Cluster Analysis is Convex

Madhushini Narayana Prasad and Grani A. Hanasusanto

Graduate Program in Operations Research and Industrial Engineering, The University of Texas at Austin, USA

June 21, 2017

Abstract

In this paper, we show that the popular K-means clustering problem can equivalently be reformulated as a conic program of polynomial size. The arising convex optimization problem is NP-hard, but amenable to a tractable semidefinite programming (SDP) relaxation that is tighter than the current SDP relaxation schemes in the literature. In contrast to the existing schemes, our proposed SDP formulation gives rise to solutions that can be leveraged to identify the clusters. We devise a new approximation algorithm for K-means clustering that utilizes the improved formulation and empirically illustrate its superiority over the state-of-the-art solution schemes.

1 Introduction

Given an input set of data points, cluster analysis endeavors to discover a fixed number of disjoint clusters so that the data points in the same cluster are closer to each other than to those in other clusters. Cluster analysis is fundamental to a wide array of applications in science, engineering, economics, psychology, marketing, etc. [15, 14]. One of the most popular approaches for cluster analysis is the *K-means clustering* [21, 19, 14]. The goal of K-means clustering is to partition the data points into K clusters so that the sum of squared distances to the respective cluster centroids is minimized. Formally, the K-means clustering seeks for a solution to the mathematical optimization problem

$$\begin{array}{lll} \min & \displaystyle\sum_{i=1}^{K} \sum_{n \in \mathcal{P}_i} \|\boldsymbol{x}_n - \boldsymbol{c}_i\|^2 & \\ \text{s.t.} & \mathcal{P}_i \subseteq \{1, \ldots, N\},\ \boldsymbol{c}_i \in \mathbb{R}^D & \forall i \in \{1, \ldots, K\} \\ & \boldsymbol{c}_i = \dfrac{1}{|\mathcal{P}_i|} \displaystyle\sum_{n \in \mathcal{P}_i} \boldsymbol{x}_n & \\ & \mathcal{P}_1 \cup \cdots \cup \mathcal{P}_K = \{1, \ldots, N\},\ \ \mathcal{P}_i \cap \mathcal{P}_j = \emptyset & \forall i, j \in \{1, \ldots, K\} : i \neq j. \end{array} \tag{1}$$

Here, $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ are the input data points, while $\mathcal{P}_1, \ldots, \mathcal{P}_K \subseteq \{1, \ldots, N\}$ are the output clusters. The vectors $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_K \in \mathbb{R}^D$ in (1) determine the cluster centroids, while the constraints on the last row of (1) ensure that the subsets $\mathcal{P}_1, \ldots, \mathcal{P}_K$ constitute a partition of the set $\{1, \ldots, N\}$.

Due to its combinatorial nature, the K-means clustering problem (1) is generically NP-hard [2]. A popular solution scheme for this intractable problem is the heuristic algorithm developed by Lloyd [19]. The algorithm initializes by randomly selecting the K cluster centroids. It then proceeds by alternating between the *assignment* step and the *update* step. In the assignment step the algorithm designates each data point to the closest centroid, while in the update step the algorithm determines new cluster centroids according to current assignment.

Another popular approach arises in the form of convex relaxation schemes [24, 4, 26]. In this approach, tractable semidefinite programming (SDP) lower bounds for (1) are derived. Solutions of these optimization problems are then transformed into the cluster assignments via well-constructed rounding procedures. Such convex relaxation schemes have a number of theoretically appealing properties. If the data points are supported on K disjoint balls then exact recovery is possible with high probability whenever the distance between any two balls is sufficiently large [4, 13]. A stronger model-free result is achievable if the cardinalities of the clusters are prescribed to the problem [26].

A closely related problem is the non-negative matrix factorization with orthogonality constraints (ONMF). Given an input data matrix $\boldsymbol{X}$, the ONMF problem seeks for non-negative matrices $\boldsymbol{F}$ and $\boldsymbol{U}$ so that both the product $\boldsymbol{F}\boldsymbol{U}^\top$ is close to $\boldsymbol{M}$ in view of the Frobenius norm and the orthogonality constraint $\boldsymbol{U}^\top\boldsymbol{U} = \mathbb{I}$ is satisfied. Although ONMF is not precisely equivalent to K-means, solutions to this problem have the clustering property [10, 18, 11, 16]. In [25], it is shown that the ONMF problem is in fact equivalent to a weighted variant of the K-means clustering problem.

In this paper, we attempt to obtain equivalent convex reformulations for the ONMF and the K-means clustering problems. To derive these reformulations, we adapt the results by Burer and Hong [8] who showed that any (non-convex) quadratically constrained quadratic program can be reformulated as a linear program over the convex cone of completely positive matrices. The resulting optimization problem is called a *completely positive program*. Such a transformation does not immediately mitigate the intractability of the original problem, since solving a generic completely positive program is NP-hard. However, the complexity of the problem is now entirely absorbed in the cone of completely positive matrices which admits tractable semidefinite representable outer approximations [23, 9, 17]. Replacing the cone with these outer approximations gives rise to SDP relaxations of the original problem that in principle can be solved efficiently.

As byproducts of our derivations, we identify a new condition that makes the ONMF and the K-means clustering problems equivalent and we obtain a new SDP relaxation for the K-means clustering problem that is tighter than the well-known relaxation proposed by Peng and Wei [24]. The contributions of this paper can be summarized as follows.

1. We disclose a new connection between ONMF and K-means clustering. We show that K-means clustering is equivalent to ONMF if an additional requirement on the binarity of the solution to the latter problem is imposed. This amends the previous incorrect result by Ding et al. [10, Section 2] and Li and Ding [18, Theorem 1] who claimed that both problems are equivalent.[1]

2. We derive exact conic programming reformulations for the ONMF and K-means clustering problems that are principally amenable to numerical solutions. To our best knowledge, we are the first to obtain equivalent convex reformulations for these problems.

3. In view of the equivalent convex reformulation, we derive a tighter SDP relaxation for the K-means clustering problem whose solutions can be used to construct high quality estimates of the cluster assignment.

4. We devise a new approximation algorithm for the K-means clustering problem that leverages the improved relaxation and numerically highlight its superiority over the state-of-the-art SDP approximation scheme and the Lloyd's algorithm.

The remainder of the paper is structured as follows. We derive a conic programming reformulation for the ONMF problem in Section 2. We extend this result to the setting of K-means clustering in Section 3. Section 4 develops an SDP relaxation and designs a new approximation algorithm for K-means clustering. Finally, we empirically assess the performance of our proposed algorithm in Section 5.

Notation: For any $K \in \mathbb{N}$, we define $[K]$ as the index set $\{1, \ldots, K\}$. We denote by $\mathbb{I}$ the identity matrix and by $\mathbf{e}$ the vector of all ones. We also define $\mathbf{e}_i$ as the i-th canonical basis vector. Their dimensions will be clear from the context. The trace of a square matrix $\boldsymbol{M}$ is denoted as $\mathrm{tr}(\boldsymbol{M})$. We define $\mathrm{diag}(\boldsymbol{v})$ as the diagonal matrix whose diagonal elements comprise the entries of $\boldsymbol{v}$. For any non-negative vector $\boldsymbol{v} \in \mathbb{R}^K_+$, we define the cardinality of all positive elements of $\boldsymbol{v}$ by $\#\boldsymbol{v} = |\{i \in [K] : v_i > 0\}|$. For any matrix $\boldsymbol{M} \in \mathbb{R}^{M \times N}$, we denote by $\boldsymbol{m}_i \in \mathbb{R}^M$ the vector that corresponds to the i-th column of $\boldsymbol{M}$. The set of all symmetric matrices in $\mathbb{R}^{K \times K}$ is denoted as $\mathbb{S}^K$, while the cone of positive semidefinite matrices in $\mathbb{R}^{K \times K}$ is denoted as $\mathbb{S}^K_+$. The cone of completely positive matrices over a set $\mathcal{K}$ is denoted as $\mathcal{C}(\mathcal{K}) = \mathrm{cl\ conv}\{\boldsymbol{x}\boldsymbol{x}^\top : \boldsymbol{x} \in \mathcal{K}\}$. For any $\boldsymbol{Q}, \boldsymbol{R} \in \mathbb{S}^K$ and any cone $\mathcal{C}$, the relations $\boldsymbol{Q} \succeq \boldsymbol{R}$ and $\boldsymbol{Q} \succeq_{\mathcal{C}} \boldsymbol{R}$ denote that $\boldsymbol{Q} - \boldsymbol{R}$ is an element of $\mathbb{S}^K_+$ and $\mathcal{C}$, respectively. The $(K+1)$-dimensional second-order cone is defined as $\mathbb{SOC}^{K+1} = \{(\boldsymbol{x}, t) \in \mathbb{R}^{K+1} : \|\boldsymbol{x}\| \leq t\}$. We denote by $\mathbb{SOC}^{K+1}_+ = \mathbb{SOC}^{K+1} \cap \mathbb{R}^{K+1}_+$ the intersection of the $K+1$-dimensional second-order cone and the non-negative orthant.

[1]To our best understanding, they have shown only one of the implications that establish an equivalence.

2 Orthogonal Non-Negative Matrix Factorization

In this section, we consider the ONMF problem given by

$$\begin{array}{ll} \min & \|\boldsymbol{X} - \boldsymbol{F}\boldsymbol{U}^\top\|_F^2 \\ \text{s.t.} & \boldsymbol{F} \in \mathbb{R}_+^{D\times K},\ \boldsymbol{U} \in \mathbb{R}_+^{N\times K} \\ & \boldsymbol{U}^\top\boldsymbol{U} = \mathbb{I}. \end{array} \tag{2}$$

Here, $\boldsymbol{X} \in \mathbb{R}^{D\times N}$ is a matrix whose columns comprise the N data points $\{\boldsymbol{x}_n\}_{n\in[N]}$ in $\mathbb{R}^D$. We remark that the problem (2) is generically intractable since we are minimizing a non-convex quadratic objective function over the Stiefel manifold [1, 3]. By expanding the Frobenius norm in the objective function and noting that $\boldsymbol{U}^\top\boldsymbol{U} = \mathbb{I}$, we find that (2) is equivalent to the problem

$$\begin{array}{ll} \min & \operatorname{tr}\left(\boldsymbol{X}^\top\boldsymbol{X} - 2\boldsymbol{X}\boldsymbol{U}\boldsymbol{F}^\top + \boldsymbol{F}^\top\boldsymbol{F}\right) \\ \text{s.t.} & \boldsymbol{F} \in \mathbb{R}_+^{D\times K},\ \boldsymbol{U} \in \mathbb{R}_+^{N\times K} \\ & \boldsymbol{U}^\top\boldsymbol{U} = \mathbb{I}. \end{array} \tag{3}$$

If all elements of $\boldsymbol{X}$ are non-negative, then we can reduce (2) into a simpler problem involving only the decision matrix $\boldsymbol{U}$.

Lemma 1. *If $\boldsymbol{X}$ is a non-negative matrix then the problem* (2) *is equivalent to the non-convex program*

$$\begin{array}{ll} \min & \operatorname{tr}(\boldsymbol{X}^\top\boldsymbol{X} - \boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{U}\boldsymbol{U}^\top) \\ \text{s.t.} & \boldsymbol{U} \in \mathbb{R}_+^{N\times K} \\ & \boldsymbol{U}^\top\boldsymbol{U} = \mathbb{I}. \end{array} \tag{4}$$

Proof. Solving the minimization over $\boldsymbol{F} \in \mathbb{R}_+^{D\times K}$ analytically in (3), we find that the solution $\boldsymbol{F} = \boldsymbol{X}\boldsymbol{U}$ is feasible and optimal. Substituting this solution into the objective function of (3), we arrive at the equivalent problem (4). This completes the proof. □

In the following, we first derive a convex reformulation for the simpler problem (4). We remark that this problem is still intractable due to the non-convexity of the objective function and the constraint system. Thus, any resulting convex formulation will in general remain intractable. To derive our reformulation, we rely on the following lemma which is proven in [6].

Lemma 2 ([6, Lemma 2.2])**.** *Let the non-negative matrix $\boldsymbol{Z} \in \mathbb{R}_+^{N\times N}$ and vector $\boldsymbol{z} \in \mathbb{R}_+^N$ satisfy the linear constraints*

$$\boldsymbol{f}_\ell^\top\boldsymbol{z} = g_\ell, \quad \boldsymbol{f}_\ell^\top\boldsymbol{Z}\boldsymbol{f}_\ell = g_\ell^2 \qquad \forall \ell \in \mathcal{L},$$

where the constraint system $\boldsymbol{f}_\ell^\top\boldsymbol{z} = g_\ell$, $\ell \in \mathcal{L}$, defines a bounded polytope *in $\mathbb{R}_+^N$. Consider the completely positive matrix decomposition*

$$\begin{bmatrix} \boldsymbol{Z} & \boldsymbol{z} \\ \boldsymbol{z}^\top & 1 \end{bmatrix} = \sum_{t\in\mathcal{T}} \begin{bmatrix} \boldsymbol{\zeta}_t \\ \eta_t \end{bmatrix} \begin{bmatrix} \boldsymbol{\zeta}_t \\ \eta_t \end{bmatrix}^\top,$$

with $\boldsymbol{\zeta}_t \in \mathbb{R}_+^N$ *and* $\eta_t \in \mathbb{R}_+$ *for all* $t \in \mathcal{T}$, *and define the subsets* $\mathcal{T}_+ = \{t \in \mathcal{T} : \eta_t > 0\}$ *and* $\mathcal{T}_0 = \{t \in \mathcal{T} : \eta_t = 0\}$ *of* $\mathcal{T}$. *Then we have*

(i) $\boldsymbol{f}_\ell^\top(\boldsymbol{\zeta}_t/\eta_t) = g_\ell$ *for all* $t \in \mathcal{T}_+$ *and* $\ell \in \mathcal{L}$;

(ii) $\boldsymbol{\zeta}_t = \mathbf{0}$ *for all* $t \in \mathcal{T}_0$.

We are now ready to state our first main result.

Theorem 1. *Problem* (4) *is equivalent to the following completely positive program.*

$$
\begin{array}{lll}
\min & \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X}) - \displaystyle\sum_{i \in [K]} \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{V}_{ii}) & \\
\text{s.t.} & \boldsymbol{p}_i \in \mathbb{SOC}_+^{N+1},\ \boldsymbol{Q}_{ij} \in \mathbb{R}_+^{(N+1)\times(N+1)},\ \boldsymbol{u}_i \in \mathbb{R}_+^N,\ \boldsymbol{V}_{ij} \in \mathbb{R}_+^{N\times N} & \forall i,j \in [K] \\
& \boldsymbol{p}_i = \begin{bmatrix} \boldsymbol{u}_i \\ 1 \end{bmatrix},\ \boldsymbol{Q}_{ij} = \begin{bmatrix} \boldsymbol{V}_{ij} & \boldsymbol{u}_i \\ \boldsymbol{u}_j^\top & 1 \end{bmatrix} & \forall i,j \in [K] \\
& \operatorname{tr}(\boldsymbol{V}_{ii}) = 1 & \forall i \in [K] \\
& \operatorname{tr}(\boldsymbol{V}_{ij}) = 0 & \forall i,j \in [K] : i \neq j \\
& \begin{bmatrix} \boldsymbol{Q}_{11} & \cdots & \boldsymbol{Q}_{1K} & \boldsymbol{p}_1 \\ \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{Q}_{K1} & \cdots & \boldsymbol{Q}_{KK} & \boldsymbol{p}_K \\ \boldsymbol{p}_1^\top & \cdots & \boldsymbol{p}_K^\top & 1 \end{bmatrix} \in \mathcal{C}\left((\mathbb{SOC}_+^{N+1})^K \times \mathbb{R}_+\right) &
\end{array}
\tag{5}
$$

Proof. By employing the notation for column vectors $\{\boldsymbol{u}_i\}_{i\in[K]}$, we can reformulate (4) equivalently as the problem

$$
\begin{array}{lll}
\min & \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X}) - \displaystyle\sum_{i \in [I]} \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{u}_i \boldsymbol{u}_i^\top) & \\
\text{s.t.} & \boldsymbol{U} \in \mathbb{R}_+^{N\times K} & \\
& \boldsymbol{u}_i^\top \boldsymbol{u}_i = 1 & \forall i \in [K] \\
& \boldsymbol{u}_i^\top \boldsymbol{u}_j = 0 & \forall i,j \in [K] : i \neq j.
\end{array}
\tag{6}
$$

Next, we introduce the auxiliary decision variables $\boldsymbol{p}_i$, $\boldsymbol{Q}_{ij}$, $i,j \in [K]$, that satisfy

$$
\boldsymbol{p}_i = \begin{bmatrix} \boldsymbol{u}_i \\ 1 \end{bmatrix},\ \boldsymbol{Q}_{ij} = \begin{bmatrix} \boldsymbol{u}_i \boldsymbol{u}_j^\top & \boldsymbol{u}_i \\ \boldsymbol{u}_j^\top & 1 \end{bmatrix} \qquad \forall i,j \in [K].
$$

Since $\boldsymbol{u}_i^\top \boldsymbol{u}_i = 1$, and consequently $\|\boldsymbol{u}_i\|_2 = 1$, we can without loss of generality impose the vector $\boldsymbol{p}_i$ to reside in the cone $\mathbb{SOC}_+^{N+1} = \mathbb{SOC}^{N+1} \cap \mathbb{R}_+^{N+1}$. Thus, we obtain the equivalent reformulation

$$
\begin{array}{lll}
\min & \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X}) - \sum\limits_{i \in [K]} \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{V}_{ii}) & \\
\text{s.t.} & \boldsymbol{p}_i \in \mathbb{SOC}_+^{N+1},\ \boldsymbol{Q}_{ij} \in \mathbb{R}_+^{(N+1)\times(N+1)},\ \boldsymbol{u}_i \in \mathbb{R}_+^N,\ \boldsymbol{V}_{ij} \in \mathbb{R}_+^{N\times N} & \forall i,j \in [K] \\
& \boldsymbol{p}_i = \begin{bmatrix} \boldsymbol{u}_i \\ 1 \end{bmatrix},\ \boldsymbol{Q}_{ij} = \begin{bmatrix} \boldsymbol{V}_{ij} & \boldsymbol{u}_i \\ \boldsymbol{u}_j^\top & 1 \end{bmatrix} & \forall i,j \in [K] \\
& \operatorname{tr}(\boldsymbol{V}_{ii}) = 1 & \forall i \in [K] \\
& \operatorname{tr}(\boldsymbol{V}_{ij}) = 0 & \forall i,j \in [K] : i \neq j \\
& \begin{bmatrix} \boldsymbol{Q}_{11} & \cdots & \boldsymbol{Q}_{1K} & \boldsymbol{p}_1 \\ \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{Q}_{K1} & \cdots & \boldsymbol{Q}_{KK} & \boldsymbol{p}_K \\ \boldsymbol{p}_1^\top & \cdots & \boldsymbol{p}_K^\top & 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{p}_1 \\ \vdots \\ \boldsymbol{p}_K \\ 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{p}_1 \\ \vdots \\ \boldsymbol{p}_K \\ 1 \end{bmatrix}^\top . &
\end{array}
\tag{7}
$$

The last constraint of (7) implies that the left-hand side matrix has rank 1. If we relax this constraint to the requirement that the matrix should merely belong to the completely positive cone $\mathcal{C}\left((\mathbb{SOC}_+^{N+1})^K \times \mathbb{R}_+\right)$, then we arrive at the completely positive program (5). The arising problem is a relaxation whose optimal value constitutes a lower bound on the optimal value of (7). Exactness of this reformulation is then obtained by adapting the procedures in [8]. Since we will employ and extend the same reformulation techniques in the subsequent section, we provide the derivations for the specific instance above as follows.

We first consider the completely positive decomposition of the rank-1 matrix given by

$$
\begin{bmatrix} \boldsymbol{Q}_{11} & \cdots & \boldsymbol{Q}_{1K} & \boldsymbol{p}_1 \\ \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{Q}_{K1} & \cdots & \boldsymbol{Q}_{KK} & \boldsymbol{p}_K \\ \boldsymbol{p}_1^\top & \cdots & \boldsymbol{p}_K^\top & 1 \end{bmatrix} = \sum_{t \in \mathcal{T}} \begin{bmatrix} \boldsymbol{\theta}_1^t \\ \vdots \\ \boldsymbol{\theta}_K^t \\ \eta_t \end{bmatrix} \begin{bmatrix} \boldsymbol{\theta}_1^t \\ \vdots \\ \boldsymbol{\theta}_K^t \\ \eta_t \end{bmatrix}^\top ,
$$

where $\mathcal{T}$ is a finite index set, and $\boldsymbol{\theta}_i^t \in \mathbb{SOC}_+^{N+1}$ and $\eta_t \in \mathbb{R}_+$ for every $i \in [K]$ and $t \in \mathcal{T}$. We next partition the index set $\mathcal{T}$ into the sets $\mathcal{T}_+ = \{t \in \mathcal{T} : \eta_t > 0\}$ and $\mathcal{T}_0 = \{t \in \mathcal{T} : \eta_t = 0\}$, which gives rise to the decomposition

$$
\begin{bmatrix} \boldsymbol{Q}_{11} & \cdots & \boldsymbol{Q}_{1K} & \boldsymbol{p}_1 \\ \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{Q}_{K1} & \cdots & \boldsymbol{Q}_{KK} & \boldsymbol{p}_K \\ \boldsymbol{p}_1^\top & \cdots & \boldsymbol{p}_K^\top & 1 \end{bmatrix} = \sum_{t \in \mathcal{T}_+} \eta_t^2 \begin{bmatrix} \frac{\boldsymbol{\theta}_1^t}{\eta_t} \\ \vdots \\ \frac{\boldsymbol{\theta}_K^t}{\eta_t} \\ 1 \end{bmatrix} \begin{bmatrix} \frac{\boldsymbol{\theta}_1^t}{\eta_t} \\ \vdots \\ \frac{\boldsymbol{\theta}_K^t}{\eta_t} \\ 1 \end{bmatrix}^\top + \sum_{t \in \mathcal{T}_0} \begin{bmatrix} \boldsymbol{\theta}_1^t \\ \vdots \\ \boldsymbol{\theta}_K^t \\ 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\theta}_1^t \\ \vdots \\ \boldsymbol{\theta}_K^t \\ 0 \end{bmatrix}^\top . \tag{8}
$$

For every $i \in [K]$ and $t \in \mathcal{T}$, we partition the vector $\boldsymbol{\theta}_i^t$ into the constituents $\boldsymbol{a}_i^t \in \mathbb{R}_+^N$ and $b_i^t \in \mathbb{R}_+$ such that

$$
\boldsymbol{\theta}_i^t = \begin{bmatrix} \boldsymbol{a}_i^t \\ b_i^t \end{bmatrix} .
$$

Since $\boldsymbol{\theta}_i^t \in \mathbb{SOC}_+^{N+1}$, we have by construction

$$\left\| \frac{\boldsymbol{a}_i^t}{\eta_t} \right\| \leq \frac{b_i^t}{\eta_t} \quad \forall t \in \mathcal{T}_+ \;\text{ and }\; \|\boldsymbol{a}_i^t\| \leq b_i^t \quad \forall t \in \mathcal{T}_0. \tag{9}$$

Next, for every $i \in [K]$ we consider the following completely positive decomposition involving only the matrix $\boldsymbol{Q}_{ii}$ and the vector $\boldsymbol{p}_i$ in (8).

$$\begin{bmatrix} \boldsymbol{Q}_{ii} & \boldsymbol{p}_i \\ \boldsymbol{p}_i^\top & 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{V}_{ii} & \boldsymbol{u}_i & \boldsymbol{u}_i \\ \boldsymbol{u}_i^\top & 1 & 1 \\ \boldsymbol{u}_i^\top & 1 & 1 \end{bmatrix} = \sum_{t\in\mathcal{T}_+} \eta_t^2 \begin{bmatrix} \frac{\boldsymbol{a}_i^t}{\eta_t} \\ \frac{b_i^t}{\eta_t} \\ 1 \end{bmatrix} \begin{bmatrix} \frac{\boldsymbol{a}_i^t}{\eta_t} \\ \frac{b_i^t}{\eta_t} \\ 1 \end{bmatrix}^\top + \sum_{t\in\mathcal{T}_0} \begin{bmatrix} \boldsymbol{a}_i^t \\ b_i^t \\ 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{a}_i^t \\ b_i^t \\ 0 \end{bmatrix}^\top \tag{10}$$

Lemma 2 then implies that

$$\frac{b_i^t}{\eta_t} = 1 \quad \forall t \in \mathcal{T}_+ \;\text{ and }\; b_i^t = 0 \quad \forall t \in \mathcal{T}_0.$$

Substituting these identities into (9), we obtain

$$\left\| \frac{\boldsymbol{a}_i^t}{\eta_t} \right\| \leq 1 \qquad \forall t \in \mathcal{T}_+ \quad \Longleftrightarrow \quad \left(\frac{\boldsymbol{a}_i^t}{\eta_t}\right)^\top \frac{\boldsymbol{a}_i^t}{\eta_t} \leq 1 \qquad \forall t \in \mathcal{T}_+ \tag{11}$$

and

$$\boldsymbol{a}_i^t = \mathbf{0} \qquad \forall t \in \mathcal{T}_0. \tag{12}$$

Next, the constraint $\operatorname{tr}(\boldsymbol{V}_{ii}) = 1$ in (5) together with the decomposition in (10) gives rise to

$$\sum_{t\in\mathcal{T}_+} \eta_t^2 \left(\frac{\boldsymbol{a}_i^t}{\eta_t}\right)^\top \frac{\boldsymbol{a}_i^t}{\eta_t} + \sum_{t\in\mathcal{T}_0} \left(\boldsymbol{a}_i^t\right)^\top \boldsymbol{a}_i^t = 1.$$

Thus, (12) yields

$$\sum_{t\in\mathcal{T}_+} \eta_t^2 \left(\frac{\boldsymbol{a}_i^t}{\eta_t}\right)^\top \frac{\boldsymbol{a}_i^t}{\eta_t} = 1.$$

Since $\sum_{t\in\mathcal{T}_+} \eta_t^2 = 1$, the point 1 on the right-hand-side of the equation lies in the convex hull of the scalars $\left(\boldsymbol{a}_i^t/\eta_t\right)^\top \boldsymbol{a}_i^t/\eta_t$, $t \in \mathcal{T}_+$. Thus, in view of the inequalities in (11), we must have $\left(\boldsymbol{a}_i^t/\eta_t\right)^\top \boldsymbol{a}_i^t/\eta_t = 1$ for all $t \in \mathcal{T}_+$ and $i \in [K]$. A similar argument allows us to further conclude that the penultimate constraint in (5) yields $\left(\boldsymbol{a}_i^t/\eta_t\right)^\top \boldsymbol{a}_j^t/\eta_t = 0$ for all $t \in \mathcal{T}_+$ and $i, j \in [K]$ such that $i \neq j$. In summary, we find that for any index $t \in \mathcal{T}_+$ the solution $(\hat{\boldsymbol{u}}_i)_{i\in[K]}$ satisfying

$$\hat{\boldsymbol{u}}_i = \frac{\boldsymbol{a}_i^t}{\eta_t} \qquad \forall i \in [K] \tag{13}$$

is feasible to the original problem (6). Verifying the objective value of the relaxed problem (5), we further find that it constitutes a convex combination of the objective values of (6) evaluated at the respective points in (13), *i.e.*,

$$\begin{aligned} \operatorname{tr}(\boldsymbol{X}^\top\boldsymbol{X}) - \sum_{i\in[K]} \operatorname{tr}(\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{V}_{ii}) &= \operatorname{tr}(\boldsymbol{X}^\top\boldsymbol{X}) - \sum_{i\in[K]} \operatorname{tr}\left(\boldsymbol{X}^\top\boldsymbol{X} \sum_{t\in\mathcal{T}_+} \eta_t^2 \frac{\boldsymbol{a}_i^t}{\eta_t}\frac{\boldsymbol{a}_i^t}{\eta_t}^\top\right) \\ &= \sum_{t\in\mathcal{T}_+} \eta_t^2 \left(\operatorname{tr}(\boldsymbol{X}^\top\boldsymbol{X}) - \sum_{i\in[K]} \operatorname{tr}\left(\boldsymbol{X}^\top\boldsymbol{X}\frac{\boldsymbol{a}_i^t}{\eta_t}\frac{\boldsymbol{a}_i^t}{\eta_t}^\top\right)\right). \end{aligned}$$

Thus, there exists an index $t^\star \in \mathcal{T}_+$ such that

$$\mathrm{tr}(\boldsymbol{X}^\top \boldsymbol{X}) - \sum_{i\in[K]} \mathrm{tr}(\boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{V}_{ii}) \geq \mathrm{tr}(\boldsymbol{X}^\top \boldsymbol{X}) - \sum_{i\in[K]} \mathrm{tr}\left(\boldsymbol{X}^\top \boldsymbol{X} \frac{\boldsymbol{a}_i^{t^\star}}{\eta_{t^\star}} \frac{\boldsymbol{a}_i^{t^\star}}{\eta_{t^\star}}^\top\right).$$

As the solution $(\hat{\boldsymbol{u}}_i)_{i\in[K]}$ in (13) corresponding to $t = t^\star$ is feasible in (6), we conclude that the optimal value of (5) constitutes an upper bound on the optimal value of (6). Our previous argument that (5) is a relaxation of (6) thus implies that both problems are indeed equivalent. This completes the proof. $\square$

By employing the same reformulation techniques as in the proof of Theorem 1, we can show that the generic problem (2) is amenable to an exact convex reformulation.

Proposition 1. *Problem* (2) *is equivalent to the following completely positive program.*

$$\begin{array}{lll}
\min & \mathrm{tr}(\boldsymbol{X}^\top \boldsymbol{X}) + \sum_{i\in[K]} \mathrm{tr}(\boldsymbol{G}_{ii} - 2\boldsymbol{X}\boldsymbol{W}_{ii}) & \\
\text{s.t.} & \boldsymbol{p}_i \in \mathbb{SOC}_+^{N+1} \times \mathbb{R}_+^D,\ \boldsymbol{Q}_{ij} \in \mathbb{R}_+^{(N+1+D)\times(N+1+D)} & \forall i,j \in [K] \\
& \boldsymbol{u}_i \in \mathbb{R}_+^N,\ \boldsymbol{V}_{ij} \in \mathbb{R}_+^{N\times N},\ \boldsymbol{h}_i \in \mathbb{R}_+^D,\ \boldsymbol{G}_{jj} \in \mathbb{R}_+^{D\times D},\ \boldsymbol{W}_{ij} \in \mathbb{R}_+^{N\times D} & \forall i,j \in [K] \\
& \boldsymbol{p}_i = \begin{bmatrix} \boldsymbol{u}_i \\ 1 \\ \boldsymbol{h}_i \end{bmatrix},\ \boldsymbol{Q}_{ij} = \begin{bmatrix} \boldsymbol{V}_{ij} & \boldsymbol{u}_i & \boldsymbol{W}_{ij} \\ \boldsymbol{u}_j^\top & 1 & \boldsymbol{h}_j^\top \\ \boldsymbol{W}_{ji}^\top & \boldsymbol{h}_i & \boldsymbol{G}_{ij} \end{bmatrix} & \forall i,j \in [K] \\
& \mathrm{tr}(\boldsymbol{V}_{ii}) = 1 & \forall i \in [K] \\
& \mathrm{tr}(\boldsymbol{V}_{ij}) = 0 & \forall i,j \in [K] : i \neq j \\
& \begin{bmatrix} \boldsymbol{Q}_{11} & \cdots & \boldsymbol{Q}_{1K} & \boldsymbol{p}_1 \\ \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{Q}_{K1} & \cdots & \boldsymbol{Q}_{KK} & \boldsymbol{p}_K \\ \boldsymbol{p}_1^\top & \cdots & \boldsymbol{p}_K^\top & 1 \end{bmatrix} \in \mathcal{C}\left((\mathbb{SOC}_+^{N+1} \times \mathbb{R}_+^D)^K \times \mathbb{R}_+\right) &
\end{array} \tag{14}$$

3 K-means Clustering

Building upon the results in the previous section, we now derive an exact completely positive programming reformulation for the K-means clustering problem (1). To this end, we note that the problem can equivalently be solved via the following mixed-integer nonlinear program [12].

$$\begin{array}{rlll}
Z^\star = & \min & \sum_{i\in[K]} \sum_{n:\pi_{in}=1} \|\boldsymbol{x}_n - \boldsymbol{c}_i\|^2 & \\
& \text{s.t.} & \boldsymbol{\pi}_i \in \{0,1\}^N,\ \boldsymbol{c}_i \in \mathbb{R}^D & \forall i \in [K] \\
& & \boldsymbol{c}_i = \frac{1}{\mathbf{e}^\top \boldsymbol{\pi}_i} \sum_{n:\pi_{in}=1} \boldsymbol{x}_n & \forall i \in [K] \\
& & \mathbf{e}^\top \boldsymbol{\pi}_i \geq 1 & \forall i \in [K] \\
& & \sum_{i\in[K]} \boldsymbol{\pi}_i = \mathbf{e} &
\end{array} \tag{15}$$

Here, $\boldsymbol{c}_i$ is the centroid of the i-th cluster, while $\boldsymbol{\pi}_i$ is the assignment vector for the i-th cluster, *i.e.*, $\pi_{in} = 1$ if and only if the data point $\boldsymbol{x}_n$ is assigned to the cluster i. The last constraint in (15) ensures that each data point is assigned to a cluster, while the constraint system in the penultimate row ensures that there are exactly K clusters. We now show that we can solve the K-means clustering problem by solving a modified problem (4) with an additional constraint $\sum_{i\in[K]} \boldsymbol{u}_i\boldsymbol{u}_i^\top \mathbf{e} = \mathbf{e}$.

Theorem 2. *The following non-convex program solves the K-means clustering problem.*

$$\begin{array}{rll} Z^\star = \min & \operatorname{tr}(\boldsymbol{X}^\top\boldsymbol{X}) - \sum_{i\in[K]} \operatorname{tr}(\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{u}_i\boldsymbol{u}_i^\top) & \\ \text{s.t.} & \boldsymbol{U} \in \mathbb{R}_+^{N\times K} & \\ & \boldsymbol{u}_i^\top\boldsymbol{u}_i = 1 & \forall i \in [K] \\ & \boldsymbol{u}_i^\top\boldsymbol{u}_j = 0 & \forall i,j \in [K] : i \neq j \\ & \sum_{i\in[K]} \boldsymbol{u}_i\boldsymbol{u}_i^\top\mathbf{e} = \mathbf{e} & \end{array} \tag{$\mathcal{Z}$}$$

Proof. We first observe that the centroids in (15) can be expressed as

$$\boldsymbol{c}_i = \frac{1}{\mathbf{e}^\top\boldsymbol{\pi}_i} \sum_{n\in[N]} \pi_{in}\boldsymbol{x}_n \qquad \forall i \in [K].$$

Substituting these terms into the objective function and expanding the squared norm yield

$$\begin{aligned} \sum_{i\in[K]} \sum_{n:\pi_{in}=1} \|\boldsymbol{x}_n - \boldsymbol{c}_i\|^2 &= \sum_{i\in[K]} \sum_{n\in[N]} \pi_{in}\|\boldsymbol{x}_n - \boldsymbol{c}_i\|^2 \\ &= \left(\sum_{n\in[N]} \|\boldsymbol{x}_n\|^2\right) - \left(\sum_{i\in[K]} \frac{1}{\mathbf{e}^\top\boldsymbol{\pi}_i} \sum_{p,q\in[N]} \pi_{ip}\pi_{iq}\boldsymbol{x}_p^\top\boldsymbol{x}_q\right) \\ &= \operatorname{tr}(\boldsymbol{X}^\top\boldsymbol{X}) - \sum_{i\in[K]} \frac{1}{\mathbf{e}^\top\boldsymbol{\pi}_i}\operatorname{tr}(\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{\pi}_i\boldsymbol{\pi}_i^\top). \end{aligned}$$

Thus, (15) can be simplified into the problem

$$\begin{array}{lll} \min & \operatorname{tr}(\boldsymbol{X}^\top\boldsymbol{X}) - \sum_{i\in[K]} \frac{1}{\mathbf{e}^\top\boldsymbol{\pi}_i}\operatorname{tr}(\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{\pi}_i\boldsymbol{\pi}_i^\top) & \\ \text{s.t.} & \boldsymbol{\pi}_i \in \{0,1\}^N & \forall i \in [K] \\ & \mathbf{e}^\top\boldsymbol{\pi}_i \geq 1 & \forall i \in [K] \\ & \sum_{i\in[K]} \boldsymbol{\pi}_i = \mathbf{e}. & \end{array} \tag{16}$$

For any feasible solution $(\boldsymbol{\pi}_i)_{i\in[K]}$ to (16) we define the vectors $(\boldsymbol{u}_i)_{i\in[K]}$ that satisfy

$$\boldsymbol{u}_i = \frac{\boldsymbol{\pi}_i}{\sqrt{\mathbf{e}^\top\boldsymbol{\pi}_i}} \quad \forall i \in [K].$$

We argue that the solution $(\boldsymbol{u}_i)_{i\in[K]}$ is feasible to ($\mathcal{Z}$) and yields the same objective value. Indeed, we have

$$\boldsymbol{u}_i^\top\boldsymbol{u}_i = \frac{\boldsymbol{\pi}_i^\top\boldsymbol{\pi}_i}{\mathbf{e}^\top\boldsymbol{\pi}_i} = 1 \quad \forall i \in [K]$$

because $\boldsymbol{\pi}_i \in \{0,1\}^N$ and $\mathbf{e}^\top \boldsymbol{\pi}_i \geq 1$ for all $i \in [K]$. We also have

$$\sum_{i\in[K]} \boldsymbol{u}_i \boldsymbol{u}_i^\top \mathbf{e} = \sum_{i\in[K]} \frac{\boldsymbol{\pi}_i}{\sqrt{\mathbf{e}^\top \boldsymbol{\pi}_i}} \frac{\mathbf{e}^\top \boldsymbol{\pi}_i}{\sqrt{\mathbf{e}^\top \boldsymbol{\pi}_i}} = \mathbf{e},$$

and

$$\boldsymbol{u}_i^\top \boldsymbol{u}_j = 0 \quad \forall i, j \in [K] : i \neq j$$

since the constraint $\sum_{i\in[K]} \boldsymbol{\pi}_i = \mathbf{e}$ in (16) ensures that each data point is assigned to at most 1 cluster. Verifying the objective value of this solution, we obtain

$$\operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X}) - \sum_{i\in[K]} \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{u}_i \boldsymbol{u}_i^\top) = \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X}) - \sum_{i\in[K]} \frac{1}{\mathbf{e}^\top \boldsymbol{\pi}_i} \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{\pi}_i \boldsymbol{\pi}_i^\top).$$

Thus, we conclude that problem ($\mathcal{Z}$) constitutes a relaxation of (16).

To show that ($\mathcal{Z}$) is indeed an exact reformulation, consider any feasible solution $(\boldsymbol{u}_i)_{i\in[K]}$ to this problem. For any fixed $i, j \in [K]$, the complementary constraint $\boldsymbol{u}_i^\top \boldsymbol{u}_j = 0$ in ($\mathcal{Z}$) means that

$$u_{in} > 0 \Longrightarrow u_{jn} = 0 \quad \text{and} \quad u_{jn} > 0 \Longrightarrow u_{in} = 0 \quad \text{for all} \quad n \in [N].$$

Thus, in view of the last constraint in ($\mathcal{Z}$), we must have $\boldsymbol{u}_i \in \{0, 1/\boldsymbol{u}_i^\top \mathbf{e}\}^N$ for every $i \in [K]$. Using this observation, we next define the binary vectors $(\boldsymbol{\pi}_i)_{i\in[K]}$ that satisfy

$$\boldsymbol{\pi}_i = \boldsymbol{u}_i \boldsymbol{u}_i^\top \mathbf{e} \in \{0,1\}^N \qquad \forall i \in [K].$$

For every $i \in [K]$, we find that $\mathbf{e}^\top \boldsymbol{\pi}_i \geq 1$ since $\boldsymbol{u}_i^\top \boldsymbol{u}_i = 1$. Furthermore, we have

$$\sum_{i\in[K]} \boldsymbol{\pi}_i = \sum_{i\in[K]} \boldsymbol{u}_i \boldsymbol{u}_i^\top \mathbf{e} = \mathbf{e}.$$

Substituting the constructed solution $(\boldsymbol{\pi}_i)_{i\in[K]}$ into the objective function of (16), we obtain

$$\begin{aligned} \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X}) - \sum_{i\in[K]} \frac{1}{\mathbf{e}^\top \boldsymbol{\pi}_i} \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{\pi}_i \boldsymbol{\pi}_i^\top) &= \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X}) - \sum_{i\in[K]} \frac{(\boldsymbol{u}_i^\top \mathbf{e})^2}{\mathbf{e}^\top \boldsymbol{u}_i \boldsymbol{u}_i^\top \mathbf{e}} \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{u}_i \boldsymbol{u}_i^\top) \\ &= \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X}) - \sum_{i\in[K]} \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{u}_i \boldsymbol{u}_i^\top). \end{aligned}$$

Thus, any feasible solution to ($\mathcal{Z}$) can be used to construct a feasible solution to (16) that yields the same objective value. Our previous argument that (16) is a relaxation of ($\mathcal{Z}$) then implies that both problems are indeed equivalent. This completes the proof. □

Remark 1. *The constraint $\sum_{i\in[K]} \boldsymbol{u}_i \boldsymbol{u}_i^\top \mathbf{e} = \mathbf{e}$ in ($\mathcal{Z}$) ensures that there are no fractional values in the resulting cluster assignment vectors $(\boldsymbol{\pi}_i)_{i\in[K]}$. While the formulation (4) is only applicable for instances of ONMF with non-negative input data $\boldsymbol{X}$, the reformulation ($\mathcal{Z}$) remains valid for any instances of K-means clustering, even if the input data matrix $\boldsymbol{X}$ contains negative elements.*

Remark 2. *We can reformulate the objective function of* ($\mathcal{Z}$) *as* $\frac{1}{2}\text{tr}\left(\boldsymbol{D}\sum_{i\in[K]}\boldsymbol{u}_i\boldsymbol{u}_i^\top\right)$, *where* $\boldsymbol{D}$ *is the matrix with elements* $D_{pq} = \|\boldsymbol{x}_p - \boldsymbol{x}_q\|^2$, $p, q \in [N]$. *To obtain this reformulation, define* $\boldsymbol{Y} = \sum_{i\in[K]}\boldsymbol{u}_i\boldsymbol{u}_i^\top$. *Then we have*

$$\begin{aligned}
\frac{1}{2}\text{tr}(\boldsymbol{D}\boldsymbol{Y}) &= \frac{1}{2}\sum_{p,q\in[N]}\|\boldsymbol{x}_p - \boldsymbol{x}_q\|^2 Y_{pq} \\
&= \frac{1}{2}\sum_{p,q\in[N]}\left(\boldsymbol{x}_p^\top\boldsymbol{x}_p + \boldsymbol{x}_q^\top\boldsymbol{x}_q - 2\boldsymbol{x}_p^\top\boldsymbol{x}_q\right)Y_{pq} \\
&= \frac{1}{2}\left(2\sum_{p\in[N]}\sum_{q\in[N]}\boldsymbol{x}_p^\top\boldsymbol{x}_p Y_{pq}\right) - \sum_{p,q\in[N]}\boldsymbol{x}_p^\top\boldsymbol{x}_q Y_{pq} \\
&= \left(\sum_{p\in[N]}\boldsymbol{x}_p^\top\boldsymbol{x}_p\right) - \left(\sum_{p,q\in[N]}\boldsymbol{x}_p^\top\boldsymbol{x}_q Y_{pq}\right) = \text{tr}(\boldsymbol{X}^\top\boldsymbol{X}) - \text{tr}(\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{Y}).
\end{aligned}$$

Here, the fourth equality holds because of the last constraint of ($\mathcal{Z}$) *which ensures that* $\sum_{q\in[N]}Y_{pq} = 1$ *for all* $p \in [N]$.

We are now well-positioned to derive an equivalent completely positive program for the K-means clustering problem. To simplify our notation we will employ the sets

$$\mathcal{U}(N,K) = \left\{\boldsymbol{U} \in \mathbb{R}_+^{N\times K}:\ \boldsymbol{u}_i^\top\boldsymbol{u}_i = 1\quad \forall i\in[K],\quad \boldsymbol{u}_i^\top\boldsymbol{u}_j = 0\quad \forall i,j\in[K]: i\neq j\ \right\}\quad \text{and}$$

$$\mathcal{V}(N,K) = \left\{(\boldsymbol{V}_{ij})_{i,j\in[K]} \in \mathbb{R}_+^{N^2\times K^2}:\ \text{tr}(\boldsymbol{V}_{ii}) = 1\quad \forall i\in[K],\quad \text{tr}(\boldsymbol{V}_{ij}) = 0\quad \forall i,j\in[K]: i\neq j\ \right\}$$

in all reformulations in the remainder of this section.

Theorem 3. *The following completely positive program solves the* K*-means clustering problem.*

$$\begin{aligned}
Z^\star = \quad \min\quad & \text{tr}(\boldsymbol{X}^\top\boldsymbol{X}) - \sum_{i\in[K]}\text{tr}(\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{V}_{ii}) \\
\text{s.t.}\quad & \boldsymbol{p}_i \in \mathbb{SOC}_+^{N+1}\times\mathbb{R}_+^{N+1},\ \boldsymbol{Q}_{ij}\in\mathbb{R}_+^{2(N+1)\times 2(N+1)},\ (\boldsymbol{V}_{ij})_{i,j\in[K]}\in\mathcal{V}(N,K) \\
& \boldsymbol{w}\in\mathbb{R}_+^K,\ z_{ij}\in\mathbb{R}_+,\ \boldsymbol{u}_i,\boldsymbol{s}_i,\boldsymbol{h}_{ij},\boldsymbol{r}_{ij}\in\mathbb{R}_+^N,\ \boldsymbol{Y}_{ij},\boldsymbol{G}_{ij}\in\mathbb{R}_+^{N\times N} \qquad \forall i,j\in[K] \\
& \boldsymbol{p}_i = \begin{bmatrix}\boldsymbol{u}_i\\ 1\\ \boldsymbol{s}_i\\ w_i\end{bmatrix},\ \boldsymbol{Q}_{ij} = \begin{bmatrix}\boldsymbol{V}_{ij} & \boldsymbol{u}_i & \boldsymbol{G}_{ij} & \boldsymbol{h}_{ij}\\ \boldsymbol{u}_j^\top & 1 & \boldsymbol{s}_j^\top & w_j\\ \boldsymbol{G}_{ji}^\top & \boldsymbol{s}_i & \boldsymbol{Y}_{ij} & \boldsymbol{r}_{ij}\\ \boldsymbol{h}_{ji}^\top & w_i & \boldsymbol{r}_{ji}^\top & z_{ij}\end{bmatrix} \qquad \forall i,j\in[K] \\
& \sum_{i\in[K]}\boldsymbol{V}_{ii}\mathbf{e} = \mathbf{e} \\
& \text{diag}(\boldsymbol{V}_{ii}) = \boldsymbol{h}_{ii},\ \ \boldsymbol{u}_i + \boldsymbol{s}_i = w_i\mathbf{e},\ \ \text{diag}(\boldsymbol{V}_{ii} + \boldsymbol{Y}_{ii} + 2\boldsymbol{G}_{ii}) + z_{ii}\mathbf{e} - 2\boldsymbol{h}_{ii} - 2\boldsymbol{r}_{ii} = 0\quad \forall i\in[K] \\
& \begin{bmatrix}\boldsymbol{Q}_{11} & \cdots & \boldsymbol{Q}_{1K} & \boldsymbol{p}_1\\ \vdots & \ddots & \vdots & \vdots\\ \boldsymbol{Q}_{K1} & \cdots & \boldsymbol{Q}_{KK} & \boldsymbol{p}_K\\ \boldsymbol{p}_1^\top & \cdots & \boldsymbol{p}_K^\top & 1\end{bmatrix} \in \mathcal{C}\left((\mathbb{SOC}_+^{N+1}\times\mathbb{R}_+^{N+1})^K\times\mathbb{R}_+\right)
\end{aligned} \qquad (\overline{\mathcal{Z}})$$

Proof. We consider the following equivalent reformulation of $(\mathcal{Z})$ with two additional strengthening constraint systems.

$$
\begin{array}{lll}
\min & \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X}) - \sum_{i\in[K]} \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{u}_i\boldsymbol{u}_i^\top) & \\
\text{s.t.} & \boldsymbol{U} \in \mathcal{U}(N,K),\ \boldsymbol{S} \in \mathbb{R}_+^{N\times K},\ \boldsymbol{w} \in \mathbb{R}_+^K & \\
& \sum_{i\in[K]} \boldsymbol{u}_i\boldsymbol{u}_i^\top \mathbf{e} = \mathbf{e} & \\
& \boldsymbol{u}_i \circ \boldsymbol{u}_i = w_i \boldsymbol{u}_i & \forall i \in [K] \\
& \boldsymbol{u}_i + \boldsymbol{s}_i = w_i \mathbf{e} & \forall i \in [K]
\end{array}
\tag{17}
$$

Here, since $\boldsymbol{s}_i \geq \mathbf{0}$, the last constraint system in (17) implies that $\boldsymbol{u}_i \leq w_i\mathbf{e}$, while the penultimate constraint system ensures that $\boldsymbol{u}_i$ is a binary vector, *i.e.*, $\boldsymbol{u}_i \in \{0, w_i\}^N$ for some $w_i \in \mathbb{R}_+$. Since any feasible solution to $(\mathcal{Z})$ satisfies these conditions, we may thus conclude that the problems $(\mathcal{Z})$ and (17) are indeed equivalent. As we will see below, the exactness of the completely positive programming reformulation is reliant on these two redundant constraint systems.

Next, as in the proof of Theorem 1, we consider an equivalent rank-1 reformulation of (17) given by

$$
\begin{array}{lll}
\min & \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X}) - \sum_{i\in[K]} \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{V}_{ii}) & \\
\text{s.t.} & \boldsymbol{p}_i \in \mathbb{SOC}_+^{N+1} \times \mathbb{R}_+^{N+1},\ \boldsymbol{Q}_{ij} \in \mathbb{R}_+^{2(N+1)\times 2(N+1)},\ (\boldsymbol{V}_{ij})_{i,j\in[K]} \in \mathcal{V}(N,K) & \\
& \boldsymbol{w} \in \mathbb{R}_+^K,\ z_{ij} \in \mathbb{R}_+,\ \boldsymbol{u}_i, \boldsymbol{s}_i, \boldsymbol{h}_{ij}, \boldsymbol{r}_{ij} \in \mathbb{R}_+^N,\ \boldsymbol{Y}_{ij}, \boldsymbol{G}_{ij} \in \mathbb{R}_+^{N\times N} & \forall i,j \in [K] \\
& \boldsymbol{p}_i = \begin{bmatrix} \boldsymbol{u}_i \\ 1 \\ \boldsymbol{s}_i \\ w_i \end{bmatrix},\ \boldsymbol{Q}_{ij} = \begin{bmatrix} \boldsymbol{V}_{ij} & \boldsymbol{u}_i & \boldsymbol{G}_{ij} & \boldsymbol{h}_{ij} \\ \boldsymbol{u}_j^\top & 1 & \boldsymbol{s}_j^\top & w_j \\ \boldsymbol{G}_{ji}^\top & \boldsymbol{s}_i & \boldsymbol{Y}_{ij} & \boldsymbol{r}_{ij} \\ \boldsymbol{h}_{ji}^\top & w_i & \boldsymbol{r}_{ji}^\top & z_{ij} \end{bmatrix} & \forall i,j \in [K] \\
& \sum_{i\in[K]} \boldsymbol{V}_{ii}\mathbf{e} = \mathbf{e} & \\
& \operatorname{diag}(\boldsymbol{V}_{ii}) = \boldsymbol{h}_{ii},\ \ \boldsymbol{u}_i + \boldsymbol{s}_i = w_i\mathbf{e},\ \ \operatorname{diag}(\boldsymbol{V}_{ii} + \boldsymbol{Y}_{ii} + 2\boldsymbol{G}_{ii}) + z_{ii}\mathbf{e} - 2\boldsymbol{h}_{ii} - 2\boldsymbol{r}_{ii} = 0 & \forall i \in [K] \\
& \begin{bmatrix} \boldsymbol{Q}_{11} & \cdots & \boldsymbol{Q}_{1K} & \boldsymbol{p}_1 \\ \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{Q}_{K1} & \cdots & \boldsymbol{Q}_{KK} & \boldsymbol{p}_K \\ \boldsymbol{p}_1^\top & \cdots & \boldsymbol{p}_K^\top & 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{p}_1 \\ \vdots \\ \boldsymbol{p}_K \\ 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{p}_1 \\ \vdots \\ \boldsymbol{p}_K \\ 1 \end{bmatrix}^\top. &
\end{array}
\tag{18}
$$

Here, we have introduced new decision variables $z_{ij} \in \mathbb{R}_+$, $\boldsymbol{h}_{ij}, \boldsymbol{r}_{ij} \in \mathbb{R}_+^N$, $\boldsymbol{Y}_{ij}, \boldsymbol{G}_{ij} \in \mathbb{R}_+^{N\times N}$, $i,j \in [K]$, that, in view of the last constraint in (18), must satisfy

$$
z_{ij} = w_i w_j,\ \boldsymbol{h}_{ij} = \boldsymbol{u}_i w_j,\ \boldsymbol{r}_{ij} = \boldsymbol{s}_i w_j,\ \boldsymbol{Y}_{ij} = \boldsymbol{s}_i \boldsymbol{s}_j^\top,\ \boldsymbol{G}_{ij} = \boldsymbol{u}_i \boldsymbol{s}_j^\top \qquad \forall i,j \in [K].
$$

The constraint system on the penultimate row of (18) arises from squaring the left-hand sides of the equalities

$$
u_{in} + s_{in} - w_i = 0 \qquad \forall i \in [K]\ \forall n \in [N],
$$

which correspond to the last constraint system in (17). Next, we relax the rank-1 condition in the last constraint of (18) with the requirement that the left-hand side matrix should merely belong to the completely positive cone $\mathcal{C}\left((\mathbb{SOC}_{+}^{N+1} \times \mathbb{R}_{+}^{N+1})^{K} \times \mathbb{R}_{+}\right)$. This yields the lower bounding problem $(\overline{\mathcal{Z}})$. To show that the emerging completely positive program is exact, we consider the completely positive decomposition given by

$$\begin{bmatrix} \boldsymbol{Q}_{11} & \cdots & \boldsymbol{Q}_{1K} & \boldsymbol{p}_1 \\ \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{Q}_{K1} & \cdots & \boldsymbol{Q}_{KK} & \boldsymbol{p}_K \\ \boldsymbol{p}_1^\top & \cdots & \boldsymbol{p}_K^\top & 1 \end{bmatrix} = \sum_{t\in\mathcal{T}} \begin{bmatrix} \boldsymbol{\theta}_1^t \\ \vdots \\ \boldsymbol{\theta}_K^t \\ \eta_t \end{bmatrix} \begin{bmatrix} \boldsymbol{\theta}_1^t \\ \vdots \\ \boldsymbol{\theta}_K^t \\ \eta_t \end{bmatrix}^\top .$$

As in the proof of Theorem 1, we partition $\mathcal{T}$ into the subsets $\mathcal{T}_+ = \{t \in \mathcal{T} : \eta_t > 0\}$ and $\mathcal{T}_0 = \{t \in \mathcal{T} : \eta_t = 0\}$, which results in the decomposition

$$\begin{bmatrix} \boldsymbol{Q}_{11} & \cdots & \boldsymbol{Q}_{1K} & \boldsymbol{p}_1 \\ \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{Q}_{K1} & \cdots & \boldsymbol{Q}_{KK} & \boldsymbol{p}_K \\ \boldsymbol{p}_1^\top & \cdots & \boldsymbol{p}_K^\top & 1 \end{bmatrix} = \sum_{t\in\mathcal{T}_+} \eta_t^2 \begin{bmatrix} \frac{\boldsymbol{\theta}_1^t}{\eta_t} \\ \vdots \\ \frac{\boldsymbol{\theta}_K^t}{\eta_t} \\ 1 \end{bmatrix} \begin{bmatrix} \frac{\boldsymbol{\theta}_1^t}{\eta_t} \\ \vdots \\ \frac{\boldsymbol{\theta}_K^t}{\eta_t} \\ 1 \end{bmatrix}^\top + \sum_{t\in\mathcal{T}_0} \begin{bmatrix} \boldsymbol{\theta}_1^t \\ \vdots \\ \boldsymbol{\theta}_K^t \\ 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\theta}_1^t \\ \vdots \\ \boldsymbol{\theta}_K^t \\ 0 \end{bmatrix}^\top . \tag{19}$$

For every fixed $i \in [K]$ and $t \in \mathcal{T}$, we can further partition the vector $\boldsymbol{\theta}_i^t$ into the components $(\boldsymbol{a}_i^t, b_i^t) \in \mathbb{SOC}_{+}^{N+1}$, $\boldsymbol{c}_i^t \in \mathbb{R}_{+}^{N}$, and $d_i^t \in \mathbb{R}_+$ such that

$$\boldsymbol{\theta}_i^t = \begin{bmatrix} (\boldsymbol{a}_i^t)^\top & b_i^t & (\boldsymbol{c}_i^t)^\top & d_i^t \end{bmatrix}^\top .$$

The argument in the proof of Theorem 1 allows us to conclude that

$$\boldsymbol{a}_i^t = \mathbf{0} \quad \forall t \in \mathcal{T}_0.$$

Furthermore, for any fixed $t \in \mathcal{T}_+$, the solution $(\hat{\boldsymbol{u}}_i)_{i\in[K]}$ satisfying

$$\hat{\boldsymbol{u}}_i = \frac{\boldsymbol{a}_i^t}{\eta_t} \qquad \forall i \in [K] \tag{20}$$

is feasible to the constraint systems $\boldsymbol{u}_i^\top \boldsymbol{u}_i = 1$, $i \in [K]$, and $\boldsymbol{u}_i^\top \boldsymbol{u}_j = 0$, $i, j \in [K] : i \neq j$, in $(\mathcal{Z})$.

Next, we show that the solution (20) is also feasible to the constraint $\sum_{i\in[K]} \boldsymbol{u}_i \boldsymbol{u}_i^\top \mathbf{e} = \mathbf{e}$ in $(\mathcal{Z})$. To this end, since $\boldsymbol{a}_i^t = \mathbf{0}$ for all $t \in \mathcal{T}_0$, we will employ the index set $\mathcal{T}_+$ in place of $\mathcal{T}$ in all subsequent derivations. From Lemma 2, the constraints $\boldsymbol{u}_i + \boldsymbol{s}_i = w_i \mathbf{e}$ and $\operatorname{diag}(\boldsymbol{V}_{ii} + \boldsymbol{Y}_{ii} + 2\boldsymbol{G}_{ii}) + z_{ii}\mathbf{e} - 2\boldsymbol{h}_{ii} - 2\boldsymbol{r}_{ii} = 0$ in $(\overline{\mathcal{Z}})$ imply that

$$\frac{\boldsymbol{a}_i^t}{\eta_t} \leq \frac{d_i^t}{\eta_t}\mathbf{e} \quad \forall t \in \mathcal{T}_+. \tag{21}$$

Furthermore, the constraint $\operatorname{diag}(\boldsymbol{V}_{ii}) = \boldsymbol{h}_{ii}$ in $(\overline{\mathcal{Z}})$ and the decomposition (19) yield

$$\sum_{t\in\mathcal{T}_+} \eta_t^2 \frac{\boldsymbol{a}_i^t}{\eta_t} \circ \frac{\boldsymbol{a}_i^t}{\eta_t} = \sum_{t\in\mathcal{T}_+} \eta_t^2 \frac{\boldsymbol{a}_i^t}{\eta_t} \frac{d_i^t}{\eta_t}.$$

Rearranging the terms, we obtain equivalently

$$0 = \sum_{t \in \mathcal{T}_+} \eta_t^2 \frac{\boldsymbol{a}_i^t}{\eta_t} \circ \left(\frac{d_i^t}{\eta_t} \mathbf{e} - \frac{\boldsymbol{a}_i^t}{\eta_t} \right).$$

The inequalities in (21) and the fact that $\boldsymbol{a}_i^t/\eta_t \geq \mathbf{0}$, $t \in \mathcal{T}_+$, then imply that each summand on the right-hand side must be equal to 0. Thus,

$$\frac{\boldsymbol{a}_i^t}{\eta_t} \circ \frac{\boldsymbol{a}_i^t}{\eta_t} = \frac{d_i^t}{\eta_t} \frac{\boldsymbol{a}_i^t}{\eta_t} \quad \forall t \in \mathcal{T}_+ \implies \frac{\boldsymbol{a}_i^t}{\eta_t} \in \left\{ 0, \frac{d_i^t}{\eta_t} \right\}^N \quad \forall t \in \mathcal{T}_+. \tag{22}$$

Observe that since $(\boldsymbol{a}_i^t/\eta_t)^\top \boldsymbol{a}_i^t/\eta_t = 1$, we must have $d_i^t/\eta_t = 1/\sqrt{\#(\boldsymbol{a}_i^t/\eta_t)}$. Next, the constraint $\sum_{i \in [K]} \boldsymbol{V}_{ii} \mathbf{e} = \mathbf{e}$ in $(\overline{\mathcal{Z}})$ and the decomposition (19) yield

$$\sum_{i \in [K]} \sum_{t \in \mathcal{T}_+} \eta_t^2 \frac{\boldsymbol{a}_i^t}{\eta_t} \left(\frac{\boldsymbol{a}_i^t}{\eta_t} \right)^\top \mathbf{e} = \sum_{t \in \mathcal{T}_+} \eta_t^2 \left(\sum_{i \in [K]} \frac{\boldsymbol{a}_i^t}{\eta_t} \left(\frac{\boldsymbol{a}_i^t}{\eta_t} \right)^\top \mathbf{e} \right) = \mathbf{e}. \tag{23}$$

Our previous argument in (22) however implies that for every $t \in \mathcal{T}_+$ we have

$$\frac{\boldsymbol{a}_i^t}{\eta_t} \left(\frac{\boldsymbol{a}_i^t}{\eta_t} \right)^\top \mathbf{e} = \frac{\boldsymbol{a}_i^t}{\eta_t} \frac{d_i^t}{\eta_t} \times \# \frac{\boldsymbol{a}_i^t}{\eta_t} = \frac{\boldsymbol{a}_i^t}{\eta_t} \frac{d_i^t}{\eta_t} \left(\frac{\eta_t}{d_i^t} \right)^2 = \frac{\boldsymbol{a}_i^t}{\eta_t} \frac{\eta_t}{d_i^t} \in \{0, 1\}^N.$$

Since $(\boldsymbol{a}_i^t/\eta_t)^\top (\boldsymbol{a}_j^t/\eta_t) = 0$ for all $i, j \in [K]$ such that $i \neq j$, the equation (23) then implies that

$$\sum_{i \in [K]} \frac{\boldsymbol{a}_i^t}{\eta_t} \left(\frac{\boldsymbol{a}_i^t}{\eta_t} \right)^\top \mathbf{e} \leq \mathbf{e} \qquad \forall t \in \mathcal{T}_+.$$

These inequalities are in fact binding because $\sum_{t \in \mathcal{T}_+} \eta_t^2 = 1$ in (23). Thus, for any fixed $t \in \mathcal{T}_+$ the solution $(\hat{\boldsymbol{u}}_i)_{i \in [K]}$ in (20) is also feasible to the constraint $\sum_{i \in [K]} \boldsymbol{u}_i \boldsymbol{u}_i^\top \mathbf{e} = \mathbf{e}$ in $(\mathcal{Z})$.

In summary, we find that for any $t \in \mathcal{T}_+$ the solution $(\hat{\boldsymbol{u}}_i)_{i \in [K]}$ in (20) is feasible to $(\mathcal{Z})$. The claim then immediately follows from employing the same convex hull argument as in the proof of Theorem 1. □

4 Approximation Algorithm for K-means Clustering

In this section, we develop a new approximation algorithm for the K-means clustering problem. To this end, we observe that in the reformulation $(\overline{\mathcal{Z}})$ the difficulty of the original problem is now entirely absorbed in the completely positive cone $\mathcal{C}(\cdot)$ which has been well studied in the literature [5, 9, 7]. Any such completely positive program admits the hierarchy of increasingly accurate SDP relaxations which are obtained by replacing the completely positive cone with progressively tighter semidefinite representable outer approximations [23, 9, 17]. For the completely positive program $(\overline{\mathcal{Z}})$, however, even if the simplest outer approximation is employed, the size of the resulting semidefinite program can be relatively substantial and may prevent efficient solution schemes. In order to alleviate this intractability, we propose to perform the relaxation on a simpler completely positive program where we have eliminated the constraints on the penultimate row of $(\overline{\mathcal{Z}})$.

Proposition 2. *The optimal value of the following completely positive program constitutes a lower bound on $Z^\star$.*

$$
\begin{array}{rll}
R_0^\star = & \min & \mathrm{tr}(\boldsymbol{X}^\top\boldsymbol{X}) - \sum_{i\in[K]} \mathrm{tr}(\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{V}_{ii}) \\
& \text{s.t.} & \boldsymbol{p}_i \in \mathbb{SOC}_+^{N+1},\ \boldsymbol{Q}_{ij} \in \mathbb{R}_+^{(N+1)\times(N+1)},\ \boldsymbol{u}_i \in \mathbb{R}_+^N,\ \boldsymbol{V}_{ij} \in \mathbb{R}_+^{N\times N} \quad \forall i,j \in [K] \\
& & \boldsymbol{p}_i = \begin{bmatrix} \boldsymbol{u}_i \\ 1 \end{bmatrix},\ \boldsymbol{Q}_{ij} = \begin{bmatrix} \boldsymbol{V}_{ij} & \boldsymbol{u}_i \\ \boldsymbol{u}_j^\top & 1 \end{bmatrix} \quad \forall i,j \in [K] \\
& & \mathrm{tr}(\boldsymbol{V}_{ii}) = 1 \quad \forall i \in [K] \\
& & \mathrm{tr}(\boldsymbol{V}_{ij}) = 0 \quad \forall i,j \in [K] : i \neq j \\
& & \sum_{i\in[K]} \boldsymbol{V}_{ii}\mathbf{e} = \mathbf{e} \\
& & \begin{bmatrix} \boldsymbol{Q}_{11} & \cdots & \boldsymbol{Q}_{1K} & \boldsymbol{p}_1 \\ \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{Q}_{K1} & \cdots & \boldsymbol{Q}_{KK} & \boldsymbol{p}_K \\ \boldsymbol{p}_1^\top & \cdots & \boldsymbol{p}_K^\top & 1 \end{bmatrix} \in \mathcal{C}\left((\mathbb{SOC}_+^{N+1})^K \times \mathbb{R}_+\right)
\end{array}
\tag{$\mathcal{R}_0$}
$$

Remark 3. *The completely positive program* ($\mathcal{R}_0$) *corresponds to the reformulation* (5) *for ONMF with the additional constraint* $\sum_{i\in[K]} \boldsymbol{V}_{ii}\mathbf{e} = \mathbf{e}$.

We are now ready to derive an SDP relaxation for the K-means clustering problem. The relaxation is obtained by replacing the completely positive cone $\mathcal{C}\left((\mathbb{SOC}_+^{N+1})^K \times \mathbb{R}_+\right)$ in ($\mathcal{R}_0$) with its coarsest outer approximation [27] which is given by the cone

$$
\left\{\boldsymbol{M} \in \mathbb{S}^{K(N+1)+1} : \boldsymbol{M} \succeq \mathbf{0},\ \boldsymbol{M} \geq \mathbf{0},\ \mathrm{tr}(\mathbb{J}_i\boldsymbol{M}) \geq 0 \ \ i \in [K]\right\},
$$

where

$$
\begin{aligned}
\mathbb{J}_1 &= \mathrm{diag}\left([-\mathbf{e}^\top, 1, \mathbf{0}^\top, 0, \cdots, \mathbf{0}^\top, 0, 0]^\top\right), \\
\mathbb{J}_2 &= \mathrm{diag}\left([\mathbf{0}^\top, 0, -\mathbf{e}^\top, 1, \cdots, \mathbf{0}^\top, 0, 0]^\top\right), \\
&\cdots \\
\mathbb{J}_K &= \mathrm{diag}\left([\mathbf{0}^\top, 0, -\mathbf{0}^\top, 0, \cdots, \mathbf{e}^\top, 1, 0]^\top\right).
\end{aligned}
$$

If $\boldsymbol{M}$ has the structure of the large matrix in ($\mathcal{R}_0$) then the constraint $\mathrm{tr}(\mathbb{J}_i\boldsymbol{M}) \geq 0$ reduces to $\mathrm{tr}(\boldsymbol{V}_{ii}) \leq 1$, which is redundant and can safely be omitted in view of the stronger equality constraint $\mathrm{tr}(\boldsymbol{V}_{ii}) = 1$ in ($\mathcal{R}_0$). In this case, the outer approximation can be simplified to the cone of doubly non-negative matrices given by

$$
\left\{\boldsymbol{M} \in \mathbb{S}^{K(N+1)+1} : \boldsymbol{M} \succeq \mathbf{0},\ \boldsymbol{M} \geq \mathbf{0}\right\}.
$$

Proposition 3. *The following SDP constitutes a valid relaxation to the K-means clustering problem.*

$$\begin{array}{rlll} R_1^\star = & \min & \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X}) - \sum_{i\in[K]} \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{V}_i) & \\ & \text{s.t.} & \boldsymbol{V}_i \in \mathbb{R}_+^{N\times N} & \forall i \in [K] \\ & & \operatorname{tr}(\boldsymbol{V}_i) = 1 & \forall i \in [K] \\ & & \sum_{i\in[K]} \boldsymbol{V}_i \mathbf{e} = \mathbf{e} & \\ & & \boldsymbol{V}_i \succeq \mathbf{0} & \forall i \in [K] \\ & & \mathbf{e}_1^\top \boldsymbol{V}_1 \mathbf{e} = 1 & \end{array} \tag{$\mathcal{R}_1$}$$

Proof. Without loss of generality, we can assign the first data point $\boldsymbol{x}_1$ to the first cluster. The argument in the proof of Theorem 2 indicates that the assignment vector for the first cluster is given by

$$\boldsymbol{\pi}_1 = \boldsymbol{u}_1 \boldsymbol{u}_1^\top \mathbf{e} = \boldsymbol{V}_{11} \mathbf{e}.$$

Thus, the data point $\boldsymbol{x}_1$ is assigned to the first cluster if and only if the first element of $\boldsymbol{\pi}_1$ is equal to 1, *i.e.*, $1 = \mathbf{e}_1^\top \boldsymbol{\pi}_1 = \mathbf{e}_1^\top \boldsymbol{V}_{11}\mathbf{e}$. Henceforth, we shall add this constraint to $(\mathcal{R}_0)$. While the constraint is redundant for the completely positive program $(\mathcal{R}_0)$, it will cut-off any symmetric solution in the resulting SDP relaxation.

We next replace the completely positive cone in $(\mathcal{R}_0)$ with the corresponding cone of doubly non-negative matrices. This yields the following SDP relaxation.

$$\begin{array}{lll} \min & \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X}) - \sum_{i\in[K]} \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{V}_{ii}) & \\ \text{s.t.} & \boldsymbol{p}_i \in \mathbb{SOC}_+^{N+1},\ \boldsymbol{Q}_{ij} \in \mathbb{R}_+^{(N+1)\times(N+1)},\ \boldsymbol{u}_i \in \mathbb{R}_+^N,\ \boldsymbol{V}_{ij} \in \mathbb{R}_+^{N\times N} & \forall i,j \in [K] \\ & \boldsymbol{p}_i = \begin{bmatrix} \boldsymbol{u}_i \\ 1 \end{bmatrix},\ \boldsymbol{Q}_{ij} = \begin{bmatrix} \boldsymbol{V}_{ij} & \boldsymbol{u}_i \\ \boldsymbol{u}_j^\top & 1 \end{bmatrix} & \forall i,j \in [K] \\ & \operatorname{tr}(\boldsymbol{V}_{ii}) = 1 & \forall i \in [K] \\ & \operatorname{tr}(\boldsymbol{V}_{ij}) = 0 & \forall i,j \in [K] : i \neq j \\ & \sum_{i\in[K]} \boldsymbol{V}_{ii}\mathbf{e} = \mathbf{e} & \\ & \mathbf{e}_1^\top \boldsymbol{V}_{11}\mathbf{e} = 1 & \\ & \begin{bmatrix} \boldsymbol{Q}_{11} & \cdots & \boldsymbol{Q}_{1K} & \boldsymbol{p}_1 \\ \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{Q}_{K1} & \cdots & \boldsymbol{Q}_{KK} & \boldsymbol{p}_K \\ \boldsymbol{p}_1^\top & \cdots & \boldsymbol{p}_K^\top & 1 \end{bmatrix} \succeq \mathbf{0} & \end{array} \tag{24}$$

By substituting the definitions of $\boldsymbol{Q}_{ij}$ and $\boldsymbol{p}_i$, $i,j \in [K]$, and by the Schur complement, the last constraint of (24) can be reformulated as

$$\begin{bmatrix} \boldsymbol{V}_{11} & \boldsymbol{u}_1 & \cdots & \boldsymbol{V}_{1K} & \boldsymbol{u}_1 \\ \boldsymbol{u}_1^\top & 1 & \cdots & \boldsymbol{u}_K^\top & 1 \\ \vdots & \cdots & \ddots & \vdots & \vdots \\ \boldsymbol{V}_{K1} & \boldsymbol{u}_K & \cdots & \boldsymbol{V}_{KK} & \boldsymbol{u}_K \\ \boldsymbol{u}_1^\top & 1 & \cdots & \boldsymbol{u}_K^\top & 1 \end{bmatrix} \succeq \begin{bmatrix} \boldsymbol{u}_1 \\ 1 \\ \vdots \\ \boldsymbol{u}_K \\ 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_1 \\ 1 \\ \vdots \\ \boldsymbol{u}_K \\ 1 \end{bmatrix}^\top = \begin{bmatrix} \boldsymbol{u}_1\boldsymbol{u}_1^\top & \boldsymbol{u}_1 & \cdots & \boldsymbol{u}_1\boldsymbol{u}_K^\top & \boldsymbol{u}_1 \\ \boldsymbol{u}_1^\top & 1 & \cdots & \boldsymbol{u}_K^\top & 1 \\ \vdots & \cdots & \ddots & \vdots & \vdots \\ \boldsymbol{u}_K\boldsymbol{u}_1^\top & \boldsymbol{u}_K & \cdots & \boldsymbol{u}_K\boldsymbol{u}_K^\top & \boldsymbol{u}_K \\ \boldsymbol{u}_1^\top & 1 & \cdots & \boldsymbol{u}_K^\top & 1 \end{bmatrix},$$

which can further be simplified to

$$\begin{bmatrix} \boldsymbol{V}_{11} & 0 & \cdots & \boldsymbol{V}_{1K} & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & \cdots & \ddots & \vdots & \vdots \\ \boldsymbol{V}_{K1} & \mathbf{0} & \cdots & \boldsymbol{V}_{KK} & \mathbf{0} \\ \mathbf{0}^\top & 0 & \cdots & \mathbf{0}^\top & 0 \end{bmatrix} \succeq \begin{bmatrix} \boldsymbol{u}_1\boldsymbol{u}_1^\top & \mathbf{0} & \cdots & \boldsymbol{u}_1\boldsymbol{u}_K^\top & \mathbf{0} \\ \mathbf{0}^\top & 0 & \cdots & \mathbf{0}^\top & 0 \\ \vdots & \cdots & \ddots & \vdots & \vdots \\ \boldsymbol{u}_K\boldsymbol{u}_1^\top & \mathbf{0} & \cdots & \boldsymbol{u}_K\boldsymbol{u}_K^\top & \mathbf{0} \\ \mathbf{0}^\top & 0 & \cdots & \mathbf{0}^\top & 0 \end{bmatrix}. \tag{25}$$

Note that the resulting right-hand side matrix is positive semidefinite as it coincides with the outer product of the vector $[\boldsymbol{u}_1^\top \ 0 \ \cdots \ \boldsymbol{u}_K^\top \ 0]^\top$. As no other constraints in (24) involve the decision variables $(\boldsymbol{u}_i)_{i\in[K]}$, the feasible set of the problem is enlarged if the right-hand side matrix is equal to $\mathbf{0}$. Thus, since the objective function of (24) also does not involve these decision variables, we find that at optimality $\boldsymbol{u}_i = \mathbf{0}$, $i \in [K]$. Similarly, any such solution will remain feasible and yield the same objective value if we set $\boldsymbol{V}_{ij} = \mathbf{0}$, $i, j \in [K] : i \neq j$. In this case, the constraint in (25) can be reduced to the K semidefinite constraints

$$\boldsymbol{V}_{ii} \succeq \mathbf{0} \qquad \forall i \in [K].$$

Finally, by renaming the variable $\boldsymbol{V}_{ii}$ to $\boldsymbol{V}_i$ for all $i \in [K]$, we arrive at the desired semidefinite program $(\mathcal{R}_1)$. This completes the proof. □

The symmetry breaking constraint $\mathbf{e}_1^\top \boldsymbol{V}_1 \mathbf{e} = 1$ in $(\mathcal{R}_1)$ ensures that the solution $\boldsymbol{V}_1$ will be different from any of the solutions $\boldsymbol{V}_i$, $i \geq 2$. Specifically, the constraint $\sum_{i\in[K]} \boldsymbol{V}_i \mathbf{e} = \mathbf{e}$ in $(\mathcal{R}_1)$ along with the aforementioned symmetry breaking constraint implies that $\mathbf{e}_1^\top \boldsymbol{V}_i \mathbf{e} = 0$ for all $i \geq 2$. Thus, any rounding scheme that identifies the clusters using the solution $(\boldsymbol{V}_i)_{i\in[K]}$ will always assign the data point $\boldsymbol{x}_1$ to the first cluster. It can be shown, however, that there exists a *partially* symmetric optimal solution to $(\mathcal{R}_1)$ with $\boldsymbol{V}_2 = \cdots = \boldsymbol{V}_K$. This enables us to derive a further simplification to $(\mathcal{R}_1)$.

Corollary 1. *Problem* $(\mathcal{R}_1)$ *is equivalent to the semidefinite program given by*

$$\begin{aligned} R_1^\star = \quad \min \quad & \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X}) - \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{W}_1) - \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{W}_2) \\ \text{s.t.} \quad & \boldsymbol{W}_1, \boldsymbol{W}_2 \in \mathbb{R}_+^{N \times N} \\ & \operatorname{tr}(\boldsymbol{W}_1) = 1, \ \operatorname{tr}(\boldsymbol{W}_2) = K - 1 \\ & \boldsymbol{W}_1 \mathbf{e} + \boldsymbol{W}_2 \mathbf{e} = \mathbf{e} \\ & \boldsymbol{W}_1, \boldsymbol{W}_2 \succeq \mathbf{0} \\ & \mathbf{e}_1^\top \boldsymbol{W}_1 \mathbf{e} = 1. \end{aligned} \tag{$\overline{\mathcal{R}}_1$}$$

Proof. Any feasible solution $(\boldsymbol{V}_i)_{i \in [K]}$ to $(\mathcal{R}_1)$ can be used to construct a feasible solution $(\boldsymbol{W}_1, \boldsymbol{W}_2)$ to $(\overline{\mathcal{R}}_1)$ with the same objective value, as follows:

$$\boldsymbol{W}_1 = \boldsymbol{V}_1, \quad \boldsymbol{W}_2 = \sum_{i=2}^{K} \boldsymbol{V}_i.$$

Conversely, any feasible solution $(\boldsymbol{W}_1, \boldsymbol{W}_2)$ to $(\overline{\mathcal{R}}_1)$ can also be used to construct the following feasible solution to $(\mathcal{R}_1)$ with the same objective value.

$$\boldsymbol{V}_1 = \boldsymbol{W}_1, \quad \boldsymbol{V}_i = \frac{1}{K-1} \boldsymbol{W}_i \qquad \forall i = 1, \ldots, K-1$$

Thus, the claim follows. □

As it is a relaxation, the optimal value of the semidefinite program $(\mathcal{R}_1)$ constitutes a lower bound on the optimal value of the true problem $(\mathcal{Z})$. This new formulation is reminiscent of the well-known SDP relaxation for the K-means clustering problem [24] given by

$$\begin{aligned} R_2^\star = \quad \min \quad & \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X}) - \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{Y}) \\ \text{s.t.} \quad & \boldsymbol{Y} \in \mathbb{R}_+^{N \times N} \\ & \boldsymbol{Y} \mathbf{e} = \mathbf{e} \\ & \operatorname{tr}(\boldsymbol{Y}) = K \\ & \boldsymbol{Y} \succeq \mathbf{0}. \end{aligned} \tag{$\mathcal{R}_2$}$$

Indeed, a similar argument as in the proof of Corollary 1 allows us to conclude that the problem $(\mathcal{R}_2)$ is in fact equivalent to the problem $(\mathcal{R}_1)$ once we remove the constraint $\mathbf{e}_1^\top \boldsymbol{V}_1 \mathbf{e} = 1$ from the latter problem. The symmetry breaking constraint that we have introduced however ensures that the new reformulation $(\mathcal{R}_1)$ is never weaker than the reformulation $(\mathcal{R}_2)$.

Theorem 4. *We have*

$$Z^\star \geq R_0^\star \geq R_1^\star \geq R_2^\star.$$

Proof. The first and the second inequalities hold by construction. To prove the third inequality, we consider any feasible solution $(\boldsymbol{V}_i)_{i \in [K]}$ to $(\mathcal{R}_1)$. Then, the solution $\boldsymbol{Y} = \sum_{i \in [K]} \boldsymbol{V}_i$ is feasible to $(\mathcal{R}_2)$ and gives rise to the same objective value. Thus, the claim follows. □

Obtaining any estimations of the best cluster assignment using the solution of $(\mathcal{R}_2)$ is a non-trivial endeavor. If we have *exact recovery*, *i.e.*, $Z^\star = R_2^\star$, then an optimal solution of $(\mathcal{R}_2)$ assumes the form

$$\boldsymbol{Y} = \sum_{i \in [K]} \frac{1}{\mathbf{e}^\top \boldsymbol{\pi}_i} \boldsymbol{\pi}_i \boldsymbol{\pi}_i^\top, \tag{26}$$

where $\boldsymbol{\pi}_i$ is the assigment vector for the i-th cluster. Such a solution $\boldsymbol{Y}$ allows for an easy identification of the clusters. If there is no exact recovery then a few additional steps need to be carried out. In [24], the approximate cluster assignment is obtained by solving exactly another K-means clustering problem on a lower dimensional data set (whose computational complexity scales with $\mathcal{O}(N^{(K-1)^2})$). If the solution of the SDP relaxation $(\mathcal{R}_2)$ is similar to the exact recovery solution (26), then the columns of the matrix $\boldsymbol{Y}\boldsymbol{X}$ will comprise of *denoised* data points that are close to the respective optimal cluster centroids. In [22], this strengthened signal is leveraged to identify the clusters of the original data points.

The promising result portrayed in Theorem 4 implies that any well-constructed rounding scheme that utilizes the improved formulation $(\mathcal{R}_1)$ (or $(\overline{\mathcal{R}}_1)$) will never generate inferior cluster assignments to the ones generated from the schemes that employ the formulation $(\mathcal{R}_2)$. Our new SDP relaxation further inspires us to devise an improved approximation algorithm for the K-means clustering problem. The central idea of the algorithm is to construct high quality estimates of the cluster assignment vectors $(\boldsymbol{\pi}_i)_{i \in [K]}$ using the solution $(\boldsymbol{V}_i)_{i \in [K]}$ as follows:

$$\boldsymbol{\pi}_i = \boldsymbol{V}_i \mathbf{e} \qquad \forall i \in [K].$$

To eliminate any symmetric solutions, the algorithm gradually introduces symmetry breaking constraints $\mathbf{e}_{n_i}^\top \boldsymbol{V}_i \mathbf{e} = 1$, $i \geq 2$, to $(\mathcal{R}_1)$, where the indices n_i, $i \geq 2$, are chosen judiciously. The main component of the algorithm runs in K iterations and proceeds as follows. It first solves the problem $(\mathcal{R}_1)$ and records the optimal solutions $(\boldsymbol{V}_i^\star)_{i \in [K]}$. In each of the subsequent iterations $k = 2, \ldots, K$, the algorithm identifies the best unassigned data point $\boldsymbol{x}_n$ for the k-th cluster. Here, the best data point corresponds to the index n that maximizes the quantity $\mathbf{e}_n^\top \boldsymbol{V}_k^\star \mathbf{e}$. For this index n, the algorithm then appends the constraint $\mathbf{e}_n^\top \boldsymbol{V}_k^\star \mathbf{e} = 1$ to the problem $(\mathcal{R}_1)$, which breaks any symmetry in the solution $(\boldsymbol{V}_i)_{i \geq k}$. The algorithm then solves the augmented problem and proceeds to the next iteration. At the end of the iterations, the algorithm assigns each data point $\boldsymbol{x}_n$ to the cluster k that maximizes the quantity $\mathbf{e}_n^\top \boldsymbol{V}_k^\star \mathbf{e}$. The algorithm then concludes with a single step of the Lloyd's algorithm. A summary of the overall procedure is given in Algorithm 1.

5 Numerical Results

In this section, we assess the performance of the algorithm described in Section 4. All optimization problems are solved with MOSEK v8 using the YALMIP interface [20] on a 16-core 3.4 GHz computer with 32 GB RAM.

Algorithm 1 Approximation Algorithm for K-Means Clustering

Input: Data matrix $\boldsymbol{X}$ and number of clusters K.

Initialization: Let $\boldsymbol{V}_i^\star = \mathbf{0}$ and $\mathcal{P}_i = \emptyset$ for all $i = 1, \dots, K$, and $n_k = 0$ for all $= 2, \dots, K$.

Solve the semidefinite program $(\mathcal{R}_1)$ with input $\boldsymbol{X}$ and K. Update $(\boldsymbol{V}_i^\star)_{i\in[K]}$ with the current solution.

for $k = 2, \dots, K$ **do**

  Update $n_k = \arg\max_{n\in[N]} \mathbf{e}_n^\top \boldsymbol{V}_k^\star \mathbf{e}$.

  Append the constraints $\mathbf{e}_{n_i}^\top \boldsymbol{V}_i \mathbf{e} = 1 \ \forall i = 2, \dots, k$ to the problem $(\mathcal{R}_1)$.

  Solve the resulting semidefinite program with input $\boldsymbol{X}$ and K. Update $(\boldsymbol{V}_i^\star)_{i\in[K]}$.

end for

for $n = 1, \dots, N$ **do**

  Set $k^\star = \arg\max_{k\in[K]} \mathbf{e}_n^\top \boldsymbol{V}_k^\star \mathbf{e}$ and update $\mathcal{P}_{k^\star} = \mathcal{P}_{k^\star} \cup \{n\}$.

end for

Compute the centroids $\boldsymbol{c}_k = \frac{1}{|\mathcal{P}_k|}\sum_{n\in\mathcal{P}_k} \boldsymbol{x}_n$ for all $k = 1, \dots, K$.

Reset $\mathcal{P}_k = \emptyset$ for all $k = 1, \dots, K$.

for $n = 1, \dots, N$ **do**

  Set $k^\star = \arg\min_{k\in[K]} \|\boldsymbol{x}_n - \boldsymbol{c}_k\|$ and update $\mathcal{P}_{k^\star} = \mathcal{P}_{k^\star} \cup \{n\}$.

end for

Output: Clusters $\mathcal{P}_1, \dots, \mathcal{P}_K$.

We compare the performance of the Algorithm 1 with the classical Lloyd's algorithm and the approximation algorithm[2] proposed in [22] on 100 randomly generated instances of the K-means clustering problem. While our proposed algorithm employs the improved formulation $(\mathcal{R}_1)$ to identify the clusters, the algorithm in [22] utilizes the existing SDP relaxation $(\mathcal{R}_2)$.

In each trial, we randomly draw N_1, N_2, and N_3 points, respectively, from the Gaussian distributions $\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$, $\mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$, and $\mathcal{N}(\boldsymbol{\mu}_3, \boldsymbol{\Sigma}_3)$ in $\mathbb{R}^2$. Here, the integers N_1, N_2, and N_3 are sampled uniformly at random from the set $\{1, 2, \dots, 50\}$. We fix the mean values of the distributions to $\boldsymbol{\mu}_1 = [1 \ \ 0]^\top$, $\boldsymbol{\mu}_2 = [-1 \ \ 0]^\top$, and $\boldsymbol{\mu}_3 = [0 \ \ c]^\top$, where the scalar c is drawn uniformly at random from $[2, 10]$. We set the vector of standard deviations to $\boldsymbol{\sigma} = 1/2\mathbf{e}$ and generate random correlation matrices $\boldsymbol{C}_1$, $\boldsymbol{C}_2$, and $\boldsymbol{C}_3$ using the MATLAB command '`gallery('randcorr',3)`'. The covariance matrices of the distributions are then defined as $\boldsymbol{\Sigma}_i = \operatorname{diag}(\boldsymbol{\sigma})\boldsymbol{C}_i \operatorname{diag}(\boldsymbol{\sigma})$ for all $i \in [3]$, respectively.

Table 1 reports the performance of the cluster assignments generated from the Algorithm 1 relative to the ones generated from the algorithm in [22] and the Lloyd's algorithm. We find that our proposed algorithm significantly outperforms both the other algorithms in view of the mean and the 95th percentile statistics.

[2]MATLAB implementation of the algorithm is available at `https://github.com/solevillar/kmeans_sdp`.

The percentile statistics further indicate that while the other algorithms can generate extremely poor cluster assignments, our algorithm consistently produces high quality cluster assignments and rarely loses by more than 1%.

	Statistic		
Algorithm	Mean	5th Percentile	95th Percentile
Algorithm in [22]	61.8%	-0.6%	575.1%
Lloyd's Algorithm	18.3%	-0.7%	111.3%

Table 1. Improvement of the true K-means objective value of the cluster assignment generated from the Algorithm 1 relative to the ones generated from the algorithm in [22] and the Lloyd's Algorithm.

References

[1] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization algorithms on matrix manifolds.* Princeton University Press, 2009.

[2] D. Aloise, A. Deshpande, P. Hansen, and P. Popat. NP-hardness of Euclidean sum-of-squares clustering. *Machine learning*, 75(2):245–248, 2009.

[3] M. Asteris, D. Papailiopoulos, and A. Dimakis. Nonnegative sparse PCA with provable guarantees. In *International Conference on Machine Learning*, pages 1728–1736, 2014.

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

[5] I. M. Bomze and E. de Klerk. Solving standard quadratic optimization problems via linear, semidefinite and copositive programming. *Journal of Global Optimization*, 24(2):163–185, 2002.

[6] S. Burer. On the copositive representation of binary and continuous nonconvex quadratic programs. *Mathematical Programming A*, 120(2):479–495, 2009.

[7] S. Burer. Copositive programming. In *Handbook on semidefinite, conic and polynomial optimization*, pages 201–218. Springer, 2012.

[8] S. Burer and H. Dong. Representing quadratically constrained quadratic programs as generalized copositive programs. *Operations Research Letters*, 40(3):203–206, 2012.

[9] E. de Klerk and D. V. Pasechnik. Approximation of the stability number of a graph via copositive programming. *SIAM Journal on Optimization*, 12(4):875–892, 2002.

[10] C. Ding, X. He, and H. D. Simon. On the equivalence of nonnegative matrix factorization and spectral clustering. In *International Conference on Data Mining*, pages 606–610. SIAM, 2005.

[11] C. Ding, T. Li, W. Peng, and H. Park. Orthogonal nonnegative matrix t-factorizations for clustering. In *International Conference on Knowledge Discovery and Data Mining*, pages 126–135, 2006.

[12] P. Hansen and B. Jaumard. Cluster analysis and mathematical programming. *Mathematical Programming*, 79(1-3):191–215, 1997.

[13] T. Iguchi, D. G. Mixon, J. Peterson, and S. Villar. On the tightness of an SDP relaxation of k-means. *arXiv preprint arXiv:1505.04778*, 2015.

[14] A. K. Jain. Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666, 2010.

[15] L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*, volume 344. John Wiley & Sons, 2009.

[16] D. Kuang, C. Ding, and H. Park. Symmetric nonnegative matrix factorization for graph clustering. In *International Conference on Data Mining*, pages 106–117, 2012.

[17] J. B. Lasserre. Convexity in semialgebraic geometry and polynomial optimization. *SIAM Journal on Optimization*, 19(4):1995–2014, 2009.

[18] T. Li and C. Ding. The relationships among various nonnegative matrix factorization methods for clustering. In *International Conference on Data Mining*, pages 362–371, 2006.

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

[20] J. Löfberg. YALMIP: A toolbox for modeling and optimization in MATLAB. In *IEEE International Symposium on Computer Aided Control Systems Design*, pages 284–289, 2004.

[21] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, Berkeley, Calif., 1967.

[22] D. G. Mixon, S. Villar, and R. Ward. Clustering subgaussian mixtures by semidefinite programming. *Forthcoming in Information and Inference: A Journal of the IMA*, 2017.

[23] P. A. Parrilo. *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization.* PhD thesis, California Institute of Technology, 2000.

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

[25] F. Pompili, N. Gillis, P.-A. Absil, and F. Glineur. Two algorithms for orthogonal nonnegative matrix factorization with application to clustering. *Neurocomputing*, 141:15–25, 2014.

[26] N. Rujeerapaiboon, K. Schindler, D. Kuhn, and W. Wiesemann. Size matters: Cardinality-constrained clustering and outlier detection via conic optimization. *arXiv preprint arXiv:1705.07837*, 2017.

[27] J. F. Sturm and S. Zhang. On cones of nonnegative quadratic functions. *Mathematics of Operations Research*, 28(2):246–267, 2003.