Correlation Clustering with Low-Rank Matrices

Nate Veldt
Purdue University
Mathematics Department
West Lafayette, IN
lveldt@purdue.edu

Anthony Wirth
The University of Melbourne
Computing and Information
Systems Department
Parkville, VIC, Australia
awirth@unimelb.edu.au

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

ABSTRACT

Correlation clustering is a technique for aggregating data based on qualitative information about which pairs of objects are labeled 'similar' or 'dissimilar.' Because the optimization problem is NP-hard, much of the previous literature focuses on finding approximation algorithms. In this paper we explore how to solve the correlation clustering objective exactly when the data to be clustered can be represented by a low-rank matrix. We prove in particular that correlation clustering can be solved in polynomial time when the underlying matrix is positive semidefinite with small constant rank, but that the task remains NP-hard in the presence of even one negative eigenvalue. Based on our theoretical results, we develop an algorithm for efficiently "solving" low-rank positive semidefinite correlation clustering by employing a procedure for zonotope vertex enumeration. We demonstrate the effectiveness and speed of our algorithm by using it to solve several clustering problems on both synthetic and real-world data.

Keywords

Correlation clustering; Low-rank matrices

1. INTRODUCTION

Correlation clustering is a method for partitioning a dataset based on pairwise information that indicates whether pairs of objects in the given dataset are 'similar' or 'dissimilar.' Typically correlation clustering is cast as a graph optimization problem where the nodes of a graph represent objects from the dataset. In its most basic form, the graph is assumed to be complete and unweighted, with each edge being labeled '+' or '−' depending on whether the two nodes are 'similar' or 'dissimilar.' Given this input, the objective is to partition the graph in a way that maximizes the number of agreements, where an agreement is a '+' edge that is included inside a cluster, or a '−' edge that links nodes in different clusters. An equivalent objective, though more difficult to approximate, is the goal of minimizing disagreements, i.e.,

'similar' nodes that are separated or 'dissimilar' nodes that are clustered together. A more general form of correlation clustering associates each pair of objects with not only a label but also a weight indicating how similar or dissimilar the two objects are. In this case, the goal is to maximize the weight of agreements or minimize the weight of disagreements.

One attractive property of this clustering approach is that the number of clusters formed is determined automatically by optimizing the objective function, rather than being a required input. In practice, correlation clustering has been applied in a wide variety of disciplines to solve problems such as cross-lingual link detection [28], gene clustering [6], image segmentation [15], and record linkage in natural language processing [18].

Because correlation clustering is NP-hard [5], much of the previous literature has focused on developing approximation algorithms. In this paper, we consider a new approach, exploring instances of the problem where the weighted labels can be represented by a low-rank matrix. Studying this case provides a new means for dealing with the intractability of the problem, and also allows us to apply the framework of correlation clustering in the broader task of analyzing low-dimensional datasets.

Our Contributions: In this paper we prove that correlation clustering can be solved in polynomial time when the similarity labels can be represented by a positive semidefinite matrix of low rank. We also show that the problem remains NP-hard when the underlying matrix has even one negative eigenvalue. To solve correlation clustering problems in practice, we implement an algorithm called ZoNoCC based on the randomized zonotope vertex enumeration procedure of Stinson, Gleich, and Constantine [26]. This algorithm is capable of optimally solving low-rank positive semidefinite correlation clustering. It is most useful, however, when it is truncated at a fixed number of iterations in order to quickly obtain a very good approximation – sans formal guarantee – to the optimal solution. We demonstrate the effectiveness of ZoNoCC by obtaining clusterings for both synthetic and real-world datasets, including social network datasets and search-query data for well-known computer science conferences.

2. PROBLEM STATEMENT

We begin with the standard approach to correlation clustering by considering a graph with n nodes where edges are labeled either '+' or '−'. Typically the correlation clustering objective is cast as an integer linear program in the following way. For every pair of nodes i and j we are given two

nonnegative weights, w_{ij}^+ and w_{ij}^-, which indicate a score for how similar the two nodes are and a score for how dissimilar they are respectively. Traditionally, we assume that only one of these weights is nonzero (if not, they can be adjusted so this is the case without changing the objective function by more than an additive constant). For every pair of nodes i, j we introduce a binary variable d_{ij} such that

$$d_{ij} = \begin{cases} 0 & \text{if } i \text{ and } j \text{ are clustered together}; \\ 1 & \text{if } i \text{ and } j \text{ are separated}. \end{cases}$$

In other words, $d_{ij} = 1$ indicates we have cut the edge between nodes i and j. The maximization version of correlation clustering is given by the following ILP (integer linear program). We include triangle constraints on the d_{ij} variables to guarantee that they define a valid clustering on the nodes.

$$\begin{aligned} \text{maximize} \quad & \sum_{i<j} w_{ij}^+(1 - d_{ij}) + \sum_{i<j} w_{ij}^- d_{ij}, \\ \text{subject to} \quad & d_{ij} \in \{0, 1\}, \\ & d_{ik} \le d_{ij} + d_{jk} \text{ for all } i, j, k. \end{aligned} \quad (1)$$

The first term counts the weight of agreements from clustering similar nodes together, and the second counts the weight of agreements from dissimilar nodes that are clustered apart.

For convenience, we encode the weights of a correlation clustering problem into a matrix $\boldsymbol{A}$ by defining $A_{ij} = w_{ij}^+ - w_{ij}^-$. We think of $\boldsymbol{A}$ as the adjacency matrix of a graph that has both positive and negative edges. We can express the objective function in terms of $\boldsymbol{A}$ as

$$\text{maximize} \quad -\sum_{i<j} A_{ij} d_{ij} + \sum_{i<j} w_{ij}^+.$$

Since the second term is only a constant, to solve this problem optimally, we can focus on just solving this ILP:

$$\begin{aligned} \text{maximize} \quad & -\sum_{i<j} A_{ij} d_{ij} \\ \text{subject to} \quad & d_{ij} \in \{0, 1\}, \\ & d_{ik} \le d_{ij} + d_{jk} \text{ for all } i, j, k. \end{aligned} \quad (2)$$

We can provide an alternative formulation of the correlation clustering objective by introducing an indicator vector $\mathbf{x}_i \in \{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \ldots, \mathbf{e}_n\}$ for each node i, where $\mathbf{e}_j$ is the jth standard basis vector in $\mathbb{R}^n$. This indicates which cluster node i belongs to. Unless each node ends up in its own singleton cluster, some of these basis vectors will be unused. We can then make the substitution $d_{ij} = 1 - \mathbf{x}_i^T \mathbf{x}_j$, since $\mathbf{x}_i^T \mathbf{x}_j$ will be 1 if both nodes are in the same cluster but will be 0 otherwise. After making this substitution and dropping a constant term in the objective, the problem becomes

$$\begin{aligned} \text{maximize} \quad & \sum_{i<j} A_{ij} \mathbf{x}_i^T \mathbf{x}_j \\ \text{subject to} \quad & \mathbf{x}_i \in \{\mathbf{e}_1, \ldots, \mathbf{e}_n\} \text{ for all } i = 1, \ldots, n. \end{aligned} \quad (3)$$

3. THEORETICAL RESULTS

In this section, we present new results on the complexity of correlation clustering under low-rank assumptions. In particular, we prove the problem remains NP-hard when the underlying matrix has even one negative eigenvalue. We are more concerned, however, with solving correlation clustering on low-rank positive semidefinite adjacency matrices, in which case we give a polynomial time solution. This scenario is analogous to results for other related optimization problems that admit polynomial-time solutions for low-rank positive semidefinite input matrices [10, 17], and is also a particularly natural assumption for the correlation clustering objective.

For example, if the adjacency matrix represents a correlation matrix (i.e., each entry is the Pearson correlation coefficient between two random variables), the input is already positive semidefinite. So taking a low-rank approximation will yield the type input matrix studied here; we give two examples of correlation clustering on a correlation matrix in Section 5.

Even when the input is not a correlation matrix, we note that the correlation clustering objective does not depend on the diagonal of the input matrix, so we are able to shift the diagonal entries until the matrix is positive definite before taking a low-rank approximation. Though the quality of the approximation will vary depending on how much the diagonal needs to be increased, this provides a means to apply our methods to get an approximate solution for every full-rank dataset.

3.1 Positive Semidefinite CC

The simplest case to consider is when $\boldsymbol{A}$ is rank-1 with one positive eigenvalue. Because $\boldsymbol{A}$ is symmetric, we can express it as $\boldsymbol{A} = \mathbf{v}\mathbf{v}^T$ for some $\mathbf{v} \in \mathbb{R}^n$. In this case a perfect clustering always exists and is easy to find: one cluster includes all nodes with negative entries in $\mathbf{v}$, while the other includes those with positive entries. Nodes i and j are similar if and only if $A_{ij} > 0$, which is true if and only if entries i and j of the vector $\mathbf{v}$ have the same sign. So this simple two clustering agrees perfectly with the similarity labels.

In fact, the rank-1 posotive semidefinite correlation clustering problem is equivalent to maximizing a quadratic form $\mathbf{x}^T \boldsymbol{A} \mathbf{x}$ in binary variables $\mathbf{x} \in \{-1, 1\}^n$, under the assumption that $\boldsymbol{A}$ is rank-1. This maximization problem can be solved in polynomial time for every fixed low rank, d [17]. While this gives us a nice result for correlation clustering on rank-1 matrices, it does not generalize to higher ranks as it only can partition a graph into exactly *two* clusters.

If the matrix $\boldsymbol{A}$ is positive semidefinite but of rank $d > 1$, there is no guarantee that a perfect partitioning exists, and the optimal clustering may have more than two clusters. We still begin by expressing $\boldsymbol{A}$ in terms of low rank factors, i.e., $\boldsymbol{A} = \boldsymbol{V}\boldsymbol{V}^T$ for some $\boldsymbol{V} \in \mathbb{R}^{n \times d}$. Each node in the signed graph can now be associated with one of the row vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n \in \mathbb{R}^{d \times 1}$ of $\boldsymbol{V}$. The similarity scores between nodes i and j are given by $A_{ij} = \mathbf{v}_i^T \mathbf{v}_j$, so we can view this version of correlation clustering as the following vector partitioning problem. Separate n points, or vectors, in $\mathbb{R}^d$ based on similarity scores given by dot products of the vectors:

THEOREM 1. *If $\boldsymbol{A} = \boldsymbol{V}\boldsymbol{V}^T$ for $\boldsymbol{V} \in \mathbb{R}^{n \times d}$, then problem (3) can be solved by partitioning the row vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots \mathbf{v}_n \in \mathbb{R}^{d \times 1}$ of $\boldsymbol{V}$ into $d + 1$ clusters $\{C_1, C_2, \ldots, C_{d+1}\}$ to solve*

$$\text{maximize} \quad \sum_{i=1}^{d+1} \|S_i\|_2^2, \quad (4)$$

*where we refer to the vector $S_i = \sum_{\mathbf{v} \in C_i} \mathbf{v}$ as the **sum point** of the ith cluster (in an empty cluster, defined to be the zero vector).*

PROOF. We will show in two steps that when $\boldsymbol{A} = \boldsymbol{V}\boldsymbol{V}^T$, the clustering that maximizes objective (3) also maximizes objective (4). The first step is to prove that (3) is equivalent to maximizing the sum of squared norms of sum points, where the maximization is taken over every possible clustering. Second, we show that the objective function is maximized by a clustering with $d + 1$ or fewer clusters.

By doubling the objective function in (3) and adding the constant $\sum_{i=1}^n \mathbf{v}_i^T \mathbf{v}_i$, we obtain a related objective function that is maximized by the same clustering:

$$2\sum_{i<j} \mathbf{v}_i^T \mathbf{v}_j (\mathbf{x}_i^T \mathbf{x}_j) + \sum_{i=1}^n \mathbf{v}_i^T \mathbf{v}_i = \sum_{i=1}^n \sum_{j=1}^n \mathbf{v}_i^T \mathbf{v}_j (\mathbf{x}_i^T \mathbf{x}_j). \quad (5)$$

Since $\mathbf{x}_i$ and $\mathbf{x}_j$ are indicator vectors, identifying which clusters nodes i and j belong to, the right-hand side of equation (5) only counts the product $\mathbf{v}_i^T \mathbf{v}_j$ when $\mathbf{x}_i^T \mathbf{x}_j = 1$. So we are restricting our attention to inner products between vectors that belong to the same cluster. The contribution to the objective from cluster C_k is

$$\sum_{a\in C_k} \sum_{b\in C_k} \mathbf{v}_a^T \mathbf{v}_b = S_k^T S_k = ||S_k||_2^2 \,.$$

Summing over all clusters completes step one of the proof.

To see that the number of clusters is bounded, observe that in the optimal clustering all the sum points must have pairwise non-positive dot products. Otherwise, there would exist distinct clusters C_i and C_j with $S_i^T S_j > 0$, and therefore

$$(S_i + S_j)^T (S_i + S_j) = S_i^T S_i + 2S_i^T S_j + S_j^T S_j > S_i^T S_i + S_j^T S_j \,.$$

Hence we could get a better clustering by combining C_i and C_j. Now, if there were an optimal clustering with two sum points that are orthogonal – $S_i^T S_j = 0$, we could combine the two clusters without changing the objective score. Therefore, among all optimal clusterings, the one with the fewest clusters has the property that all sum points have pairwise negative dot products. The bound of $d+1$ clusters then follows from the fact that the maximum number of vectors in $\mathbb{R}^d$ with pairwise negative inner products is $d+1$ (Lemma 8 of Rankin [25]). □

It is worth noting that despite a significant difference in motivation, our new objective function (4) is nearly identical to one used by Newman as a means to approximately solve maximum modularity clustering [19]. This is an interesting new connection between two clustering techniques that were not previously known to be related. Newman and Zhang's work contains further information on modularity [29, 19].

The importance of Theorem 1 is that it expresses the low-rank positive semidefinite correlation clustering problem as a convex functional on sums of vectors in $\mathbb{R}^d$. Our problem is therefore an instance of the well-studied vector partition problem [20, 13]. Onn and Schulman showed that for dimension d and a fixed number of clusters p, this problem can be solved in polynomial time by exploring the $O(n^{d(p-1)-1})$ vertices of a $d(p-1)$-dimensional polytope called the *signing zonotope*.

COROLLARY 1. *Correlation clustering with rank-d positive semidefinite matrices (PSD-CC) is a special case of the vector partition problem with $d+1$ clusters, and is therefore solvable in polynomial time.*

We later show how to construct a polynomial-time algorithm for PSD-CC by reviewing the results of Onn and Schulman [20]. Before this, we observe a second theorem, which highlights an important geometric feature satisfied by the optimal clustering. It provides a first intuition as to how we can solve the problem in polynomial time.

THEOREM 2. *In the clustering $\mathbf{C}_{opt}$, which maximizes (4), the n row vectors of $\mathbf{V}$ will be separated into distinct convex cones that intersect only at the origin. More precisely, if vectors $\mathbf{v}_{x_1}, \mathbf{v}_{x_2}, \ldots, \mathbf{v}_{x_k}$ are all in the same cluster C_x in $\mathbf{C}_{opt}$, and $\mathbf{v}_y \in \mathbb{R}^d$ is another row vector that satisfies $\mathbf{v}_y = \sum_{i=1}^k c_i \mathbf{v}_{x_i}$ for $c_i \in \mathbb{R}_0^+$, then $\mathbf{v}_y$ is also in cluster C_x.*

PROOF. First notice that in $\mathbf{C}_{opt}$, every vector $\mathbf{v}$ must be more similar to its own sum point than to any sum point of a different cluster. To see this, assume that $\mathbf{v}$ is in cluster C_i with sum point S_i, but $\mathbf{v}$ is more similar to another sum point S_j, i.e.

$$\mathbf{v}^T S_i < \mathbf{v}^T S_j.$$

The contribution to the objective from the two sum points is $S_i^T S_i + S_j^T S_j$. If we move $\mathbf{v}$ from cluster C_i to cluster C_j, the contribution to the objective for the two new clusters is

$$(S_i - \mathbf{v})^T (S_i - \mathbf{v}) + (S_j + \mathbf{v})^T (S_j + \mathbf{v}) =$$
$$S_i^T S_i - 2\mathbf{v}^T S_i + \mathbf{v}^T \mathbf{v} + S_j^T S_j + 2\mathbf{v}^T S_j + \mathbf{v}^T \mathbf{v}$$

which is a higher score since $\mathbf{v}^T S_i < \mathbf{v}^T S_j$, contradicting the optimality of the first clustering. So in the optimal clustering every point is more similar to its own sum point than any other sum point.

Given this first observation we will now prove the main result of the theorem by contradiction. Assume that we have k points $\mathbf{v}_{x_1}, \mathbf{v}_{x_2}, \ldots, \mathbf{v}_{x_k}$ that in $\mathbf{C}_{opt}$ are in cluster C_x with sum point S_x. Let $\mathbf{v}_y$ be another point in the dataset such that $\mathbf{v}_y = \sum_{i=1}^k c_i \mathbf{v}_{x_i}$ where $c_i > 0$ for $i = 1, 2, \cdots, k$, and assume that $\mathbf{v}_y$ is in a different cluster C_y that has sum point S_y. By our first observation, every point in C_x must be more similar to S_x than S_y, so for $1 \le i \le k$ we have that $\mathbf{v}_{x_i}^T S_x > \mathbf{v}_{x_i}^T S_y$, which implies $c_i \mathbf{v}_{x_i}^T S_x > c_i \mathbf{v}_{x_i}^T S_y$, for any positive scalar c_i. It follows that

$$\mathbf{v}_y^T S_x = \sum_{i=1}^k c_i \mathbf{v}_{x_i}^T S_x > \sum_{i=1}^k c_i \mathbf{v}_{x_i}^T S_y = \mathbf{v}_y^T S_y.$$

This indicates that $\mathbf{v}_y$ is more similar to S_x than to the sum point of the cluster to which is belongs, which is a contradiction.

□

Theorem 2 implies that we can find an optimal clustering in polynomial time by checking every possible partitioning of the vectors into convex cones. By Theorem 2.7 of Klee [16], every pair of cones can be separated by a hyperplane through the origin. Furthermore, Cover [9] proved that for every set of n points in $\mathbb{R}^d$, there are $O(n^{d-1})$ such hyperplanes that split the points into two groups. This means that to cluster the points into $d+1$ convex cones, we must choose $\binom{d+1}{2}$ of the $O(n^{d-1})$ hyperplanes so that each distinct pair of clusters is separated by a hyperplane. This gives us a total of $O(n^{(d-1)\binom{d+1}{2}})$ ways to cluster the points into $d+1$ convex cones, which we can enumerate in polynomial time. When $d = 2$, we can efficiently find the boundaries of the optimal convex cones by considering the n rays that each connect the origin to one of the n points. Since there are at most three clusters in this case, we can solve the problem by testing $O(n^3)$ triplets of points as possible separators for the clusters, each time evaluating the objective. A visualization of this process is shown in Figure 1. Though we can make this procedure very efficient for the two-dimensional case, it is significantly less efficient for dimension d greater than 2, so we resort to the methods proposed by Onn and Schulman [20], which we review in Section 4.
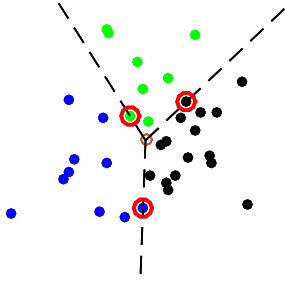
Figure 1: The three-clustering that solves a small rank-2 correlation clustering problem. Each cluster is shown by a different color, and can be delimited on each side by rays through the origin. By selecting the right combination of three points (circled in red), we will be able to find the optimal clustering.

3.2 Rank-1 Negative Semidefinite

When $\boldsymbol{A}$ has rank 1 and its nonzero eigenvalue is negative, we know $\boldsymbol{A} = -\mathbf{v}\mathbf{v}^T$ for some $\mathbf{v} \in \mathbb{R}^d$. This changes the objective (4) in three ways: the negative sign converts the maximization to a minimization, the entries of $\mathbf{v}$ are real numbers rather than row vectors, and the upper bound of $d+1$ clusters no longer applies. Solving the Rank One Negative Eigenvalue Correlation Clustering problem (RONE-CC) is therefore equivalent to

$$\min_{C,k} \ \sum_{i=1}^{k} \Big(\sum_{v \in C_i} v \Big)^2 . \qquad (6)$$

The following theorem regarding RONE-CC is analogous to a result of Papailiopoulos et al., who use a reduction from Subset Sum to prove that the Densest-k-Subgraph problem is NP-hard for input matrices of rank-1 with one negative eigenvalue [22]. Our result relies on a reduction from the related Partition problem, and accounts for the fact that in an instance of correlation clustering we must optimize over the set of partitionings with arbitrarily many clusters.

THEOREM 3. *Rank-one Negative Eigenvalue Correlation Clustering is NP-Complete.*

PROOF. Given that general correlation clustering is NP-Complete, we know that the decision version of RONE-CC must also be in NP. To show the problem is NP-hard, we can use a reduction from the Partition problem, one of the NP-hard problems listed by Garey and Johnson [11]. For this problem we are given a multiset of n positive integers and seek to partition the set into subsets of equal sum.

Consider a multiset of n positive integers. Let s be the smallest integer and B be the sum of all n integers. We assume s and B are both even–if this is not the case we can multiply all numbers in the set by two so that this assumption is satisfied. Letting $M = B/2 - s/2$, we add two copies of $-M$ to the input: now the total sum of the input integers is s. We can show that the optimal solution to RONE-CC on this input will perfectly partition the n positive integers into two subsets of equal sum if such a perfect partition exists.

Assume the positive integers can be split into two subsets of sum $B/2$. If we include exactly one copy of the $-M$ values with each of these two subsets, then each cluster sums to $B/2 - M = s/2$. The RONE-CC objective corresponding

to this two-clustering is $2(s/2)^2 = s^2/2$. We now show that every other clustering yields a worse objective value. Clearly, any clustering with one copy of $-M$ in each cluster that does not equally split the positives will have objective score $> s^2/2$. If we cluster all integers together, the sum is s, and the objective would be $s^2 > s^2/2$. If on the other hand we consider a two-clustering where both $-M$ values are in the same partition, or any clustering with more than two clusters, then there must exist some cluster with only positive integers. This cluster has sum at least s, leading to an objective of at least s^2. The best option is therefore to form two clusters, each of which contains one of the $-M$ values and a subset of the n positive integers summing to $B/2$. $\square$

4. ALGORITHMS

In this section we show how to obtain a polynomial-time algorithm for solving PSD-CC. We first review the results of Onn and Schulman [20], which establish the existence of a polynomial-time algorithm, by analyzing the properties of a d^2-dimensional polytope called the signing zonotope. We then combine this with a vertex-enumeration procedure developed by Stinson, Gleich, and Constantine [26].

4.1 Signing Zonotope

A zonotope is the linear projection of a high-dimensional hypercube into a lower-dimensional vector space. We are primarily concerned with the *signing zonotope* introduced by Onn and Schulman [20], whose vertices directly correspond to clusterings of the n vectors of a vector partition problem.

Consider a set S_V of n vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n \in \mathbb{R}^{d \times 1}$ in an instance of the vector partition problem. A *signing* of these vectors is defined to be a vector $\sigma = (\sigma_{a,b}^i) \in \{-1, 1\}^M$, where $M = n\binom{d+1}{2}$. Each entry $\sigma_{a,b}^i$ uniquely corresponds to a triplet $(\mathbf{v}_i, a, b)$, where $\mathbf{v}_i$ is one of the data points we are clustering and $1 \le a < b \le (d+1)$ are the indices for two distinct clusters in a $(d+1)$-clustering of the n vectors. If $b < a$, we define $\sigma_{a,b}^i = -\sigma_{b,a}^i$ and associate each signing with a matrix $\boldsymbol{T}_\sigma$:

$$\boldsymbol{T}_\sigma = \sum_{i=1}^{n} \sum_{1 \le a < b \le d+1} \sigma_{a,b}^i \mathbf{v}_i \cdot (\mathbf{e}_a - \mathbf{e}_b)^T \in \mathbb{R}^{d \times (d+1)}, \quad (7)$$

where $\mathbf{e}_a, \mathbf{e}_b \in \mathbb{R}^{(d+1) \times 1}$ are the ath and b^{th} standard basis vector, respectively. By construction, the row sum of $\boldsymbol{T}_\sigma$ will be the zero vector, so if we are given the first d columns of the matrix we will be able to recover the last column even if it is not given explicitly. We associate with each signing σ a vector Z_σ of length d^2 made by stacking the first d columns of $\boldsymbol{T}_\sigma$ on top of one another. Here we will refer to this vector as the Z-vector of σ. A signing σ is said to be *extremal* if its Z-vector is a vertex of the signing zonotope, which we define below. Furthermore, Onn and Schulman proved that for every vertex v of the zonotope, there exists exactly one extremal signing σ such that v is the Z-vector of σ. In other words, the extremal signings are in one-to-one correspondence with the vertices of the zonotope.

The *signing zonotope* $\mathcal{Z}$ for this instance of the vector partition problem is defined to be

$$\mathcal{Z} = \text{conv}\{Z_\sigma : \sigma \text{ is a signing of } S_V\} .$$

In other words, $\mathcal{Z}$ is the convex hull of all 2^M Z-vectors of signings of S_V. We now state two important results

established by Onn and Schulman [20] about the signing zonotope.

Theorem 4. *(Results from [20]) The following properties hold regarding the signing zonotope $\mathcal{Z}$ introduced above:*

1. *Each vertex of $\mathcal{Z}$ can be mapped to a clustering of the n vectors in S_V, where each cluster is contained in one of $d+1$ convex cones. Additionally, there exists an extremal signing that maps to the clustering which optimizes the objective of the vector partition problem.*

2. *Signing zonotope $\mathcal{Z}$ has at most $O(n^{d^2-1})$ vertices.*

All we need then to solve the vector partition problem, and hence PSD-CC, is to iterate through each extremal signing of $\mathcal{Z}$, obtain the clustering it corresponds to, and evaluate objective (4) for that clustering. At the end we output the clustering with the maximum objective value.

The procedure for associating an extremal signing with a clustering of S_V is given in Proposition 2.3 of Onn and Schulman's work [20]. This states that for all $i = 1, 2, \ldots, n$, there exists a unique index $1 \le c_i \le d+1$ such that $\sigma_{c_i,k}^i = 1$ for all $k \ne c_i$. Thus vector $\mathbf{v}_i$ belongs to cluster number c_i in the optimal clustering. With this, we are now able to state the exact runtime for solving PSD-CC.

Theorem 5. *The fixed-rank positive semidefinite correlation clustering problem can be solved in $O(n^{d^2})$ time.*

Proof. Relying on previous complexity and algorithmic results regarding zonotopes, in their Corollary 3.3, Onn and Schulman establish that the $(d+1)$-vector partition problem can be solved with $O(n^{d^2-1})$ operations and queries to an oracle for evaluating the convex objective functional [20]. We now show that it takes $O(n)$ operations to evaluate our specific oracle function for each of the $O(n^{d^2-1})$ extremal signings. In our case, the complexity of the oracle is the time it takes to evaluate summation (4) for a given extremal signing σ. Treating d as a fixed constant, this procedure involves inspecting the $M = O(n)$ entries of σ to identify a clustering, and $O(n)$ operations to add vectors in each cluster to obtain the sum points. We require only a constant number of operations to take dot products of the sum points and add the results, so the evaluation process takes $O(n)$ time, and hence the overall process $O(n^{d^2})$ time.

4.2 Practical Algorithm

Though theoretically polynomial time, the runtime given above is impractical for applications. We turn our attention to an algorithm which approximately solves the PSD-CC objective, but is much more efficient in practice. We implement a randomized algorithm for sampling vertices of a zonotope (that will eventually enumerate them all), developed by Stinson, Gleich, and Constantine [26]. When mapping a hypercube in $\mathbb{R}^M$ onto a zonotope in $\mathbb{R}^N$, the basic outline of their procedure is as follows. Form an $N \times M$ matrix $\mathbf{G}$, where each column is a generator of the zonotope, i.e., $\mathbf{G}$ is the linear map that maps the hypercube into a lower-dimensional space. Given a vector $\mathbf{x}$ drawn from a standard Gaussian distribution, compute $\mathbf{v} = \mathbf{G}\,\text{sign}(\mathbf{G}^T \mathbf{x})$, where $\text{sign}(\mathbf{u})$ returns a vector with ± 1 entries, reflecting the signs of the entries of $\mathbf{u}$. The main insight of Stinson, Gleich and Constantine is that under reasonable assumptions on

$\mathbf{G}$, $\mathbf{v}$ will be a vertex of the zonotope. One can construct the entire zonotope by generating vertices in this way, by checking whether a given vertex has been previously found, and continuing until all the vertices have been returned. In practice, it is better to just approximate the zonotope by stopping after a certain number of vertices have been found.

We alter this procedure slightly to fit our needs. Note that the generators of the signing zonotope come from outer products of the form $\mathbf{v}_i \cdot (\mathbf{e}_r - \mathbf{e}_s)^T$, for $i = 1, 2, \ldots, n$ and $1 \le r < s \le d+1$. This product gives a $d \times (d+1)$ matrix with a zero row sum, so taking the first d columns and stacking them into a vector we get one of the columns of $\mathbf{G}$. Equation (7) shows that when we form a linear combination of these generators, where the coefficients of the linear combination are entries of a signing σ, the output is exactly the Z-vector corresponding to σ. We are ultimately interested in extremal signings rather than actual zonotope vertices, so we repeatedly generate vectors $\sigma = \text{sign}(\mathbf{G}^T \mathbf{x})$. We then inspect the entries of σ and find the corresponding clustering of n vectors and thence the clustering's PSD-CC objective score (4). We do this for a very large number of randomly generated extremal signings and output the one with the highest score. We name our algorithm based on this zonotope vertex enumeration, ZonoCC, outlined in Algorithm 1.

Algorithm 1 ZonoCC

Input: rows of $\mathbf{V}$: $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n \in \mathbb{R}^d$, and $k \in \mathbb{N}$
Form generator matrix $\mathbf{G}$
Set BestClustering $\leftarrow \varnothing$, BestObjective $\leftarrow 0$
for $i = 1, 2, \ldots, k$ **do**
    1. Generate standard Gaussian $\mathbf{x} \in \mathbb{R}^M$
    2. $\sigma \leftarrow \text{sign}(\mathbf{G}^T \mathbf{x})$
    3. Determine clustering C and objective C_{obj} from σ
    if $C_{\text{obj}} >$ BestObjective **then**
        Set BestClustering $\leftarrow C$, BestObjective $\leftarrow C_{\text{obj}}$
Output: BestClustering, BestObjective

Our new algorithm is significantly faster than exploring all vertices of the zonotope. Once we have formed the $n\binom{d+1}{2}$ columns of $\mathbf{G}$, generating σ by matrix multiplication and determining the corresponding clustering both take $O(n)$ time since d is a small fixed constant. The overall runtime is therefore just $O(nk)$, where k is the number of iterations. Although ZonoCC is not guaranteed to return the optimal clustering, Stinson, Gleich, and Constantine prove that with high probability the zonotope vertices that are generated will tend to be those which most affect the overall shape of the zonotope [26]. We expect such extremal vertices of the zonotope to be associated with extremal clusterings of the n data points, i.e., clusterings with high objective score.

5. NUMERICAL EXPERIMENTS

In this section, we demonstrate the performance of Zono-CC in a variety of clustering applications. In order to understand how its behavior depends on the rank, as well as the size of the problem, we begin by illustrating the performance of Zono-CC on synthetic datasets that are low-dimensional by construction. We then study correlation clustering in two real-world scenarios: (i) the volume of search queries over time for computer science conferences and (ii) stock market closing prices for S&P 500 companies. Since neither of these cases is intrinsically low-rank, we study the per-

formance of Zono-CC on low-rank approximations of the data. Curiously, the best results are achieved on extremely low-rank approximations. Our goal in both the synthetic and real-world experiments is to compare our algorithm to other well-known correlation clustering algorithms and, when possible, see how well our algorithm is able to approximate the optimal solution. On the real-world data, we also run the k-means procedure and find that it is unable to create the clustering we find via correlation clustering.

For our last experiment, we show how to cluster any unsigned network with ZonoCC, by first obtaining an embedding of the network's vertices into a low-dimensional space. We use this technique to cluster and study the structure of several networks from the Facebook 100 dataset [27]. Here we compare ZonoCC against k-means, an algorithm that is very commonly applied to cluster data in low-dimensional vector spaces. The goal of this final experiment is not to show that ZonoCC is better in itself, but to analyze the results of ZonoCC when both high-quality and low-quality embeddings of the vertices are applied. We find that with lower-quality embeddings, ZonoCC is able to better uncover meaningful structure in the networks than k-means.

Our experiments revolve around the following four algorithms, three of which are specifically intended for correlation clustering. In our experiments we show runtimes for guidance only and note that these are non-optimized implementations. We make code for all of our algorithms and experiments available at `https://github.com/nveldt/PSDCC`.

Exact ILP.
For small problems, we compute the optimal solution to the correlation clustering problem by solving an integer linear program with the commercial software Gurobi.

CGW.
The 0.7664-approximation for maximizing agreements in weighted graphs, based on a semidefinite programming relaxation, by Charikar, Guruswami, and Wirth [8].

Pivot.
The fast algorithm developed by Ailon, Charikar, and Newman for ±1 correlation clustering instances [1]. It uniformly randomly selects a vertex, clusters it with all nodes similar to it, and repeats. Depending on problem size, we return the best result from 1000 or 2000 instantiations.

k-means.
The standard Lloyd's k-means algorithm, as implemented in MATLAB, with k-means++ initialization [2]. Because of its speed, we return the best of 100 instantiations.

5.1 Synthetic Datasets

We begin by showing that ZonoCC computes a very good approximation to the optimal PSD-CC objective even though it does not test all vertices of the zonotope. We use several synthetic datasets, each with a true planted clustering, for a range of values of rank. We also compare the performance of Pivot and ZonoCC on larger datasets where we can no longer solve the problem optimally or run the semidefinite relaxation. The final experiment on synthetic data demonstrates how ZonoCC performs for a varying number of iterations.

For the first experiment we generate datasets with a true planted clustering perturbed by a small amount of Gaussian noise. We run each algorithm and measure its objective score relative to the score of the planted clustering. This setup allows us to test how well each method is able to perform in the high signal-to-noise ratio regime. In Figure 2 we display relative objective scores and runtimes for different algorithms on synthetic datasets with planted clusters for a range of rank, d, values from 2 to 20, and problem size $n = 10d$. We use 50,000 iterations of ZonoCC and 1000 instantiations of Pivot in each case. For this experiment ZonoCC outperforms all other methods for matrices up to rank 15. At this point CGW begins to take over in objective score, although it does so at the expense of a much longer runtime. Recall that we are running ZonoCC for a fixed number of iterations; we would expect to see improved objective scores for running the algorithm longer as problem size increases. Note that even for rank 20, ZonoCC still achieves an objective score that is within 85% of the score of the planted clustering.

In the second synthetic experiment, we wish to understand how ZonoCC compares with Pivot as we scale the problems up in *size*, for a fixed rank $d = 5$. This setting renders the CGW method infeasible, and so we show results only for ZonoCC with 1000 iterations, and 2000 instantiations of Pivot in Figure 3. These figures show us that ZonoCC always outperforms Pivot in objective, with the ratio of objective scores slowly growing as the problem size increases. We remark that the running time for each algorithm involves both generating clusterings as well as checking the objective value for each clustering. For small values of n, computing the objective scores becomes a more noticeable fraction of the computation time. For this reason the runtime of ZonoCC is much faster than Pivot for small n because we are checking half the number of clusterings. As problem size increases, we see that both algorithms roughly scale linearly in n.

In the final synthetic experiment, we study how the approximation changes as the number of iterations of ZonoCC increases. The results in Figure 4 show that the algorithm quickly attains a near-optimal solution, but moves closer to optimality slowly, as it continues to explore more vertices of the zonotope. The results displayed are for $n = 3000$ and $d = 5$, though this behavior is typical for most instances.

5.2 Clustering Using Search Query Data

For our first real-world application we use ZonoCC to cluster top-tier computer science conferences based on search query volume data. Each search term is either a conference acronym (e.g., "ICML"), or is an acronym concatenated with "conference" (e.g., "WWW conference"). For each search term, we obtain from Google Trends a time series of the volume of search queries for each month over the course of a six-year period, 2010–16. This data appears to fluctuate for repeated API calls, so before clustering we smooth out the data to capture the overall trend in each time series (using exponential smoothing with parameter $\alpha = 0.5$). Second, we remove the trend across all time series by fitting a quadratic polynomial to the mean volume. Finally, we 'z-score' normalize the time series. We then calculate the correlation coefficient between each pair of conferences using the processed data. This gives us a full-rank matrix of correlation values, of which we take a low-rank approximation to feed to ZonoCC.
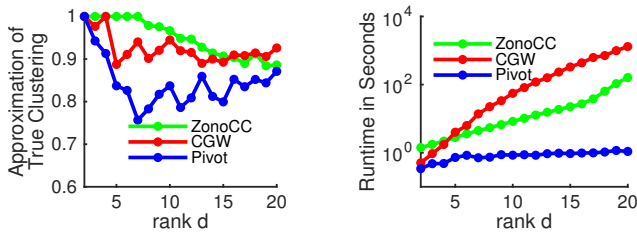
Figure 2: Results for ZonoCC (green), CGW (red), and Pivot (blue) on synthetic datasets with a true underlying clustering structure. The left plot gives each algorithm's approximation to the score of the planted clustering for d ranging from 2 to 20 and $n = 10d$. The right plot shows runtimes for each method. For both plots we take the median over five trials. ZonoCC outperforms other algorithms for all values of d up to 15. At this point CGW finds a better clustering score, but does so at the expense of a much longer runtime. Even for high values of rank, we note that ZonoCC achieves a score that is 85% of the planted clustering.
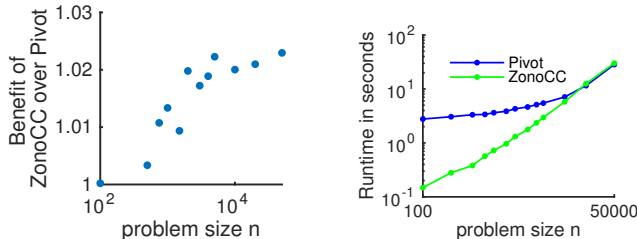

Figure 3: On the left is the ratio of ZonoCC's score to Pivot's on synthetic datasets of increasing size n and rank $d = 5$. ZonoCC always has a higher objective value, and as problem size grows, the ratio between the objective scores of the two algorithms increases. On the right are corresponding runtimes: ZonoCC is faster for problem sizes under $n = 10000$. For higher values, times are comparable and scale roughly linearly in n.

Using the rank-3 approximation gives us the optimal clustering (as determined by the ILP). In this case, both PIVOT and CGW also find the optimal solution. We show the runtime and objective values in the upper part of Table 1. The optimal clustering consists of three clusters: the two large clusters and an outlier set of only one conference, the International Conference on Computer Graphics Theory and Application.

It is interesting to observe the significance of the optimal clustering of this dataset. In Figure 5 we plot for each cluster the smoothed search query data for all of the conferences in the cluster. We observe that ZonoCC effectively partitions the dataset into conferences that have increased in search query volume over the course of the past six years, and those that have experienced an overall decrease in search volume. We expect it to be unsurprising to our readers that the WWW conference is in the "growing" cluster. We are unable to find any set of three clusters from k-MEANS that resembles the result of correlation clustering for this problem,
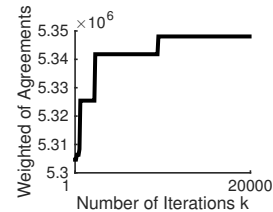

Figure 4: Best objective value, as a function of number of iterations of ZonoCC, for a synthetic dataset with $n = 3000$ and $d = 5$. ZonoCC quickly finds a good-quality clustering, and slowly improves as we let the algorithm run longer.

Table 1: Objective scores and runtimes in seconds for correlation clustering algorithms on two real-world datasets. Due to the size of the stocks dataset, we can run only ZonoCC and Pivot on it.

Dataset		ZONOCC	PIVOT	CGW	ILP
CS Conf.	Obj.	7540.0	7540.0	7540.0	7540.0
$n = 157$	Time	7	1	1380	52
Stocks	Obj.	5100.2	5099.5	—	—
$n = 497$	Time	40	20	—	—

as k-MEANS tends to generate three clusters of nearly equal size.

We also run ZONOCC for 50,000 iterations on rank-d approximations of the correlation clustering matrix, where d ranges from 2 to 15. As d increases, ZONOCC tends to form more clusters, but is always able to identify two large groups of conferences that are highly correlated. In terms of the correlation clustering objective on the original (and not low-rank) matrix, ZONOCC decreases in performance only because we must maintain a constant number of iterations for the sake of runtime, while the number of zonotope vertices increases exponentially in d. This behavior is illustrated in Figure 6: even though these are sub-optimal, the clustering returned always has two large clusters and small groups of outliers.
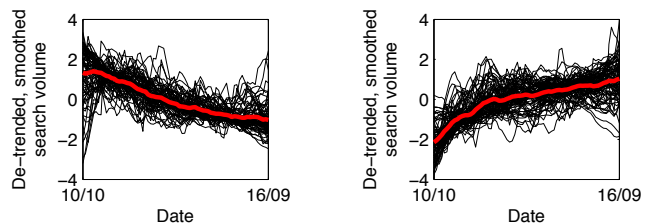

Figure 5: Smoothed search query data for top-tier computer science conferences. The partition discovered by ZonoCC splits the conferences into those whose search volume shows an overall decreasing trend (left), and an increasing trend (right); WWW is in the increasing cluster. For each plot we show an average of the clusters as a thick red line. There is also a third cluster with only one outlier conference that is not shown here.
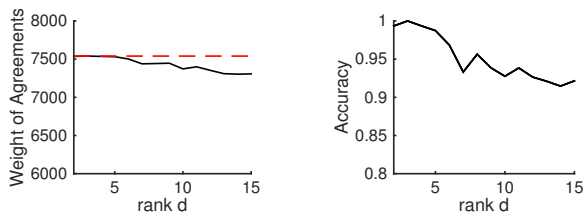
Figure 6: The left plot shows the objective scores achieved by ZonoCC for CS conference search query data as we increase rank. The dotted line shows the score of the optimal clustering, which ZonoCC remains very close to, regardless of rank. The right plot shows the accuracy of the rank-d clustering, which is computed by considering all $\binom{n}{2}$ pairs in the clustering, and finding the fraction of those pairs for which the decision to cluster together or cluster apart agrees with the optimal clustering of the dataset. In all cases, the accuracy of ZonoCC is above 90%.

5.3 Stock Market Data

The second study we consider is to cluster time series comprising stock market closing prices on different days of the year. We obtain prices for 497 stocks from the S&P 500 from Yahoo's Finance API over the 253 trading days in a year. We use the correlation between these time series to generate the input to our correlation clustering experiment. In this case we are unable to run the ILP to certify the clustering as optimal, and are unable to run the CGW algorithm due to insufficient memory. Thus, we just compare ZonoCC against Pivot: the resulting clusterings are very close, but ZonoCC finds the better clustering (see the lower part of Table 1 for the objectives and runtimes).

Similarly to our previous experiment, we discover that there are two large groups of closely correlated stocks and a third cluster with an outlier stock. The outlier is "Public Storage" (PSA), whose stock prices are largely uncorrelated with all other companies. For comparison, running k-means with 3 clusters always splits up many of the closely correlated companies.

5.4 Clustering Networks via Embeddings

We can use ZonoCC to cluster any dataset where each entry is represented by a vector in a low-dimensional vector space. This means that our algorithm can be used to cluster unsigned network data as long as we have a way to embed the nodes of the network in a low-dimensional space. Such embeddings have been an active area of research recently [12, 24]. We demonstrate how to combine ZonoCC with two different graph embedding techniques to produce large clusterings to analyze several networks from the Facebook 100 datasets [27].

The *purpose* of these experiments is not to demonstrate that ZonoCC achieves a superior clustering result given the metadata. (Indeed, there is no one algorithm that can achieve this [23].) Rather we wish to compare ZonoCC and k-means – in terms of how their clusters reflect the metadata – on low-quality embeddings from the eigenvectors of the Laplacian and on a high-quality embedding from node2vec [12].

Datasets. The datasets we use are subsets of the Facebook graph at certain US universities on a certain day in 2005.

These include an undirected graph and anonymized metadata regarding each person's student-or-faculty status, gender, major, dorm/residence, and graduation year. We run our experiments on the following networks of different sizes: Reed, Caltech, Swarthmore, Simmons, and Johns Hopkins. We aim to cluster this data based on friendship links in the graph – reflected in the embeddings – and in the process see how the clusterings might be related to different attributes.

Embeddings. We consider two different ways to embed each node into a low-dimensional space based on the edge structure of the graph. The first is to take a subset of the eigenvectors of the normalized Laplacian of the network: $\mathcal{L} = \mathcal{I} - D^{-1/2}AD^{-1/2}$ where D is the diagonal matrix of node degrees and $\mathcal{I}$ is the identity matrix. If we take the d eigenvectors corresponding to the smallest nonzero eigenvalues of $\mathcal{L}$, this gives an embedding in $\mathbb{R}^d$.

The second embedding we consider comes from an algorithmic framework developed by Grover and Leskovec [12] called *node2vec* for mapping nodes in a network to a low-dimensional feature space for representational learning. The points of the embedding lie in $\mathbb{R}^d$ for a user-specified d.

Results. For each of the networks studied, we obtain two embeddings into $\mathbb{R}^3$, one from the normalized Laplacian and the other using *node2vec*. For each embedding, we center the data by subtracting the mean point. This gives us a set of n vectors with both positive and negative entries. We then run ZonoCC on each embedding, and compare against running k-means for the same number of clusters as the output from ZonoCC.

We analyze our clusterings by observing how the clusters relate to four of the meta-data attributes: student-or-faculty status, major, dorm/residence, and graduation year. The metric we use is the proportion of pairs of people in the same cluster that share a given metadata attribute. Equivalently, we can think of this as the probability that two people selected uniformly at random from the same cluster share the attribute. We can also compute this metric for the entire network to get a baseline score. The results for this experiment are given in Table 2, where the "None" method places all nodes into a single cluster (which is the baseline probability). Note that the only meaningful column across the networks is the *Year* attribute. In addition, Caltech, which is a small school with a strong residential population, shows a similar effect for the dorm attribute. Thus, we focus our attention on the Year attribute.

The table shows that for *node2vec* embeddings, k-means always gives a higher proportion than ZonoCC except for Caltech and Johns Hopkins, where they are effectively the same. In contrast, for the embeddings from the Laplacian, the ZonoCC always shows stronger alignment with the year attribute. At the very least, this is a demonstration that ZonoCC and k-means can alternate in performance on any given clustering task. However, we suspect that this is evidence that ZonoCC is likely to be better in cases with *weak features* (such as the Laplacian).

6. RELATED WORK

Our work builds on several years of research in correlation clustering. The problem was originally introduced for complete and unweighted graphs by Bansal, Blum, and Chawla [5], who proved NP-hardness and gave a PTAS for maximizing agreements and a constant factor approximation for minimizing disagreements. Charikar, Guruswami,

Table 2: Proportion of pairs of people in the same cluster that share the given attribute. All networks display a strong connection between the clusterings and the graduation year. ZonoCC is better at detecting this trend on the low-quality Laplacian embedding, whereas k-means performs better on the more sophisticated node2vec embedding.

Network	Emb.	Method	Stud. or Fac.	Major	Dorm	Year
Reed	—	None	0.725	0.037	0.015	0.137
$n = 962$	N2V	ZonoCC	0.698	0.039	0.018	0.278
		k-means	0.756	0.039	0.020	0.325
	Lap	ZonoCC	0.744	0.038	0.018	0.298
		k-means	0.745	0.038	0.018	0.290
Caltech	—	None	0.564	0.063	0.078	0.142
$n = 769$	N2V	ZonoCC	0.576	0.064	0.160	0.151
		k-means	0.566	0.065	0.127	0.145
	Lap	ZonoCC	0.601	0.065	0.087	0.166
		k-means	0.578	0.064	0.080	0.146
Swarthmore	—	None	0.628	0.045	0.049	0.146
$n = 1659$	N2V	ZonoCC	0.620	0.048	0.055	0.262
		k-means	0.627	0.049	0.051	0.265
	Lap	ZonoCC	0.599	0.047	0.042	0.205
		k-means	0.599	0.046	0.042	0.197
Simmons	—	None	0.753	0.043	0.045	0.161
$n = 1518$	N2V	ZonoCC	0.716	0.043	0.064	0.378
		k-means	0.717	0.043	0.065	0.379
	Lap	ZonoCC	0.763	0.044	0.047	0.167
		k-means	0.761	0.044	0.045	0.162
Johns Hop.	—	None	0.618	0.036	0.020	0.134
$n = 5180$	N2V	ZonoCC	0.612	0.041	0.025	0.213
		k-means	0.592	0.041	0.024	0.203
	Lap	ZonoCC	0.632	0.046	0.026	0.229
		k-means	0.608	0.042	0.023	0.187

and Wirth [7] later extended these results by improving the constant factor approximation for minimizing disagreements, and gave a 0.7664-approximation for maximizing agreements in general weighted graphs based on a semidefinite programming relaxation. The approximation for minimizing disagreements in unweighted graphs was improved to 2.5 by Ailon, Charikar, and Newman [1], who at the same time developed the simplified Pivot algorithm. Recently Asteris et al. [3] gave a PTAS for maximizing agreements on unweighted bipartite graphs by obtaining a low-rank approximation of the graph's biadjacency matrix. Our work extends these results by showing how low-rank structure can also be exploited for general weighted correlation clustering.

Beyond the correlation clustering literature, our work shares similarities with other results on NP-hard problems that become solvable in polynomial time for low-rank positive semidefinite input. Ferrez, Fukuda, and Liebling gave a polynomial time solution for maximizing a quadratic program in $\{0, 1\}$ variables on low-rank positive semidefinite matrices [10], and Markopoulos, Karystinos, and Pados proved an analogous result for the ± 1 binary case [17]. While these results seek to optimally partition a set of vectors into two

clusters, our work can be seen as a generalization to arbitrarily many clusters.

Our approach in solving low-rank correlation clustering shares many similarities as well with the *spannogram* framework for exactly solving combinatorially constrained quadratic optimization problems on low-rank input [14, 4, 21]. In particular, for the NP-hard densest subgraph problem, Papailiopoulos et al. used this framework to prove that a low-rank bilinear relaxation of the densest subgraph problem is solvable in polynomial time for low-rank input [22].

7. CONCLUSIONS

Our results introduce a new approach to solving general weighted correlation clustering problems by considering the rank and structure of the underlying matrix associated with the problem. This opens a number of new directions in correlation clustering-based approaches. The algorithm we present offers a fast and accurate method for solving correlation clustering problems where the input can be represented or at least well approximated by a low-rank positive semidefinite matrix. We demonstrate a number of applications including clustering time series data from search queries relating to top-tier computer science conferences and stock closing prices. We also demonstrate how this method can be used with embeddings of network data into low-dimensional spaces.

In future work we wish to prove more rigorous theoretical approximation results for our methods. Specifically, we would like to give an approximation bound for k iterations of ZonoCC, and also give rigorous bounds on the correlation clustering objective when taking a low-rank approximation.

8. ACKNOWLEDGMENTS

References

[1] N. Ailon, M. Charikar, and A. Newman. Aggregating inconsistent information: ranking and clustering. *Journal of the ACM (JACM)*, 55(5):23:1–23:27, 2008.

[2] D. Arthur and S. Vassilvitskii. *k*-means++: The advantages of careful seeding. In *Proceedings of the 2007 ACM-SIAM Symposium on Discrete Algorithms (SODA '07)*, pages 1027–1035, 2007.

[3] M. Asteris, A. Kryillidis, D. Papailiopoulos, and A. G. Dimakis. Bipartite correlation clustering: Maximizaing agreements. In *Proceedings of the 2016 International Conference on Artificial Intelligence and Statistics (AISTATS '16)*, volume 51, pages 121–129, 2016.

[4] M. Asteris, D. S. Papailiopoulos, and G. N. Karystinos. The sparse principal component of a constant-rank matrix. *IEEE Transactions on Information Theory*, 60(4):2281–2290, 2014.

[5] N. Bansal, A. Blum, and S. Chawla. Correlation clustering. *Machine Learning*, 56(1-3):89–113, 2004.

[6] A. Bhattacharya and R. K. De. Divisive correlation clustering algorithm (DCCA) for grouping of genes: detecting varying patterns in expression profiles. *Bioinformatics*, 24(11):1359–1366, 2008.

[7] M. Charikar, V. Guruswami, and A. Wirth. Clustering with qualitative information. In *Proceedings of the 2003 International Symposium on Foundations of Computer Science, (FOCS '03)*, pages 524–533. IEEE, 2003.

[8] M. Charikar, V. Guruswami, and A. Wirth. Clustering with qualitative information. *Journal of Computer and System Sciences*, 71(3):360–383, 2005.

[9] T. M. Cover and B. Efron. Geometrical probability and random points on a hypersphere. *The Annals of Mathematical Statistics*, pages 213–220, 1967.

[10] J.-A. Ferrez, K. Fukuda, and T. M. Liebling. Solving the fixed rank convex quadratic maximization in binary variables by a parallel zonotope construction algorithm. *European Journal of Operational Research*, 166(1):35–50, 2005.

[11] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*. WH Freeman and Company, New York, 1979.

[12] A. Grover and J. Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 2016 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*, pages 855–864. ACM, 2016.

[13] F. K. Hwang, S. Onn, and U. G. Rothblum. A polynomial time algorithm for shaped partition problems. *SIAM Journal on Optimization*, 10(1):70–81, 1999.

[14] G. N. Karystinos and A. P. Liavas. Efficient computation of the binary vector that maximizes a rank-deficient quadratic form. *IEEE Transactions on information theory*, 56(7):3581–3593, 2010.

[15] S. Kim, S. Nowozin, P. Kohli, and C. D. Yoo. Higher-order correlation clustering for image segmentation. In *Advances in neural information processing systems (NIPS '11)*, pages 1530–1538, 2011.

[16] V. Klee. Separation properties of convex cones. *Proceedings of the American Mathematical Society*, 6(2):313–318, 1955.

[17] P. P. Markopoulos, G. N. Karystinos, and D. A. Pados. Optimal algorithms for-subspace signal processing. *IEEE Transactions on Signal Processing*, 62(19):5046–5058, 2014.

[18] A. McCallum and B. Wellner. Conditional models of identity uncertainty with application to noun coreference. In *Advances in neural information processing systems (NIPS '05)*, pages 905–912, 2005.

[19] M. E. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, 74(3):036104:1 – 036104:19, 2006.

[20] S. Onn and L. J. Schulman. The vector partition problem for convex objective functions. *Mathematics of Operations Research*, 26(3):583–590, 2001.

[21] D. S. Papailiopoulos, A. G. Dimakis, and S. Korokythakis. Sparse pca through low-rank approximations. In *Proceedings of the 2013 International Conference on Maching Learning (ICML '13)*, pages 747–755, 2013.

[22] D. S. Papailiopoulos, I. Mitliagkas, A. G. Dimakis, and C. Caramanis. Finding dense subgraphs via low-rank bilinear optimization. In *Proceedings of the 2014 International Conference on Machine Learning (ICML '14)*, volume 14, pages 1890–1898, 2014.

[23] L. Peel, D. B. Larremore, and A. Clauset. The ground truth about metadata and community detection in networks. *arXiv*, cs.SI:1608.05878, 2016.

[24] B. Perozzi, R. Al-Rfou, and S. Skiena. DeepWalk: Online learning of social representations. In *Proceedings of the 2014 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '14)*, pages 701–710, 2014.

[25] R. A. Rankin. On the closest packing of spheres in n dimensions. *Annals of Mathematics*, 48(4):1062–1081, 1947.

[26] K. Stinson, D. F. Gleich, and P. G. Constantine. A randomized algorithm for enumerating zonotope vertices. *arXiv preprint arXiv:1602.06620*, 2016.

[27] A. L. Traud, P. J. Mucha, and M. A. Porter. Social structure of facebook networks. *Physica A: Statistical Mechanics and its Applications*, 391(16):4165–4180, 2012.

[28] J. Van Gael and X. Zhu. Correlation clustering for crosslingual link detection. In *Proceedings of the 2007 International Joint Conference on Artificial Intelligence (IJCAI '07)*, pages 1744–1749, 2007.

[29] X. Zhang and M. Newman. Multiway spectral community detection in networks. *Physical Review E*, 92(5):052808, 2015.