Non-exhaustive, Overlapping Clustering via Low-Rank Semidefinite Programming

Yangyang Hou*
Purdue University
West Lafayette, IN
hou13@purdue.edu

Joyce Jiyoung Whang*
University of Texas at Austin
Austin, TX
joyce@cs.utexas.edu

David F. Gleich
Purdue University
West Lafayette, IN
dgleich@purdue.edu

Inderjit S. Dhillon
University of Texas at Austin
Austin, TX
inderjit@cs.utexas.edu

ABSTRACT

Clustering is one of the most fundamental tasks in data mining. To analyze complex real-world data emerging in many data-centric applications, the problem of non-exhaustive, overlapping clustering has been studied where the goal is to find overlapping clusters and also detect outliers simultaneously. We propose a novel convex semidefinite program (SDP) as a relaxation of the non-exhaustive, overlapping clustering problem. Although the SDP formulation enjoys attractive theoretical properties with respect to global optimization, it is computationally intractable for large problem sizes. As an alternative, we optimize a low-rank factorization of the solution. The resulting problem is non-convex, but has a smaller number of solution variables. We construct an optimization solver using an augmented Lagrangian methodology that enables us to deal with problems with tens of thousands of data points. The new solver provides more accurate and reliable answers than other approaches. By exploiting the connection between graph clustering objective functions and a kernel k-means objective, our new low-rank solver can also compute overlapping communities of social networks with state-of-the-art accuracy.

Categories and Subject Descriptors

I.5.3 [**Pattern Recognition**]: Clustering—*Algorithms*

General Terms

Algorithms, Experimentation

Keywords

Overlapping Clustering, Community Detection, Semidefinite Programming

*Authors in alphabetical order with equal contribution.

KDD'15, August 10-13, 2015, Sydney, NSW, Australia.

ACM 978-1-4503-3664-2/15/08 ...$15.00.
DOI: http://dx.doi.org/10.1145/2783258.2783398.

1. INTRODUCTION

Clustering is one of the most widely used primitives in data mining. The goal of clustering is to take a set of data points and assign them to groups, called clusters, such that similar data points are assigned to the same cluster. The traditional clustering algorithms, e.g., k-means, assign each data point to exactly one cluster. This assignment might be appropriate when clear groups exist in the data. We consider the clustering problem when the data points do not have an obvious separation into a small number of groups and may contain both outliers and large regions of overlap between groups. This setting is especially applicable to emerging types of data such as social networks [8], where clusters (or communities) overlap due to the multiple personas that individuals adopt. Another example is clustering biological genes and functions, which overlap because genes can serve multiple functions [32].

We recently proposed a new formulation of this problem [30] called non-exhaustive, overlapping k-means (abbreviated NEO-K-Means) that seamlessly generalizes the classic k-means clustering objective. A kernelized and weighted version allows us to equate the NEO-K-Means problem to the problem of minimizing the normalized cut of an overlapping clustering of a graph. We also proposed a novel iterative algorithm for the NEO-K-Means objective that monotonically decreases the clustering objective. It was based on a generalization of the standard k-means iterative assignment algorithm (also called Lloyd's algorithm [23]). Using this procedure, we were able to automatically cluster a variety of datasets that contain overlapping clusters as well as outliers. When we tested our algorithm for the task of matching with the ground-truth clusters, we produced clusters that have state-of-the-art performance. When we used our algorithm for community detection problem, our algorithm returned communities that have the lowest normalized cut scores of any existing algorithms. That iterative procedure is *fast*, but suffers from the classic problem that iterative algorithms for k-means fall into local minimizers given poor initialization of the clusters. This is frequently addressed by running the algorithm multiple times with random initialization or using distance based initialization strategies [3]. k-means++

In this manuscript, we continue our study of the non-exhaustive, overlapping cluster objective function by proposing a convex relaxation (Section 3). This convex problem

Convex problem too expensive

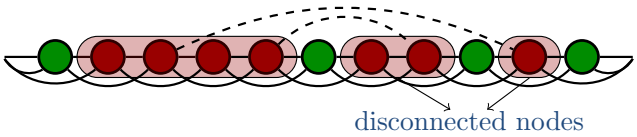
can be globally optimized in time and memory that is polynomial in the input size. The relaxed solution can then be rounded to a discrete assignment solution. Our experimental results with this algorithm show that it results in better objective function values than our previous iterative algorithm [30], albeit at a substantial computational cost.

The convex formulation is not without problems. When the NEO-K-Means problem is relaxed to a convex semidefinite program (SDP), the number of variables is *quadratic* in the number of data points. Off-the-shelf SDP solvers such as CVX [15, 14] can then only solve problems with fewer than 100 data points (this is due to a variety of complexities that arise when our SDP is converted into a standard form for existing convex solvers). Even small modern datasets have a few thousand points, and they require a different approach.

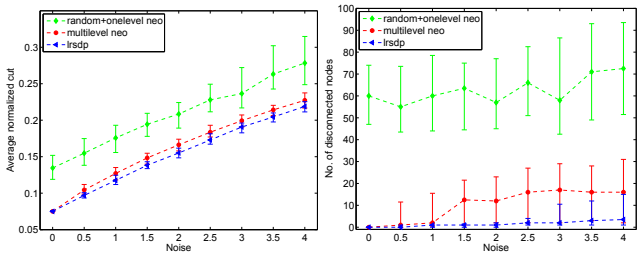
Consequently, we propose optimizing a low-rank factorization of the SDP solution matrix (Section 4). This is a standard technique to tackle large-scale SDPs [6]. The resulting optimization problem is a quadratically constrained problem with a quadratic objective that can no longer be globally optimized. An augmented Lagrangian procedure, for instance, will only converge to a local minimizer. Nevertheless, when this approach has been used in the past with high-quality optimization methods, it frequently generates solutions that are *as good as the global optimal* from convex solvers [6], a fact which has some theoretical justification [7]. Furthermore, similar ideas yielded stability improvements to convex relaxations of the classic k-means objective [18].

Our new LRSDP algorithm to optimize this non-linear problem can handle problems with tens of thousands of data points, providing an order of magnitude increase in scalability over the convex solver. On the problems where we can compare with the convex formulations, we achieve globally optimal objective values. It also consistently outperforms the iterative algorithm for NEO-K-Means [30] in terms of objective function value.

Our goal with the new procedure is to produce more accurate and reliable clusterings than the previous iterative algorithm [30] in the regime of medium-scale problems. This regime is ideal because the new method is more computationally expensive than the iterative algorithm, which was an efficient procedure designed for problems with millions of data points. To see the difference between these methods, we study the behavior on a synthetic problem with community detection on a cycle graph. The graph is a Watts-Strogatz random graph where each node has edges to five neighbors on each side of the cycle. We also add random edges based on an Erdös-Rényi graph with expected degree d, which we consider as noise edges. When the noise is low, clusterings should respect the cycle structure and be continuous, connected regions. Hence, we compute an error measure for each cluster based on the number of points disconnected from the largest connected component in the cycle; this measure is illustrated in Figure 1(a). We compare three methods: the straight-forward iterative NEO-K-Means method with random initialization, a multilevel variation on that method [30], and our LRSDP with random initialization. We run 100 trials and plot the the median, 25th and 75th percentiles of the normalized cut scores and the number of disconnected nodes by varying the noise level. Figure 1(b) & Figure 1(c) show the results. Our LRSDP method achieves the best performance in terms of both the normalized cut and the number of disconnected nodes. We observe that our



(a) The disconnected nodes error measure counts number of nodes that are disconnected from the largest connected component. These nodes are illustrated for the cluster in red. (Green nodes are not in that cluster.)



(b) Avg. normalized cut (c) No. of disconnected points

Figure 1: A synthetic study of overlapping community detection on a Watts-Strogatz cycle graph where each point should be assigned to two clusters: (a) an illustration of a portion of the cycle with dashed 'noise' edges and showing the disconnected points measure (which is 3); (b) & (c) the results of normalized cut and the number of disconnected points on graphs with 100 nodes returned by our new LRSDP procedure compared with two variations of our previous "neo" iterative algorithms.

LRSDP method often produces 0 disconnected points even as the noise increases whereas the faster iterative method starts to introduce many disconnected points with only a modest amount of error.

We now summarize the contributions of this paper:

- We propose NEO-SDP: a convex relaxation of a k-means-like objective that handles non-exhaustive, overlapping clustering problems (Section 3).
- We formulate the scalable NEO-LR objective and an LRSDP algorithm to optimize a low-rank factorization of the NEO-SDP solution (Section 4).
- We also propose a series of initialization and rounding strategies that accelerate the convergence of our optimization procedures (Section 4.3).
- We evaluate LRSDP on real-world data clustering problems and find it achieves the best F_1 performance with respect to ground-truth clusters (Section 6.3).
- For graph clustering problems, LRSDP produces the best quality communities among all clustering algorithms on real-world networks (Section 6.4).

2. PRELIMINARIES

We begin our technical exposition by reviewing the NEO-K-Means objective and the iterative algorithm that we previously proposed [30], then we briefly review related work on semidefinite programs (SDPs) and low-rank SDPs.

2.1 The NEO-K-Means objective

Given a set of data points $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$, the goal of *non-exhaustive, overlapping clustering* is to compute 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

the clusters need not be disjoint. Furthermore, we wish for the clusters to respect the natural groups of the data.

The NEO-K-Means objective [30] is a way of encoding this problem. It depends on a set of data points $\mathcal{X}$, a parameter k for the number of clusters, two parameters α and β that determine the amount of overlap and non-exhaustiveness, respectively. The weighted and kernelized version also depends on a positive weight for each data point, $w_i > 0$ and a feature map $\phi(\mathbf{x})$. Let $\boldsymbol{U} = [u_{ij}]_{n\times k}$ be an assignment matrix such that $u_{ij} = 1$ if a data point $\mathbf{x}_i$ belongs to $\mathcal{C}_j$; and $u_{ij} = 0$ otherwise. The weighted kernel NEO-K-Means objective function is defined as follows:

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

The two constraints imply that we make exactly $(1+\alpha)n$ assignments and allow at most βn data points to have no membership in any cluster. If $\alpha = 0$ and $\beta = 0$, then this objective reduces to the classic weighted kernel k-means objective. In [30], we propose a strategy to automatically select α and β given a dataset; see that manuscript for the details. Selecting k and picking a feature map must be carefully considered based on the data and application.

The NEO-K-Means iterative algorithm. The NEO-K-Means algorithm is a generalization of the k-means iterative assignment procedure that first makes $(1-\beta)n$ assignments from data points to the nearest cluster centroids to satisfy the non-exhaustiveness condition. Then it makes $(1+\alpha)n - (1-\beta)n$ additional assignments that can overlap based on the smallest distances between data points and centroids. This procedure produces a non-increasing sequence of objective function values.

Graph clustering. In another line of prior work, [12] showed that optimizing the normalized cut objective is equivalent to a particular weighted, kernel k-means objective. Given a clustering of a graph, the normalized cut of a clustering is the sum of normalized cut scores of each cluster:

$$\operatorname{ncut}(\mathcal{C}) = \sum_{j=1}^{k} \operatorname{ncut}(\mathcal{C}_j) = \sum_{j=1}^{k} \frac{\operatorname{cut}(\mathcal{C}_j)}{\operatorname{links}(\mathcal{C}_j, \mathcal{V})},$$

where $\operatorname{cut}(\mathcal{C}_j)$ is the number of edges leaving the cluster, and $\operatorname{links}(\mathcal{C}_j, \mathcal{V})$ is the sum of degrees for all vertices in $\mathcal{C}_j$; see [12] for more on this objective in the context of k-means. An extended version of this idea holds for the NEO-K-Means problem for normalized cut-based overlapping graph clustering [30]. Thus, it is possible to use the NEO-K-Means formulation to produce a set of overlapping clusters on a graph to minimize the sum of normalized cuts. The overlapping graph clustering problem is closely related to community detection, and the iterative NEO-K-Means algorithm achieves state-of-the-art performance at finding ground-truth communities in large networks.

2.2 Low-rank factorizations of SDPs

Semidefinite programs (SDPs) are one of the most general classes of tractable convex optimization problems. The canonical form and low-rank variation are:

$\mathbf{x}_i$	the data points for k-means	§2
k	the number of clusters	
α	the overlap parameter (0 means no overlap)	
β	the outlier parameter (0 means no outliers)	
$\boldsymbol{U}$	the assignment matrix for a solution	§2
$\boldsymbol{Z}$	the co-occurence matrix for the SDP relaxation	§3
$\boldsymbol{K}$	the kernel matrix for NEO-K-Means	§3
$\boldsymbol{W}$	a diagonal weight matrix for weighted problems	§3
$\mathbf{d}$	a specialized weight vector for the SDP relaxation	§3
$\mathbf{f}$	the cluster count variable for the SDP relaxation	§3
$\mathbf{g}$	the outlier indicator for the SDP relaxation	§3
$\boldsymbol{Y}$	the low-rank approximation of $\boldsymbol{Z}$ in NEO-LR	§4

Table 1: A summary of the notation used in the NEO-K-Means problem, the final assignment, and the SDP and low-rank approximations.

Canonical SDP

$$\begin{aligned} \text{maximize} \quad & \operatorname{trace}(\boldsymbol{C}\boldsymbol{X}) \\ \text{subject to} \quad & \boldsymbol{X} \succeq 0, \boldsymbol{X} = \boldsymbol{X}^T, \\ & \operatorname{trace}(\boldsymbol{A}_i\boldsymbol{X}) = b_i \\ & i = 1, \ldots, m \end{aligned}$$

Low-rank SDP

$$\begin{aligned} \text{maximize} \quad & \operatorname{trace}(\boldsymbol{C}\boldsymbol{Y}\boldsymbol{Y}^T) \\ \text{subject to} \quad & \boldsymbol{Y} : n \times k \\ & \operatorname{trace}(\boldsymbol{A}_i\boldsymbol{Y}\boldsymbol{Y}^T) = b_i \\ & i = 1, \ldots, m \end{aligned}$$

Notice the low-rank form drops the positive semidefinite ($\boldsymbol{X} \succeq 0$) and symmetry constraints ($\boldsymbol{X} = \boldsymbol{X}^T$) but replaces $\boldsymbol{X} = \boldsymbol{Y}\boldsymbol{Y}^T$, which automatically satisfies these constraints. Canonical SDPs can be optimized by a variety of solvers such as CVX [15, 14]. Low-rank SDP factorizations are non-convex and are locally optimized via an augmented Lagrangian method [6]; see the Appendix for a review of the augmented Lagrangian idea.

3. AN SDP FOR NEO-K-MEANS

We begin by stating an exact SDP-like program for the weighted kernel NEO-K-Means objective and then describe how to relax it to an SDP. We use the same notation as the previous section and summarize our common notation in Table 1. The essential idea with the SDP-like version is that we replace the assignment matrix $\boldsymbol{U}$ with a normalized cluster co-occurrence matrix $\boldsymbol{Z}$:

$$\boldsymbol{Z} = \sum_{c=1}^{k} \frac{\boldsymbol{W}\mathbf{u}_c(\boldsymbol{W}\mathbf{u}_c)^T}{s_c}$$

where $\boldsymbol{W}$ is a diagonal matrix with the data point weights w_i on the diagonal, $\mathbf{u}_c$ is the c-th column of matrix $\boldsymbol{U}$ and $s_c = \mathbf{u}_c^T\boldsymbol{W}\mathbf{u}_c$. When $\boldsymbol{Z}$ is defined from an assignment matrix $\boldsymbol{U}$, then values of Z_{ij} are non-zero when items co-occur in a cluster. With appropriate constraints on the matrix $\boldsymbol{Z}$, it serves as a direct replacement for the assignment matrix $\boldsymbol{U}$.

To state the problem, let $\boldsymbol{K}$ denote the kernel matrix of the data points, e.g., if $\boldsymbol{X}$ is the data matrix whose rows correspond to data vectors, then $\boldsymbol{K} = \boldsymbol{X}\boldsymbol{X}^T$ is just the simple linear kernel matrix. Let $\mathbf{d}$ be a vector where $d_i = w_iK_{ii}$, i.e., a weighted diagonal from $\boldsymbol{K}$. We need two new types of variables as well:

- Let $\mathbf{f}$ denote a vector of length n such that the i-th entry indicates the number of clusters that data point i belongs to.
- Similarly, let $\mathbf{g}$ denote a vector of length n such that the i-th entry is one if that data point i belongs to any clusters, and zero if the data point does not belong to any cluster.

Finally, we denote by $\mathbf{e}$ the vector of all 1s.

The following program is equivalent to the NEO-K-Means objective with a discrete assignment matrix:

$$\begin{array}{lll} \underset{\boldsymbol{Z},\mathbf{f},\mathbf{g}}{\text{maximize}} & \text{trace}(\boldsymbol{K}\boldsymbol{Z}) - \mathbf{f}^T\mathbf{d} & \\ \text{subject to} & \text{trace}(\boldsymbol{W}^{-1}\boldsymbol{Z}) = k, & (a) \\ & Z_{ij} \geq 0, & (b) \\ & \boldsymbol{Z} \succeq 0, \boldsymbol{Z} = \boldsymbol{Z}^T & (c) \\ & \boldsymbol{Z}\mathbf{e} = \boldsymbol{W}\mathbf{f}, & (d) \\ & \mathbf{e}^T\mathbf{f} = (1+\alpha)n, & (e) \\ & \mathbf{e}^T\mathbf{g} \geq (1-\beta)n, & (f) \\ & \mathbf{f} \geq \mathbf{g}, & (g) \\ & \text{rank}(\boldsymbol{Z}) = k, & (h) \\ & \mathbf{f} \in \mathcal{Z}^n_{\geq 0}, \mathbf{g} \in \{0,1\}^n. & (i) \end{array} \tag{2}$$

We omit the verification that this is actually equivalent to the NEO-K-Means objective (1) as it is not informative for our discussion. Constraints $(a), (b), (c)$, and (h) encode the fact that $\boldsymbol{Z}$ must arise from an assignment matrix. Constraints $(d), (e), (f), (g)$, and (i) are new to our NEO-K-Means formulation that express the amount of overlap and nonexhaustiveness in the solution. This is a mixed-integer, rank constrained SDP. As such, it is combinatorially hard to optimize just like the original NEO-K-Means objective.

The constraints that make this a combinatorial problem are (h) and (i). If we relax these constraints:

$$\begin{array}{lll} \underset{\boldsymbol{Z},\mathbf{f},\mathbf{g}}{\text{maximize}} & \text{trace}(\boldsymbol{K}\boldsymbol{Z}) - \mathbf{f}^T\mathbf{d} & \\ \text{subject to} & (a), (b), (c), (d), (e), (f), (g) & \\ & 0 \leq \mathbf{g} \leq 1 & \end{array} \tag{3}$$

then we arrive at a convex problem. Thus, any local optimal solution of (3) must be a global solution.

Solving (3) requires a black-box sdp solver such as CVX. As it converts this problem into a standard form for such problems, the number of variables becomes $\mathcal{O}(n^2)$ and the resulting complexity is worse than $\mathcal{O}(n^3)$ in most cases, and can be as bad as $\mathcal{O}(n^6)$. These solvers are further limited by the delicate numerical precision issues that arise as they approach a solution. The combination of these features means that off-the-shelf procedures struggle to solve problems with more than 100 data points. We now describe a means to enable us to solve larger problems.

4. A LOW-RANK SDP FOR NEO-K-MEANS

In the SDP formulation of the NEO-K-Means objective (3), the matrix $\boldsymbol{Z}$ should only be rank k. By applying the low-rank factorization idea, $\boldsymbol{Z}$ becomes $\boldsymbol{Y}\boldsymbol{Y}^T$ where $\boldsymbol{Y}$ is $n \times k$ and non-negative. Thus, the following optimization program is a low-rank SDP for (3) (we have chosen to write it in the standard form of a minimization problem with explicit slack variables $\mathbf{s}, r$ to convert the inequality constraints into equality and bound constraints).

$$\begin{array}{lll} \underset{\boldsymbol{Y},\mathbf{f},\mathbf{g},\mathbf{s},r}{\text{minimize}} & \mathbf{f}^T\mathbf{d} - \text{trace}(\boldsymbol{Y}^T\boldsymbol{K}\boldsymbol{Y}) & \\ \text{subject to} & k = \text{trace}(\boldsymbol{Y}^T\boldsymbol{W}^{-1}\boldsymbol{Y}) & (s) \\ & 0 = \boldsymbol{Y}\boldsymbol{Y}^T\mathbf{e} - \boldsymbol{W}\mathbf{f} & (t) \\ & 0 = \mathbf{e}^T\mathbf{f} - (1+\alpha)n & (u) \\ & 0 = \mathbf{f} - \mathbf{g} - \mathbf{s} & (v) \\ & 0 = \mathbf{e}^T\mathbf{g} - (1-\beta)n - r & (w) \\ & Y_{ij} \geq 0, \mathbf{s} \geq 0, r \geq 0 & \\ & 0 \leq \mathbf{f} \leq k\mathbf{e}, 0 \leq \mathbf{g} \leq 1 & \end{array} \tag{4}$$

Here we also replaced the constraint $\boldsymbol{Y}\boldsymbol{Y}^T \geq 0$ with the stronger constraint $\boldsymbol{Y} \geq 0$. This problem is a quadratic programming problem with quadratic constraints, and we will discuss how to solve it in the next subsection. We call the problem NEO-LR and the solution procedure LRSDP. Even though now we lose convexity by formulating the low rank SDP, this nonlinear programming problem only requires $\mathcal{O}(nk)$ memory and existing nonlinear programming techniques allow us to scale to large problems.

After we get a solution, the solution $\boldsymbol{Y}$ can be regarded as the normalized assignment matrix

$$\boldsymbol{Y} = \boldsymbol{W}\hat{\boldsymbol{U}}$$

where $\hat{U} = [\hat{\mathbf{u}}_1, \hat{\mathbf{u}}_2, \ldots, \hat{\mathbf{u}}_k]$, and $\hat{\mathbf{u}}_c = \mathbf{u}_c/\sqrt{s_c}$ for any $c = 1, \ldots, k$.

4.1 Solving the NEO-K-Means low-rank SDP

To solve the NEO-LR problem (4), we use an augmented Lagrangian framework. This is an iterative strategy where each step consists of minimizing an augmented Lagrangian of the problem that includes a current estimate of the Lagrange multipliers for the constraints as well as a penalty term that drives the solution towards the feasible set. Augmented Lagrangian techniques have been successful in previous studies of low-rank SDP approximations [6].

Let $\lambda = [\lambda_1; \lambda_2; \lambda_3]$ be the Lagrange multipliers associated with the three scalar constraints $(s), (u), (w)$, and $\boldsymbol{\mu}$ and $\boldsymbol{\gamma}$ be the Lagrange multipliers associated with the vector constraints (t) and (v), respectively. Let $\sigma \geq 0$ be a penalty parameter. The augmented Lagrangian for (4) is:

$$\begin{aligned} &\mathcal{L}_A(Y, \mathbf{f}, \mathbf{g}, \mathbf{s}, r; \lambda, \boldsymbol{\mu}, \boldsymbol{\gamma}, \sigma) = \\ &\quad \underbrace{\mathbf{f}^T\mathbf{d} - \text{trace}(\boldsymbol{Y}^T\boldsymbol{K}\boldsymbol{Y})}_{\text{the objective}} \\ &\quad - \lambda_1(\text{trace}(\boldsymbol{Y}^T\boldsymbol{W}^{-1}\boldsymbol{Y}) - k) \\ &\qquad + \frac{\sigma}{2}(\text{trace}(\boldsymbol{Y}^T\boldsymbol{W}^{-1}\boldsymbol{Y}) - k)^2 \\ &\quad - \boldsymbol{\mu}^T(\boldsymbol{Y}\boldsymbol{Y}^T\mathbf{e} - \boldsymbol{W}\mathbf{f}) \\ &\qquad + \frac{\sigma}{2}(\boldsymbol{Y}\boldsymbol{Y}^T\mathbf{e} - \boldsymbol{W}\mathbf{f})^T(\boldsymbol{Y}\boldsymbol{Y}^T\mathbf{e} - \boldsymbol{W}\mathbf{f}) \\ &\quad - \lambda_2(\mathbf{e}^T\mathbf{f} - (1+\alpha)n) + \frac{\sigma}{2}(\mathbf{e}^T\mathbf{f} - (1+\alpha)n)^2 \\ &\quad - \boldsymbol{\gamma}^T(\mathbf{f} - \mathbf{g} - \mathbf{s}) + \frac{\sigma}{2}(\mathbf{f} - \mathbf{g} - \mathbf{s})^T(\mathbf{f} - \mathbf{g} - \mathbf{s}) \\ &\quad - \lambda_3(\mathbf{e}^T\mathbf{g} - (1-\beta)n - r) \\ &\qquad + \frac{\sigma}{2}(\mathbf{e}^T\mathbf{g} - (1-\beta)n - r)^2 \end{aligned} \tag{5}$$

At each step in the augmented Lagrangian solution framework, we solve the following subproblem:

$$\begin{array}{lll} \text{minimize} & \mathcal{L}_A(\boldsymbol{Y}, \mathbf{f}, \mathbf{g}, \mathbf{s}, r; \lambda, \boldsymbol{\mu}, \boldsymbol{\gamma}, \sigma) & \\ \text{subject to} & Y_{ij} \geq 0, \mathbf{s} \geq 0, r \geq 0, & \\ & 0 \leq \mathbf{f} \leq k\mathbf{e}, 0 \leq \mathbf{g} \leq 1. & \end{array} \tag{6}$$

We use a limited-memory BFGS with bound constraints algorithm [9] to minimize the subproblem with respect to the variables $\boldsymbol{Y}$, $\mathbf{f}$, $\mathbf{g}$, $\mathbf{s}$ and r. This requires computation of the gradient of $\mathcal{L}_A$ with respect to the variables. We determine and validate an analytic form for the gradient in Appendix B. In Section 6.1, we provide evidence that our optimization procedure is correctly implemented. Those experiments also show that we achieve the same objective func-

Table 2: Comparison of SDP and LRSDP (objective value and run time). The small differences between the objective values are the result of differences in solution tolerances and precision in the sub-problems.

		Objective value		Run time	
		SDP	LRSDP	SDP	LRSDP
dolphins	k=2, α=0.2, β=0	-1.968893	-1.968329	107.03 seconds	2.55 seconds
	k=2, α=0.2, β=0.05	-1.969080	-1.968128	56.99 seconds	2.96 seconds
	k=3, α=0.3, β=0	-2.913601	-2.915384	160.57 seconds	5.39 seconds
	k=3, α=0.3, β=0.05	-2.921634	-2.922252	71.83 seconds	8.39 seconds
les miserables	k=2, α=0.2, β=0	-1.937268	-1.935365	453.96 seconds	7.10 seconds
	k=2, α=0.3, β=0	-1.949212	-1.945632	447.20 seconds	10.24 seconds
	k=3, α=0.2, β=0.05	-2.845720	-2.845070	261.64 seconds	13.53 seconds
	k=3, α=0.3, β=0.05	-2.859959	-2.859565	267.07 seconds	19.31 seconds

Algorithm 1 Rounding $\boldsymbol{Y}$ to a binary matrix $\boldsymbol{U}$

Input: $\boldsymbol{Y}$, $\boldsymbol{W}$, $\mathbf{f}$, $\mathbf{g}$, α, β
Output: $\boldsymbol{U}$

```
Update Y = W^{-1} Y
Set D to be the largest (n − βn) coordinates of g
for each entry i in D do
    Set S to be the top ⌊f_i⌋ entries in Y(i, :)
    Set U(i, S) = 1 /* Assign i to S */
end for
Set f̄ = f − ⌊f⌋
Set R to be the largest entries in f̄
for each entry i in R do
    Pick a cluster ℓ where Y(i, ℓ) is the maximun over all clusters where i is not currently assigned
    Set U(i, ℓ) = 1
end for
```

tion values as the convex formulation (3) in a small fraction of the time.

4.2 Rounding procedure

Solutions from the the LRSDP method are real-valued. We need to convert $\boldsymbol{Y}$ into a binary assignment matrix $\boldsymbol{U}$ through a rounding procedure. Both the vectors $\mathbf{f}$ and $\mathbf{g}$ provide important information about the solution. Namely, $\mathbf{f}$ gives us a good approximation to the number of clusters each data point is assigned to, and $\mathbf{g}$ indicates which data points are not assigned to any cluster.

The procedure we use for rounding solutions $\boldsymbol{Y}$ that arise when we run LRSDP on a unweighted kernel matrix $\boldsymbol{K}$ is given by Algorithm 1. It uses the largest $n - \beta n$ entries of the vector $\mathbf{g}$ to determine the set of nodes to assign first. Each data point i is assigned to $\lfloor f_i \rfloor$ clusters based on the values in the ith row of $\boldsymbol{Y}$. The remaining assignments are all based on the largest residual elements in $\mathbf{f} - \lfloor \mathbf{f} \rfloor$.

For our experiments with overlapping community detection, we found the following simple alternative rounding strategy more successful. Select the top $(1 + \alpha)n$ entries in $\boldsymbol{W}^{-1}\boldsymbol{Y}$ as the clustering assignment.

4.3 Practical improvements

Finally, we describe a set of practical improvements for our method. These are designed to accelerate the convergence of the augmented Lagrangian framework by moving it closer to a point that satisfies the constraints and is nearly optimal. They are designed based on commonly used strategies in the relax and round approach to discrete optimization problems.

Final rounding. At the conclusion of our rounding procedure, we have an assignment of points to clusters. We then use that as the initial cluster assignments for the iterative NEO-K-Means procedure from [30]. Since that procedure has monotone convergence behavior, this can only improve the solution.

Initialization. We run the iterative NEO-K-Means algorithm multiple times and use the result with the best objective function value as the initialization to LRSDP. For problems over a few hundred data points, this procedure results in faster convergence and better final solutions.

Sampling. For vector datasets without feature maps, we found that first using LRSDP on sampling 10% of the data points, then using this LRSDP solution as an initialization of the iterative algorithm produces similarly good results as using LRSDP on all the data points while taking significantly less time.

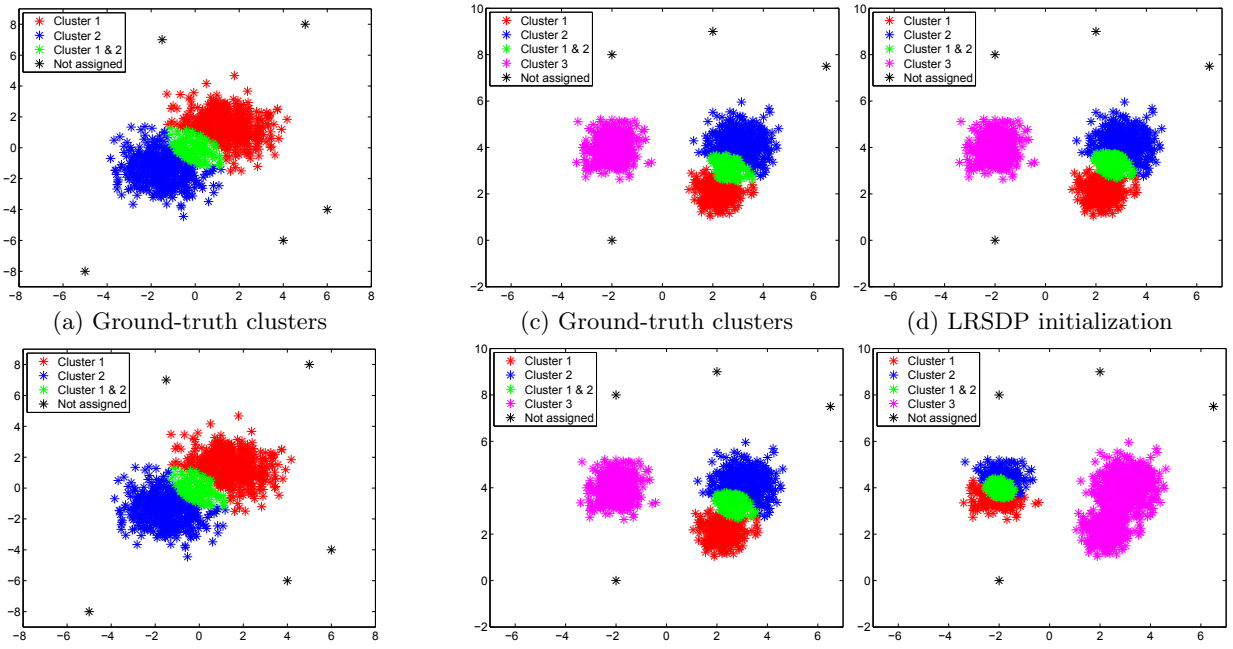
Hierarchical results. For overlapping community detection on large graph data (e.g., the HepPh and AstroPh datasets we show later), we apply a two-level hierarchical clustering. In the first level, we use LRSDP with $k' = \sqrt{k}$, $\alpha' = \sqrt{1+\alpha} - 1$ and unchanged β, then in the second level, we run LRSDP with k', α' and $\beta' = 0$ for each cluster at level 1. These parameter settings produce a final assignment result with a total of $(1 + \alpha)n$ assignments in k clusters.

5. RELATED WORK

This manuscript is most strongly related to convex relaxations of the k-means objective [18] and related SDP formulations of k-means [27, 28]. For instance, [18] employs the same general strategy of using a low-rank factorization of the SDP for k-means in concert with an augmented Lagrangian solver for the resulting nonlinear optimization problem. Even more generally, our work fits into the broad setting of convex relaxations of clustering problems including normalized cut objectives [33].

Recently, there was a proposal for a different type of convex clustering method [22, 16] which is also based on k-means. The key difference is that these relaxations model a centroid point for each data point and then attempt to penalize differences among the centroids. It is related to the lasso and the fused lasso procedures. As a convex optimization problem, it suffers the same issues as the existing SDP relaxations of k-means, namely, a quadratic number of variables to optimize.

Using augmented Lagrangian methods to solve low-rank factorizations of SDP solutions has a long history of delivering successful performance when the data arise from graphs. For instance, [6] originally proposed this idea for the MAX-



(a) Ground-truth clusters (c) Ground-truth clusters (d) LRSDP initialization

(b) k-means/LRSDP initialization (e) Success of k-means initialization (f) Failure of k-means initialization

Figure 2: The output of NEO-K-Means algorithm with two different initialization methods on two synthetic datasets. (a) & (b) On a simple dataset, NEO-K-Means can easily recover the ground-truth clusters with k-means or LRSDP initialization. (c)–(f) LRSDP initialization allows the NEO-K-Means algorithm to consistently produce a reasonable clustering structure whereas k-means initialization sometimes (4 times out of 10 trials) leads to a failure in recovering the underlying clustering structure.

CUT and minimum bisection SDPs. Later, similar ideas were used to address key weaknesses in spectral clustering [20] on power-law graphs.

6. EXPERIMENTS

We begin by validating our implementation and comparing our solutions against the global optima from the CVX program. We then show the effectiveness of LRSDP as an initialization method of the iterative NEO-K-Means algorithm [30] which is a simple greedy algorithm designed for optimizing the NEO-K-Means objective function. Finally, we show experimental results on vector and graph clustering problems by comparison with state-of-the-art clustering and community detection methods.

6.1 Algorithmic validation

We measure the objective function values produced by LRSDP compared with the convex formulation of the problem and solved by CVX. We consider two graph clustering problems using 'dolphins' [25] and 'les miserables' [17] datasets. The 'dolphins' network represents frequent associations between 62 dolphins (there are 159 undirected edges in the network), and 'les miserables' network represents the co-appearance of characters in the novel Les Miserables (there are 77 nodes and 254 edges). Table 2 shows the results. We try a set of different configurations with k, α, and β. We compare the run time of CVX solver and LRSDP and find that LRSDP is roughly an order of magnitude faster than CVX. In Table 2, we report the objective values before the relaxed solution is rounded to a discrete assignment solution to precisely measure how much our solution is different from the solution returned by CVX. We can see that the objective values returned from CVX and returned from our LRSDP solver are essentially identical—they are different in light of the solution tolerances given by the methods. *In these cases, then, we are successful in finding a globally optimal solution.*

6.2 Motivating example

Now, we show how we can exploit the benefit of LRSDP by using it as an initialization of the simple iterative NEO-K-Means algorithm. We consider two synthetic datasets shown in Figure 2(a) & Figure 2(c). In these datasets, green data points indicate the overlapped region between clusters, and black data points indicate outliers which are not supposed to belong to any cluster. The first dataset was considered in [30]. We run the iterative NEO-K-Means algorithm on these datasets with two different initialization methods: k-means and LRSDP. On a simpler dataset, Figure 2(a), we observe that the NEO-K-Means can always recover the underlying clustering structure regardless of the initialization methods. However, on Figure 2(c), we observe the advantages of LRSDP over the k-means initialization. When we use the LRSDP initialization, the NEO-K-Means always yields a similar clustering structure as the ground-truth clusters as shown in Figure 2(d). On the other hand, when the k-means initialization is used, the NEO-K-Means fails to recover the underlying clustering structure 4 times out of 10 trials as shown in Figure 2(f). Thus, we see that on more complicated datasets, the dangers of bad initialization and being stuck in local minima become clearer, and LRSDP provides a

Table 3: Comparison of NEO-K-Means objective function values.

	yeast			music			scene		
	worst	best	avg.±std.	worst	best	avg.±std.	worst	best	avg.±std.
kmeans+neo	9611	9495	9549 ± 51	87779	70158	77015 ± 7658	18905	**18745**	18806 ± 66
lrsdp+neo	**9440**	9280	9364 ± 60	**82323**	**70157**	75923 ± 5936	18904	18759	18811 ± 58
slrsdp+neo	9471	**9231**	9367 ± 90	82336	70159	75926 ± 5940	**18895**	18760	18810 ± 55

Table 4: F_1 scores on real-world vector datasets.

		moc	*esp*	*isp*	*okm*	kmeans+neo	lrsdp+neo	slrsdp+neo
yeast	worst	-	0.274	0.232	0.311	0.356	**0.390**	0.369
	best	-	0.289	0.256	0.323	0.366	**0.391**	**0.391**
	avg.±std.	-	0.284±0.006	0.248±0.010	0.317±0.004	0.360±0.004	0.391±0.001	0.382±0.011
music	worst	0.530	0.514	0.506	0.524	0.526	0.537	**0.541**
	best	0.544	0.539	0.539	0.531	0.551	**0.552**	**0.552**
	avg.±std.	0.538±0.006	0.526±0.011	0.517±0.013	0.527±0.003	0.543±0.011	0.545±0.008	0.547±0.005
scene	worst	0.466	0.569	0.586	0.571	0.597	**0.610**	0.605
	best	0.470	0.582	0.609	0.576	**0.627**	0.614	0.625
	avg.±std.	0.467±0.002	0.575±0.005	0.598±0.010	0.573±0.002	0.610±0.015	0.613±0.002	0.613±0.008

Table 5: Real-world vector datasets.

	n	dim.	$\lvert\bar{\mathcal{C}}\rvert$	k
yeast	2,417	103	731.5	14
music	593	72	184.7	6
scene	2,407	294	430.8	6

more stable initialization, which enables the NEO-K-Means algorithm to consistently produce a reasonable clustering.

6.3 Data clustering

We show some experimental results on real-world vector datasets. We use three multi-label datasets which we get from [1]. Table 5 presents some basic statistics of these datasets ('dim.' denotes the dimensionality of the vectors and $|\bar{\mathcal{C}}|$ denotes the average size of the ground-truth clusters). The 'music' dataset [29] consists of a set of feature vectors extracted from 593 different music songs. In this dataset, each song is labelled by emotions presented in the song, e.g., happy, surprised, relaxing, etc. Since several different emotions can be expressed in a song, a song can have more than one label. The 'scene' dataset [5] is a set of scene image feature vectors. Each image can be labelled by their scenes, e.g., beach, sunset, mountain, and an image can contain more than one scene. The 'yeast' dataset [13] is from a biology domain. This dataset is a set of feature vectors constructed based on micro-array expression data and phylogenetic profiles of genes. Each gene belongs to multiple functional classes, so each gene can have multiple labels. On these datasets, we treat each label as a ground-truth cluster.

To see the effectiveness of our LRSDP method, we compare LRSDP using a final iterative NEO-K-Means improvement step. This method is denoted by 'lrsdp+neo'. Also, we used the sampling method with 10% of the data points. This method is denoted by 'slrsdp+neo'. We compare these LRSDP approaches with the iterative NEO-K-Means initialized by the traditional k-means (denoted by 'kmeans+neo'). We run each method five times, and Table 3 shows the best, worst, average, and the standard deviation of the NEO-K-Means objective function values. Within all these methods, α and β values are automatically detected (see [30] for details). A lower objective value indicates a better clustering. We can see that there is a significant difference in the objective value between 'kmeans+neo' and LRSDP methods ('lrsdp+neo' and 'slrsdp+neo') on 'yeast' and 'music' datasets. By using the LRSDP solution as the initialization of the iterative algorithm, we can achieve a better objective function value for two of the datasets. This implies that LRSDP is effective in optimizing the NEO-K-Means objective, and thus provides a good initialization of the iterative algorithm. We note that the benefit of LRSDP on 'scene' dataset is not significant, but we also note that on this dataset, the average behavior of all methods is roughly the same. In this case, the overlaps among the ground-truth clusters are very small (the ground-truth α is 0.074) which implies that the traditional k-means should be a highly accurate initialization.

We also compare the clustering performance with other state-of-the-art clustering methods including model-based overlapping clustering [4], denoted by *moc*, explicit/implicit sparsity constrained clustering [24], denoted by *esp*, and *isp*, respectively, and overlapping k-means [10], denoted by *okm*. All these clustering methods are initialized by k-means, and executed five times. To see the clustering performance, we compute the F_1 score which measures the matching between algorithmic solutions and the ground-truth clusters (see [30] or [31] for details about how we compute the F_1 score). Higher F_1 scores indicate improved matches with the ground-truth clusters. Table 4 shows F_1 scores of each algorithm on the real-world datasets. On the 'yeast' dataset, *moc* produces 13 empty clusters and one cluster which contains all the data points, so we cannot report F_1 score of *moc* on this dataset. We first note that the NEO-K-Means-based methods ('kmeans+neo', 'lrsdp+neo', and 'slrsdp+neo') are consistently better than the other clustering methods; *and, the LRSDP methods are able to achieve better F_1 scores than the other methods.*

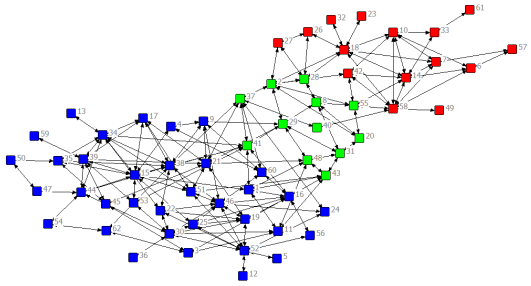


Figure 3: Visualization of the clustering result of LRSDP on 'dolphins' network. Blue nodes only belong to cluster 1, red nodes only belong to cluster 2, and green nodes belong to both of the clusters.

Table 6: Real-world network datasets.

	No. of vertices	No. of edges
Facebook1	348	2,866
Facebook2	756	30,780
HepPh	11,204	117,619
AstroPh	17,903	196,972

6.4 Overlapping community detection

The iterative NEO-K-Means method and our new LRSDP method can both be used for overlapping community detection because optimizing the NEO-K-Means objective function corresponds to optimizing an extended version of normalized cut [30]. To see whether LRSDP produces a reasonable clustering structure on graphs, we visualize the clustering result of LRSDP (k=2, α=0.2, β=0) on the 'dolphins' network [25] in Figure 3. There are two clusters where green nodes indicate the overlapped region (blue and green nodes form one cluster, and red and green nodes form the other cluster). Notice that the green nodes have many interactions with both of the clusters, which shows that LRSDP produces a plausible solution aligned with an intuitive clustering structure.

Next, we consider real-world networks from [21]. We use four different networks which are summarized in Table 6. Facebook1 and Facebook2 are social networks, and HepPh and AstroPh are collaboration networks. To run LRSDP on the two large networks, HepPh and AstroPh, we use a hierarchical clustering which we discussed in Section 4.3. Table 7 shows the comparison of the average normalized cut between the multilevel NEO-K-Means algorithm [30] and LRSDP. The multilevel NEO-K-Means (denoted by 'multilevel neo' or 'm-neo') is a variation of the iterative NEO-K-Means algorithm where the graph clustering problem is solved at multiple scales. We also use the multilevel NEO-K-Means as the final improvement step of LRSDP as we briefly discussed in Section 4.3. We see that LRSDP achieves the lower normalized cut than the multilevel NEO-K-Means, which indicates that LRSDP is beneficial to optimizing the objective function. Within these methods, we set k=32, α=3, β=0 on Facebook networks. On large networks, we determine α and β values based on the statistics of the output of *nise* method [31].

We also compare with other state-of-the-art overlapping community detection methods including *demon* [11], *bigclam* [34], *oslom* [19], and *nise* [31]. Let us first note that the runtime of LRSDP is competitive with other state-of-the-art approaches. For example, on the HepPh network with k=100, LRSDP took 18 minutes whereas *oslom* method took 19 minutes and *bigclam* method took 11 minutes. On the other hand, the multilevel NEO-K-Means algorithm completed in less than 10 seconds. Thus, our approaches and algorithms would be more suitable for applications where getting a high-quality clustering is more important than getting faster results. This is the case, for instance, in modern biology and neuroscience data. A recently collected network of the rat brain required "more than 4,000 hours to compile" [2]. On this time scale, the quality of the final results is paramount.

Table 7: Average normalized cut of the iterative multilevel NEO-K-Means and LRSDP

	multilevel neo	LRSDP
Facebook1	0.371	**0.279**
Facebook2	0.331	**0.223**
HepPh	0.185	**0.169**
AstroPh	0.240	**0.201**

Table 8: AUC of conductance-vs-graph coverage

	Facebook1	Facebook2	HepPh	AstroPh
bigclam	0.830	0.640	0.625	0.645
demon	0.495	0.318	0.503	0.570
oslom	0.319	0.445	0.465	0.580
nise	0.297	0.293	0.102	0.153
m-neo	0.285	0.269	0.206	0.190
LRSDP	**0.222**	**0.148**	**0.091**	**0.137**

We evaluate the quality of communities based on the conductance score which is one of the most commonly used metrics to evaluate the cohesiveness of communities. In particular, we compute the area under the curve (AUC) in a plot of conductance-vs-graph coverage. This metric was also studied in [31]. Given a community (set), the conductance of the community is defined to be the cut of the set divided by the least number of edges incident on either the set or its complement. By definition, a conductance score is always greater than or equal to the normalized cut. Given a set of algorithmic communities, we first compute the conductance score of each community, and then sort them in ascending order. We greedily take communities until a certain percentage of the graph is covered. So, in a conductance-vs-graph coverage plot, the x-axis is the graph coverage and y-axis is the maximum conductance score among the communities that we used to cover the corresponding portion of the graph. Finally, we compute the AUC score of this plot. The AUC score is normalized such that the maximum AUC is equal to one. A lower AUC score indicates a better clustering. Table 8 shows the results. We can see that LRSDP achieves the lowest AUC score across all the datasets, which implies that it produces the most coherent communities.

7. CONCLUSION

Our new convex and low-rank objective functions for non-exhaustive, overlapping clustering provide a new, principled

framework to cluster vector and graph data. When our non-convex low-rank method is optimized through an augmented Lagrangian method, it produces state-of-the-art quality results for both vector datasets as well as for the overlapping community detection problem on a graph.

We highlight a few directions for future work. First, our current rounding procedure provides no guarantees on the quality of the approximation. Weak guarantees on any rounding procedure would allow us to design approximation algorithms based on the convex formulation of the objective. Additionally, there are a variety of complex rounding schemes used in spectral clustering, e.g. [35], that may further improve our performance on more difficult problems. Second, there is a renaissance in fast alternating methods and proximal methods for convex and nearly convex objectives that arise in machine learning. We also plan to study variations on the low-rank approximation (4) that can utilize some of these techniques for even more scalability.

8. ACKNOWLEDGMENTS

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

9. REFERENCES

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

[2] Rat brains are basically wired up like miniature internets. `www.engadget.com/2015/04/09/rat-brains-are-basically-wired-up-like-miniature-internets/`.

[3] D. Arthur and S. Vassilvitskii. K-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1027–1035, 2007.

[4] A. Banerjee, C. Krumpelman, J. Ghosh, S. Basu, and R. J. Mooney. Model-based overlapping clustering. In *ACM SIGKDD International Conference on Knowledge Discovery in Data mining*, pages 532–537, 2005.

[5] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown. Learning multi-label scene classification. *Pattern Recognition*, 37, 2004.

[6] S. Burer and R. D. Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming*, 95:329–357, 2003.

[7] S. Burer and R. D. Monteiro. Local minima and convergence in low-rank semidefinite programming. *Mathematical Programming*, 103(3):427–444, 2005.

[8] R. Burt. *Structural Holes: The Social Structure of Competition*. Harvard University Press, 1995.

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

[10] G. Cleuziou. An extended version of the k-means method for overlapping clustering. In *International Conference on Pattern Recognition*, pages 1–4, 2008.

[11] M. Coscia, G. Rossetti, F. Giannotti, and D. Pedreschi. Demon: a local-first discovery method for overlapping communities. In *ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, pages 615–623, 2012.

[12] I. S. Dhillon, Y. Guan, and B. Kulis. Weighted graph cuts without eigenvectors a multilevel approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(11):1944–1957, 2007.

[13] A. Elisseeff and J. Weston. A kernel method for multi-labelled classification. In *Neural Information Processing Systems*, pages 681–687, 2001.

[14] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. `http://cvxr.com/cvx`, March 2014.

[15] M. C. Grant and S. P. Boyd. Graph implementations for nonsmooth convex programs. In *Recent Advances in Learning and Control*, volume 371 of *Lecture Notes in Control and Information Sciences*, pages 95–110. 2008.

[16] T. Hocking, J. Vert, A. Joulin, and F. R. Bach. Clusterpath: an algorithm for clustering using convex fusion penalties. In *Proceedings of the 28th International Conference on Machine Learning*, pages 745–752, 2011.

[17] D. E. Knuth. *The Stanford GraphBase: A Platform for Combinatorial Computing*. Addison-Wesley, 1993.

[18] B. Kulis, A. C. Surendran, and J. C. Platt. Fast low-rank semidefinite programming for embedding and clustering. In *International Conference on Artifical Intelligence and Statistics*, pages 235–242, 2007.

[19] A. Lancichinetti, F. Radicchi, J. Ramasco, and S. Fortunato. Finding statistically significant communities in networks. *PLOS ONE*, 6(4), 2011.

[20] K. Lang. Fixing two weaknesses of the spectral method. In *Advances in Neural Information Processing Systems*, pages 715–722, 2005.

[21] J. Leskovec. Stanford Network Analysis Project. `http://snap.stanford.edu/`.

[22] F. Lindsten, H. Ohlsson, and L. Ljung. Just relax and come clustering! a convexification of k-means clustering. Technical report, Linköpings universitet, 2011.

[23] S. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, March 1982.

[24] H. Lu, Y. Hong, W. N. Street, F. Wang, and H. Tong. International conference on data mining workshops. In *Overlapping clustering with sparseness constraints*, pages 486–494, 2012.

[25] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson. The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations: Can geographic isolation explain this unique trait? *Behavioral Ecology and Sociobiology*, 54(4):pp. 396–405, 2003.

[26] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, 2006.

[27] J. Peng. 0-1 semidefinite programming for spectral clustering: Modeling and approximation. Technical report, Advanced Optimization Laboratory, McMaster University, 2005.

check this for augmented Lagrangian.

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

[29] K. Trohidis, G. Tsoumakas, G. Kalliris, and I. P. Vlahavas. Multi-label classification of music into emotions. In *International Conference on Music Information Retrieval*, pages 325–330, 2008.

[30] J. J. Whang, I. S. Dhillon, and D. F. Gleich. Non-exhaustive, overlapping k-means. In *Proceedings of the SIAM International Conference on Data Mining*, pages 936–944, 2015.

[31] J. J. Whang, D. Gleich, and I. S. Dhillon. Overlapping community detection using seed set expansion. In *ACM International Conference on Information and Knowledge Management*, pages 2099–2108, 2013.

[32] L. F. Wu, T. R. Hughes, A. P. Davierwala, M. D. Robinson, R. Stoughton, and S. J. Altschuler. Large-scale prediction of saccharomyces cerevisiae gene function using overlapping transcriptional clusters. *Nature Genetics*, 31(3):255–265, June 2002.

[33] E. P. Xing and M. I. Jordan. On semidefinite relaxations for normalized k-cut and connections to spectral clustering. Technical Report UCB/USD-3-1265, University of California, Berkeley, 2003.

[34] J. Yang and J. Leskovec. Overlapping community detection at scale: a nonnegative matrix factorization approach. In *ACM International Conference on Web Search and Data Mining*, pages 587–596, 2013.

[35] S. X. Yu and J. Shi. Multiclass spectral clustering. In *IEEE International Conference on Computer Vision - Volume 2*, 2003.

APPENDIX

A. AUGMENTED LAGRANGIANS

The augmented Lagrangian framework is a general strategy to solve nonlinear optimization problems with equality constraints. We briefly review a standard textbook derivation for completeness [26]. Consider a general problem:

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & c_i(\mathbf{x}) = 0, \quad i = 1, \ldots, m \\ & \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}. \end{aligned} \tag{7}$$

The augmented Lagrangian for this problem involves a set of Lagrange multipliers λ_i to estimate the influence of each constraint on the objective as well as a quadratic penalty to satisfy the nonlinear constraints. It is defined as

$$\mathcal{L}_A(\mathbf{x}; \lambda, \sigma) = f(\mathbf{x}) - \sum_{i=1}^{m} \lambda_i c_i(\mathbf{x}) + \frac{\sigma}{2} \sum_{i=1}^{m} c_i^2(\mathbf{x}).$$

An augmented Lagrangian algorithm iteratively proceeds from an arbitrary starting point to a local solution of (7). At each step, a bound-constrained solver minimizes $\mathcal{L}_A$ over $\mathbf{x}$ subject to $\mathbf{l} \leq \mathbf{x} \leq \mathbf{u}$. Based on an approximate solution, it adjusts the Lagrange multipliers λ and may update the penalty parameter σ. See Algorithm 17.4 in Nocedal and Wright [26] for a standard strategy to adjust the multipliers, penalty, and tolerances for each subproblem.

We use the L-BFGS-B procedure [9] to solve the subproblem. This requires both a subroutine to evaluate the function and the gradient vector.

B. GRADIENTS FOR NEO-LR

We now describe the analytic form of the gradients for the augmented Lagrangian of the NEO-LR objective and a brief validation that these are correct. Consider the augmented Lagrangian (5). The gradient has five components for the five sets of variables: $\boldsymbol{Y}$, $\mathbf{f}$, $\mathbf{g}$, $\mathbf{s}$ and r:

$$\begin{aligned} &\nabla_{\boldsymbol{Y}} \mathcal{L}_A(\boldsymbol{Y}, \mathbf{f}, \mathbf{g}, \mathbf{s}, r; \lambda, \mu, \gamma, \sigma) = \\ &\quad - 2\boldsymbol{K}\boldsymbol{Y} - \mathbf{e}\mu^T\boldsymbol{Y} - \mu\mathbf{e}^T\boldsymbol{Y} \\ &\quad - 2(\lambda_1 - \sigma(\mathrm{tr}(\boldsymbol{Y}^T\boldsymbol{W}^{-1}\boldsymbol{Y}) - k))\boldsymbol{W}^{-1}\boldsymbol{Y} \\ &\quad + \sigma(\boldsymbol{Y}\boldsymbol{Y}^T\mathbf{e}\mathbf{e}^T\boldsymbol{Y} + \mathbf{e}\mathbf{e}^T\boldsymbol{Y}\boldsymbol{Y}^T\boldsymbol{Y}) - \sigma(\boldsymbol{W}\mathbf{f}\mathbf{e}^T\boldsymbol{Y} + \mathbf{e}\mathbf{f}^T\boldsymbol{W}\boldsymbol{Y}) \end{aligned}$$

$$\begin{aligned} &\nabla_{\mathbf{f}} \mathcal{L}_A(\boldsymbol{Y}, \mathbf{f}, \mathbf{g}, \mathbf{s}, r; \lambda, \mu, \gamma, \sigma) = \\ &\quad \mathbf{d} + \boldsymbol{W}\mu - \sigma(\boldsymbol{W}\boldsymbol{Y}\boldsymbol{Y}^T\mathbf{e} - \boldsymbol{W}^2\mathbf{f}) - \lambda_2\mathbf{e} + \sigma(\mathbf{e}^T\mathbf{f} - (1+\alpha)n)\mathbf{e} \\ &\quad - \gamma + \sigma(\mathbf{f} - \mathbf{g} - \mathbf{s}) \end{aligned}$$

$$\begin{aligned} &\nabla_{\mathbf{g}} \mathcal{L}_A(\boldsymbol{Y}, \mathbf{f}, \mathbf{g}, \mathbf{s}, r; \lambda, \mu, \gamma, \sigma) = \\ &\quad \gamma - \sigma(\mathbf{f} - \mathbf{g} - \mathbf{s}) - \lambda_3\mathbf{e} + \sigma(\mathbf{e}^T\mathbf{g} - (1-\beta)n - r)\mathbf{e} \end{aligned}$$

$$\nabla_{\mathbf{s}} \mathcal{L}_A(\boldsymbol{Y}, \mathbf{f}, \mathbf{g}, \mathbf{s}, r; \lambda, \mu, \gamma, \sigma) = \gamma - \sigma(\mathbf{f} - \mathbf{g} - \mathbf{s})$$

$$\nabla_{r} \mathcal{L}_A(\boldsymbol{Y}, \mathbf{f}, \mathbf{g}, \mathbf{s}, r; \lambda, \mu, \gamma, \sigma) = \lambda_3 - \sigma(\mathbf{e}^T\mathbf{g} - (1-\beta)n - r)$$

Using analytic gradients in a black-box solver such as L-BFGS-B is problematic if the gradients are even slightly incorrectly computed. To guarantee the analytic gradients we derive are correct, we use forward finite difference method to get numerical approximation of the gradients based on the objective function. We compare these with our analytic gradient and expect to see small relative differences on the order of 10^{-5} or 10^{-6}. This is exactly what Figure 4 shows.

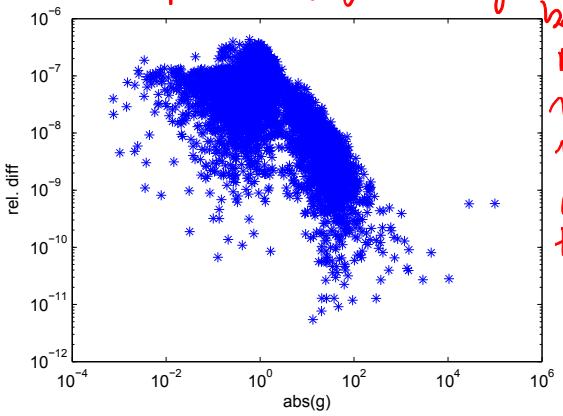


Figure 4: Finite difference comparison of gradients where $\epsilon = 10^{-6}$. This figure shows that the relative difference between the analytical gradient and the gradient computed via finite differences is small, indicating the gradient is correctly computed.