION 5.1. *Let $\bar{\mathbf{x}}$ be a solution to problem of minimizing $f(\mathbf{x})$ on $\mathcal{C}$ and suppose f is differentiable at $\bar{\mathbf{x}}$, then*

$$\nabla f(\bar{\mathbf{x}}) + N_{\mathcal{C}}(\bar{\mathbf{x}}) \ni 0.$$

Proof. We need to show that $\nabla f(\bar{\mathbf{x}}) + N_{\mathcal{C}}(\bar{\mathbf{x}}) \ni 0$ is equivalent to $\nabla f(\bar{\mathbf{x}})^T(\mathbf{y} - \bar{\mathbf{x}}) \geq 0 \quad$ for all $\mathbf{y} \in \mathcal{C}$, which is clear from the definition of the normal cone. ■

To simplify the convergence behavior analysis of the proximal method of multipliers on (2.2), we generalize the optimization problem in the following form:

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & c(\mathbf{x}) = \mathbf{0}, \\ & \mathbf{l} \leq \mathbf{x} \leq \mathbf{u} \end{aligned} \tag{5.5}$$

where $f(\mathbf{x})$ and $c(\mathbf{x})$ are continuous and differentiable. Let $\mathcal{C}$ be the closed convex sets corresponding to simple bound constrains $\mathbf{l} \leq \mathbf{x} \leq \mathbf{u}$.

The Lagrangian and augmented Lagrangian function are defined respectively as follows:

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \boldsymbol{\lambda}^T c(\mathbf{x})$$

$$\mathcal{L}_{\mathcal{A}}(\mathbf{x}, \boldsymbol{\lambda}, \sigma) = f(\mathbf{x}) + \boldsymbol{\lambda}^T c(\mathbf{x}) + \frac{\sigma}{2}\|c(\mathbf{x})\|^2.$$

Algorithm 1 Proximal Method of Multipliers

1: Input: Choose $\mathbf{x}_0$, $\boldsymbol{\lambda}_0$, set $k = 0$.
2: Repeat
3: $\quad \mathbf{x}_{k+1} \coloneqq \underset{\mathbf{x} \in \mathcal{C}}{\operatorname{argmin}}\ \mathcal{L}_{\mathcal{A}}(\mathbf{x}, \boldsymbol{\lambda}_k, \sigma_k)$
4: $\qquad\qquad + \frac{1}{2\sigma_k}\|\mathbf{x} - \mathbf{x}_k\|^2 \qquad (P^k)$
5: $\quad \boldsymbol{\lambda}_{k+1} \coloneqq \boldsymbol{\lambda}_k + \sigma_k c(\mathbf{x}_{k+1})$
6: $\quad k \coloneqq k + 1$
7: Until converged

The multipliers $\boldsymbol{\lambda}$ can be added or subtracted. We choose adding the multipliers here in order to be consistent with the analysis in [19].

A point $(\bar{\mathbf{x}}, \bar{\boldsymbol{\lambda}})$ is said to satisfy the strong second-order sufficient condition [20] for problem (5.5) if there is a $\rho \in \mathcal{R}$ such that

$$\begin{aligned} &\langle \boldsymbol{\omega}, \nabla^2_{xx}\mathcal{L}(\bar{\mathbf{x}}, \bar{\boldsymbol{\lambda}})\boldsymbol{\omega} \rangle + \rho \sum_i \langle \nabla c_i(\bar{\mathbf{x}}), \boldsymbol{\omega} \rangle^2 > 0 \\ &\forall \boldsymbol{\omega} \in T_{\mathcal{C}}(\bar{\mathbf{x}})/\{0\} \end{aligned} \tag{5.6}$$

where $T_{\mathcal{C}}(\mathbf{x})$ is the tangent cone of $\mathcal{C}$ at point $\mathbf{x}$.

We describe the proximal method of multipliers for the general form of problem (5.5) in Algorithm 1.

THEOREM 5.1. *Let $(\bar{\mathbf{x}}, \bar{\boldsymbol{\lambda}})$ be a KKT pair for problem (5.5) satisfying the strongly second order sufficient condition and assume the gradients $\nabla c(\bar{\mathbf{x}})$ are linearly independent. If the $\{\sigma_k\}$ are large enough with $\sigma_k \to \bar{\sigma} \leq \infty$ and if $\|(\mathbf{x}_0, \boldsymbol{\lambda}_0) - (\bar{\mathbf{x}}, \bar{\boldsymbol{\lambda}})\|$ is small enough, then there exists a sequence $\{(\mathbf{x}_k, \boldsymbol{\lambda}_k)\}$ conforming to Algorithm 1 along with open neighborhoods $\mathcal{C}_k$ such that for each k, $\mathbf{x}_{k+1}$ is the unique solution in $\mathcal{C}_k$ to (P^k). Then also, the sequence $\{(\mathbf{x}_k, \boldsymbol{\lambda}_k)\}$ converges linearly and Fejér monotonically to $\bar{\mathbf{x}}$, $\bar{\boldsymbol{\lambda}}$ with rate $r(\bar{\sigma}) < 1$ that is decreasing in $\bar{\sigma}$ and $r(\bar{\sigma}) \to 0$ as $\bar{\sigma} \to \infty$.*

Proof. (Note that the theorem and proof are revisions and specializations of Theorem 19 from [19].) By Robinson (1980, Theorem 4.1) [20], the strongly second-order sufficient condition and the linear independence condition imply that the KKT system for (5.5) is strongly regular at $(\bar{\mathbf{x}}, \bar{\boldsymbol{\lambda}})$.

When we solve the subproblem (P^k) with explicit bound constraints, from Proposition 5.1, we actually solve

$$\nabla f(\mathbf{x}) + \frac{1}{\sigma_k}(\mathbf{x} - \mathbf{x}_k) + N_{\mathcal{C}}(\mathbf{x}) + \nabla c(\mathbf{x})^*(\boldsymbol{\lambda}_k + \sigma_k c(\mathbf{x})) \ni 0.$$

Thus, Algorithm 1 is equivalent to Algorithm 3 in [19] (their general algorithm), and by Theorem 17 of [19], we have the local convergence result stated in Theorem 5.1.

It remains to show that for large enough σ_k, the unique stationary point is in fact a minimizer of (P^k).

We apply the second-order sufficient condition in 13.26 from [22] and the analogous derivation in the proof of Theorem 19 of [19]. Then a sufficient condition for $\mathbf{x}_{k+1}$ to be a local minimizer of (P^k) is that

$$\langle \boldsymbol{\omega}, \nabla^2_{xx}\mathcal{L}(\mathbf{x}_{k+1}, \boldsymbol{\lambda}_{k+1})\boldsymbol{\omega}\rangle + \frac{1}{\sigma_k}\|\boldsymbol{\omega}\|^2 + \sigma_k \sum_i \langle \nabla c_i(\mathbf{x}_{k+1}), \boldsymbol{\omega}\rangle^2 > 0, \forall \boldsymbol{\omega} \in T_{\mathcal{C}}(\mathbf{x}_{k+1})/\{0\}.$$

This condition holds by the continuity of $\nabla^2_{xx}\mathcal{L}$ and ∇c_i, and by (5.6), provided σ_k is large enough. ■

A main assumption for the analysis above is that we can solve the subproblem (P^k) exactly. This was adjusted in [13], which showed local convergence for approximate solutions of (P^k).

6 Simplified Alternating Direction Method of Multipliers

One downside to both of the proposed methods is that they involve using the L-BFGS-B method to solve the bound-constrained non-convex objectives in the substeps. This is a complex routine with a high runtime itself. In this section, we are interested in seeing if there are simplified ADMM variants that might futher improve runtime by avoiding this non-convex solver. This corresponds to, for example, inexact ADMM (allowing inexact primal alternating minimization solutions, e.g., one proximal gradient step per block).

In the ADMM method from Section 4.2, we know that updating the block variables $\mathbf{f}$, $\mathbf{g}$, $\mathbf{s}$ and r is simple and convex, so we can get globally optimal solutions. The only hard part is to update $\boldsymbol{Y}$, which is non-convex. However, there are few results about convergence for ADMM in the non-convex case as well as the case with multiple blocks, i.e., more than two blocks of variables (e.g., $\boldsymbol{Y}$, $\mathbf{f}$, $\mathbf{g}$, $\mathbf{s}$ and r) that would apply to this problem. For instance, in [7], it has been shown that an ADMM method does not converge for a multi-block case even for a convex problem.

In fact, in our preliminary investigations, many common variations on the ADMM methods did not yield any performance increase or resulted in slower performance or did not converge at all. For example, we tried to avoid the L-BFGS-B in the update for $\boldsymbol{Y}$ by simply using a step of projected gradient descent instead. We found the resulting Simplified ADMM (SADMM) method converges much slower than our ADMM method with the non-convex solver (more details are in Section 7.1). The same experiment with multiple steps of projected gradient descent only performed worse.

Therefore, common accelerated variants of ADMM proposed for convex problems with two-block case do not necessarily improve the performance of ADMM in our problem. We believe that the NEO-K-means low-rank SDP problem will be a useful test case for future research in this area.

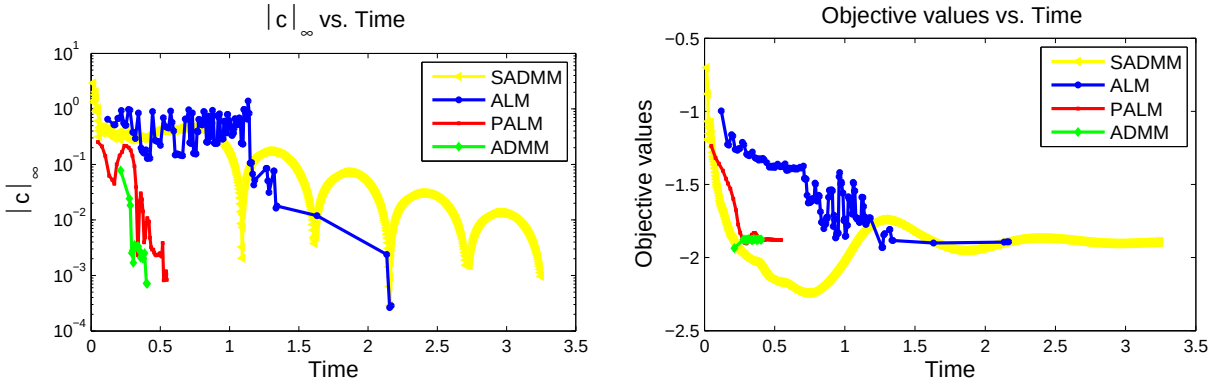
7 Experimental Results

In this section, we demonstrate the efficiency of our proposed methods on real-world problems. Our primary goal is to compare our two new methods, PALM and ADMM with the classical augmented Lagrangian method (denoted by ALM) in terms of their ability to optimize (2.2). All these three algorithms are implemented in MATLAB and use the L-BFGS-B routine [6] written in Fortran to solve the bound-constrained nonlinear subproblems.

7.1 Convergence Analysis on the Karate Club Network. We first illustrate the convergence behavior of each of the methods on an overlapping community detection task in a graph. We use the Zachary's karate club network [25] which is a small social network among 34 members of a karate club.

In Figure 1, (a) shows the infinity norm of the constraints vector and (b) shows the NEO-K-Means low-rank SDP objective function values defined in (2.2) as time progresses respectively. We set the infeasibility tolerance to be less than 10^{-3}. Both of our methods, PALM and ADMM, achieve faster convergence than ALM in terms of both the feasibility of the solution and the objective function value mainly because the subproblems for L-BFGS-B are faster to solve. To demonstrate that the common variants of ADMM do not accelerate the convergence in our problem, we also compare with the simplified alternating direction method of multipliers (Section 6, denoted by SADMM). Note that for SADMM, we do not need to use L-BFGS-B to solve the subproblems, instead, we use one single gradient-descent step to have the solution inexactly. It is clear to see that SADMM is much slower than ADMM, and even slower than ALM.

7.2 Data Clustering on Real-world Datasets. Next, we compare the three methods (ALM, PALM and ADMM) on larger datasets. We use three different datasets from [1]. The SCENE dataset [4] contains 2,407 scenes represented as feature vectors; the YEAST dataset [9] consists of 2,417 genes where the features are based on micro-array expression data and phylogenetic profiles; the MUSIC dataset [23] contains a set of 593 different songs. There are known ground-truth clusters on these datasets (we set k as the number of ground-



(a) The infinity norm of the constraints vector vs. Time

(b) Objective function values vs. Time

Figure 1: The convergence behavior of ALM, PALM, ADMM and SADMM on a Karate Club network. PALM and ADMM converge faster than ALM while SADMM is much slower.

truth clusters; k=6 for MUSIC and SCENE, and k=14 for YEAST). The goal of this comparison is to demonstrate that PALM and ADMM have performance equivalent to the ALM method, while running substantially faster. We will also compare against the iterative NEO-K-Means algorithm as a reference.

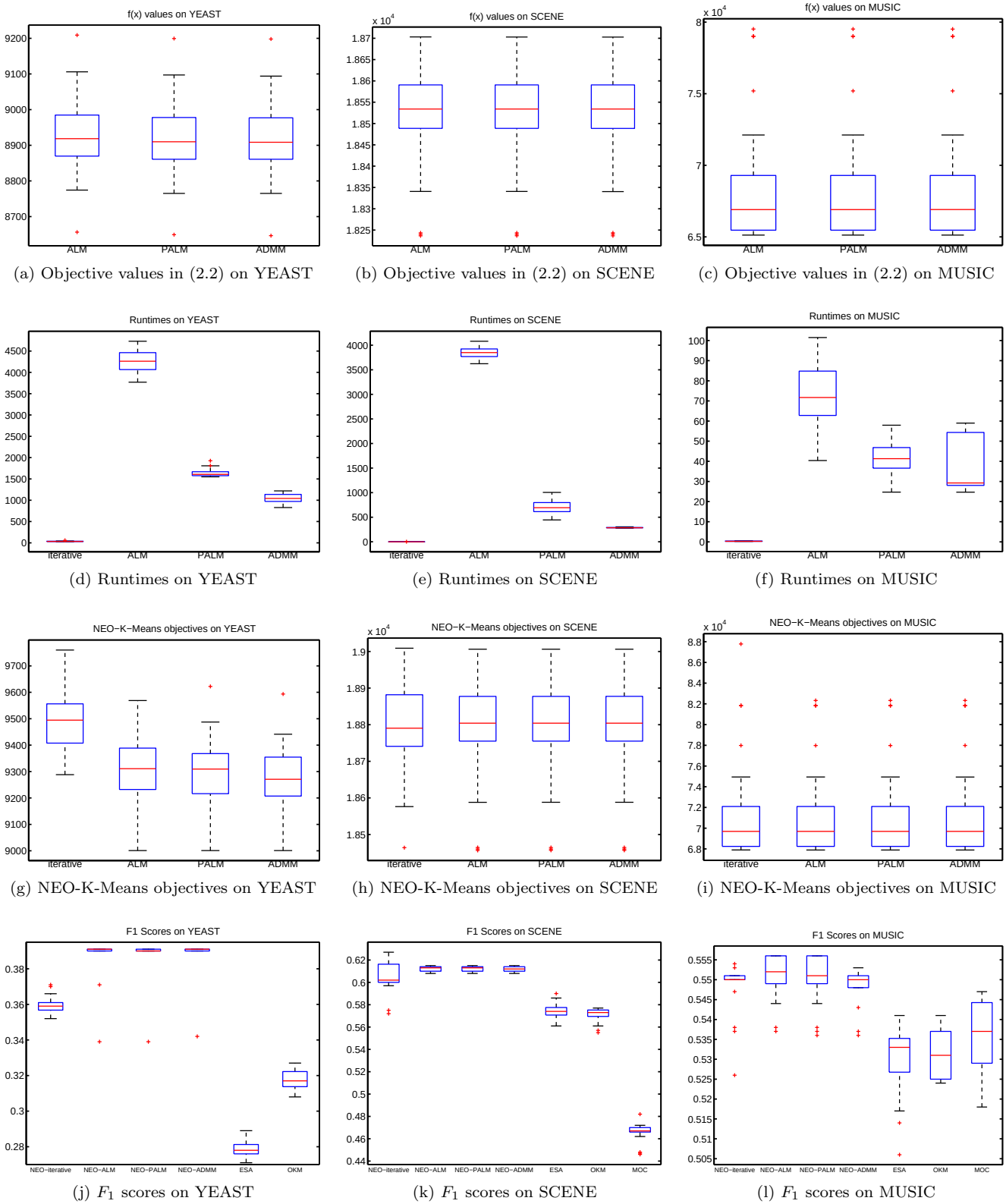
We initialize ALM, PALM, and ADMM using the iterative NEO-K-Means algorithm as also used in [11]. The parameters α and β in the NEO-K-Means are automatically estimated by the strategies proposed in [24]. This initialization renders the method sensitive to the local region selected by the iterative method, but this is usually a high-quality region. We use the procedure from [11] to round the real-valued solutions to discrete assignments. Briefly, this uses the solution vectors $\mathbf{g}$ and $\mathbf{f}$ to determine which points to assign and roughly how many clusters each data point resides in. Assignments are then greedily made based on values of the solution matrix $\boldsymbol{Y}$. We run all the methods 25 times on the three datasets, and summarize the results in Figure 2. The results from these experiments illustrate the following points:

- (Top-row – objective values) The PALM, ADMM, and ALM methods are all indistinguishable as far as their ability to optimize the objective of the NEO-K-Means low-rank SDP problem (2.2).
- (Second-row – runtimes) Both the PALM and ADMM methods are significantly faster than ALM on the larger two datasets, SCENE and YEAST. In particular, ADMM is more than 13 times faster on the SCENE dataset. Since the MUSIC dataset is relatively small, the speedup is also relatively small, but the two new methods, PALM and ADMM are consistently faster than ALM. Note that we do not expect any of the optimization-based methods will be faster than the iterative NEO-K-Means method since it is a completely different type of algorithm (In particular, it optimizes the discretized objective).

Thus, we conclude that the new optimization procedures (PALM and ADMM) are considerably faster than the ALM method while achieving similar objective function values.

The next investigation studies the discrete assignments produced by the methods. Here, we see that (third row of Figure 2) there are essentially no differences among any of the optimization methods (ALM, PALM, ADMM) in terms of their objective values after rounding to the discrete solution and evaluating the NEO-K-Means objective. The optimization methods outperform the iterative method on the YEAST dataset by a considerable margin.

Finally, to see the clustering performance, we compute the F_1 score which measures the matching between the algorithmic solutions and the ground-truth clusters in the last row of Figure 2. Higher F_1 scores indicate better alignment with the ground-truth clusters. We also compare the results with other state-of-the-art overlapping clustering methods, MOC [3], ESA [15], and OKM [8]. On the YEAST dataset, MOC returns 13 empty clusters and one large cluster which contains all the data points. So, we do not report F_1 score of MOC on this dataset. We first observe that the NEO-K-Means based methods (denoted by NEO-*) are able to achieve higher F_1 scores than other methods. When we compare the performance among the three NEO-K-Means optimization methods (NEO-ALM, NEO-PALM, and NEO-ADMM), there is largely no difference among these methods except for the MUSIC dataset. On the MUSIC problem, the ADMM method has a slightly lower F_1 score than PALM or ALM. This is because the objective values obtained by



(a) Objective values in (2.2) on YEAST (b) Objective values in (2.2) on SCENE (c) Objective values in (2.2) on MUSIC

(d) Runtimes on YEAST (e) Runtimes on SCENE (f) Runtimes on MUSIC

(g) NEO-K-Means objectives on YEAST (h) NEO-K-Means objectives on SCENE (i) NEO-K-Means objectives on MUSIC

(j) F_1 scores on YEAST (k) F_1 scores on SCENE (l) F_1 scores on MUSIC

Figure 2: Box-plots comparing the results on 25-trials of the algorithms in terms of objective values in (2.2), runtimes, NEO-K-Means objective values in (2.1), and F_1 scores on YEAST, SCENE, and MUSIC datasets. The median performance is indicated by the middle red line and the box shows the 25% and 75% percentiles.

ADMM on this dataset seem to be minutely higher than the other two optimization strategies and this manifests as a noticeable change in the F_1 score. In this case, however, the scale of the variation is low and essentially, the results from all the NEO-K-Means based methods are equivalent. On the SCENE dataset, the iterative algorithm (NEO-iterative) can sometimes outperform the optimization methods in terms of F_1 although we note that the median performance of the optimization is much better and there is essentially no difference between NEO-PALM, NEO-ADMM, and NEO-ALM. On the YEAST dataset, the reduced objective function value corresponds with an improvement in the F_1 scores for notably better results than NEO-iterative.

8 Discussion

Overall, the result from the previous section indicate that both the PALM and ADMM methods are faster than ALM with essentially no change in quality. Thus, we can easily recommend them instead of ALM for optimizing these low-rank objectives. There is still a substantial gap between the performance of the simple iterative algorithm and the optimization procedures we propose here. However, the optimization procedures avoid the worst-case behavior of the iterative method and result in more robust and reliable results as illustrated on the YEAST dataset and in other experiments from [11].

In terms of future opportunities, we are attempting to identify a convergence guarantee for the ADMM method in this non-convex case. This would put the fastest method we have for the optimization on firm theoretical ground. In terms of the clustering problem, we are exploring the integrality properties of the SDP relaxation itself [11]. Our goal here is to show a result akin to that proved in [2] about integrality in relaxations of the k-means objective. Finally, another goal we are pursuing involves understanding when our method can recover the partitions from an overlapping block-model with outliers. This should hopefully show that the optimization approaches have a wider recovery region than the simple iterative methods and provide a theoretical basis for empirically observed improvement.

Acknowledgments This research was supported by NSF CAREER award CCF-1149756 to DG, and by NSF grants CCF-1117055 and CCF-1320746 to ID.

References

[1] Mulan: A Java Library for Multi-Label Learning. `http://mulan.sourceforge.net/datasets.html`.

[2] P. Awasthi, A. S. Bandeira, M. Charikar, R. Krishnaswamy, S. Villar, and R. Ward. Relax, no need to round: integrality of clustering formulations. In *ITCS*, pages 191–200, 2015.

[3] A. Banerjee, C. Krumpelman, J. Ghosh, S. Basu, and R. J. Mooney. Model-based overlapping clustering. In *KDD*, pages 532–537, 2005.

[4] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown. Learning multi-label scene classification. *Pattern Recognition*, 37(9):1757 – 1771, 2004.

[5] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122, Jan. 2011.

[6] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. *SIAM J. Sci. Comput.*, 16(5):1190–1208, 1995.

[7] C. Chen, B. He, Y. Ye, and X. Yuan. The direct extension of admm for multi-block convex minimization problems is not necessarily convergent. *Math. Prog.*, pages 1–23, 2014.

[8] G. Cleuziou. An extended version of the k-means method for overlapping clustering. In *ICPR*, pages 1–4, 2008.

[9] A. Elisseeff and J. Weston. A kernel method for multi-labelled classification. In *NIPS*, pages 681–687, 2001.

[10] M. Friedlander and D. Orban. A primal-dual regularized interior-point method for convex quadratic programs. *Math. Prog. Comput.*, 4(1):71–107, 2012.

[11] Y. Hou, J. J. Whang, D. F. Gleich, and I. S. Dhillon. Non-exhaustive, overlapping clustering via low-rank semidefinite programming. In *KDD*, pages 427–436, 2015.

[12] J. Humes, Carlos, P. Silva, and B. Svaiter. Some inexact hybrid proximal augmented Lagrangian algorithms. *Numerical Algorithms*, 35(2-4):175–184, 2004.

[13] A. N. Iusem, T. Pennanen, and B. F. Svaiter. Inexact variants of the proximal point algorithm without monotonicity. *SIAM J. Optimiz.*, 13(4):1080–1097, 2003.

[14] S. P. Lloyd. Least squares quantization in PCM. *IEEE Trans. Inf. Theory*, 28(2):129–137, 1982.

[15] H. Lu, Y. Hong, W. N. Street, F. Wang, and H. Tong. Icdm workshops. In *Overlapping clustering with sparseness constraints*, pages 486–494, 2012.

[16] S. Magnússon, P. C. Weeraddana, M. G. Rabbat, and C. Fischione. On the convergence of alternating direction Lagrangian methods for nonconvex structured optimization problems. *IEEE Trans. Control Netw. Syst.*, 2015.

[17] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, 2nd edition, 2006.

[18] N. Parikh and S. Boyd. Proximal algorithms. *Found. Trends Opt.*, 1(3):127–239, 2014.

[19] T. Pennanen. Local convergence of the proximal point algorithm and multiplier methods without monotonicity. *Math. Oper. Res.*, 27(1):170–191, 2002.

[20] S. M. Robinson. Strongly regular generalized equations. *Math. Oper. Res.*, 5(1):43–62, 1980.

[21] R. T. Rockafellar. Augmented Lagrangians and applications of the proximal point algorithm in convex programming. *Math. Oper. Res.*, 1(2):97–116, 1976.

[22] R. T. Rockafellar and R. J.-B. Wets. *Variational analysis*. Springer, 2009.

[23] K. Trohidis, G. Tsoumakas, G. Kalliris, and I. P. Vlahavas. Multi-label classification of music into emotions. In *ISMIR*, pages 325–330, 2008.

[24] J. J. Whang, I. S. Dhillon, and D. F. Gleich. Non-exhaustive, overlapping k-means. In *SDM*, pages 936–944, 2015.

[25] W. W. Zachary. An information flow model for conflict and fission in small groups. *J. Anthropol. Res.*, 33(4):452–473, 1977.