# A Convex Relaxation Approach to the Affine Subspace Clustering Problem

Francesco Silvestri[1,2], Gerhard Reinelt[1], Christoph Schnörr[2]

[1] Discrete and Combinatorial Optimization Group, Heidelberg University
[2] IPA & HCI, Heidelberg University

**Abstract.** Prototypical data clustering is known to suffer from poor initializations. Recently, a semidefinite relaxation has been proposed to overcome this issue and to enable the use of convex programming instead of ad-hoc procedures. Unfortunately, this relaxation does not extend to the more involved case where clusters are defined by parametric models, and where the computation of means has to be replaced by parametric regression. In this paper, we provide a novel convex relaxation approach to this more involved problem class that is relevant to many scenarios of unsupervised data analysis. Our approach applies, in particular, to data sets where assumptions of model recovery through sparse regularization, like the independent subspace model, do not hold. Our mathematical analysis enables to distinguish scenarios where the relaxation is tight enough and scenarios where the approach breaks down.

## 1 Introduction

Given data (measurement, pattern, observation, ...) vectors $b_i \in \mathbb{R}^d$, $i \in [n] := \{1, 2, \ldots, n\}$, the basic clustering problem amounts to jointly minimize the objective function

$$\min_{u,x} \sum_{i\in[n]} \sum_{j\in[k]} u_{ij} \|x_j - b_i\|^2 \tag{1}$$

with respect to *prototypes* $x_j \in \mathbb{R}^d$, $j \in [k]$, and *assignment variables* $u_{ij} \in \{0, 1\}$, $i \in [n]$. The well-known $k$-means algorithm shows that, if either set of variables is fixed, then solving for the other set of variables is trivial. However, the task to *jointly* solve for *both* assignment variables and prototypes is inherently combinatorial. Accordingly, there exist a broad range of heuristic algorithms ($k$-means, mean-shift, etc.) that locally solve this chicken-and-egg problem in an EM-like alternating fashion and hence strongly depend on proper initializations. To overcome this shortcoming, combinatorial optimization techniques (e.g. [10]) have been applied, but they do not scale up to large data sets. Alternatively, semidefinite convex relaxations [11] have been suggested along with extensions to ensemble clustering [14], using the same relaxation.

In this paper, we adopt the latter focus on convex relaxation but study the more involved problem

$$\min_{u,x} \sum_{i\in[n]} \sum_{j\in[k]} u_{ij} \|A_i x_j - b_i\|^2 \tag{2}$$

with *given* data $(A_i, b_i) \in \mathbb{R}^{l \times d} \times \mathbb{R}^l$, $i \in [n]$, *unknown* model parameters $x_j \in \mathbb{R}^d$, $j \in [k]$, and *unknown* assignments $u_{ij} \in \{0, 1\}$ of datum $i$ to model $j$, to be determined by minimizing the objective (2). In comparison with (1), this approach extends the representation of data by *points* (prototypes, centroids) to *affine subspaces*, which is significant for many applications.

Regarding the fitting of such "union of subspaces" models to data, significant progress has been recently made by assuming the dimensions of these spaces to be *low relative* to the ambient space [3]. This enables to establish recovery guarantees based on sparsity priors and basic convex programming techniques [6] that are more convenient and robust than alternatives like, e.g., algebraic techniques [9]. *In this paper*, however, *we do not rely on such low-rank assumptions.* A simple such problem, illustrated by Figure 1, concerns the clustering of one-dimensional linear subspaces in $\mathbb{R}^2$, which clearly violates the "independent subspaces" assumption of [6, Section 4].

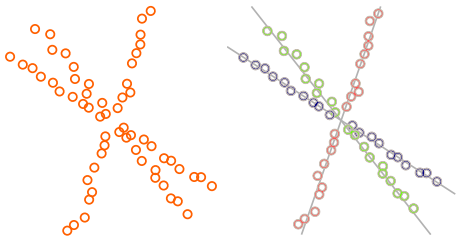

**Fig. 1.** *Left:* An unsupervised subspace clustering problem where recovery guarantees by sparse regularization fail. *Right:* Our approach jointly partitions the data and estimates the model parameter by solving a single convex optimization problem (relaxation) followed by spectral clustering.

Another and equally important line of research concerns *pairwise, graph-based clustering* [8], where locally converging methods like mean-field annealing have been developed and also extended to piecewise regression problems [13]. To reduce the susceptibility to local initializations, spectral relaxation is commonly applied [17, 4]. However, while Euclidean embeddings [1] of pairwise data provide a connection to central clustering, working out the implications for our novel mathematical approach to solve problem (2) is beyond the scope of this paper.

**Contribution, Organization.** We sketch in Section 2 the semidefinite relaxation of the basic problem (1) and elucidate why this relaxation is specific to (1) and does *not* generalize to problem (2). As a consequence, we present in Section 3 our novel mathematical approach to the relaxation of the *joint* optimization problem (2). In Section 4, some properties of the approach are derived together with limitations that are inherent to any non-tight relaxation of a combinatorial problem. The approach is illustrated by few academical examples in Section 5. We point out that working out applications is beyond the scope of our theoretical work that has been motivated by the class of unsupervised learning problems (2).

## 2 Prototypical Clustering by Convex Programming

### 2.1 Problem, Convex Relaxation

Collecting the assignment variables into a matrix $U$, the basic clustering problem (1) reads

$$\min_{u,x} \sum_{i\in[n]} \sum_{j\in[k]} u_{ij}\|x_j - b_i\|^2 \qquad \text{s. t.} \qquad Ue = e, \quad U \in \{0,1\}^{n\times k}, \tag{3}$$

where $e = (1, 1, \ldots, 1)^\top$. The derivation of a convex relaxation is based on the simple observation that, for any subset $S \subseteq [n]$ of data vectors $\{b_i\}_{i\in S}$, one has

$$\frac{1}{|S|}\sum_{i\in S} b_i \;\in\; \arg\min_x \sum_{i\in S} \|x - b_i\|^2. \tag{4}$$

Thus, given a *fixed* assignment $\{u_{ij}\}$, one can express every $x_j$ in terms of the respective $u_{ij}$ variables by setting

$$x_j(U) = \frac{\sum_{i\in[n]} u_{ij} b_i}{\sum_{i\in[n]} u_{ij}}. \tag{5}$$

Collecting all data vectors $b_i \in \mathbb{R}^d$, $i \in [n]$, as columns of a matrix $B \in \mathbb{R}^{d\times n}$, insertion of (5) into (3) yields after an elementary rearrangement

$$\min_U \langle B^\top B, I - U(U^\top U)^{-1}U^\top\rangle \quad \text{s. t.} \quad Ue = e, \quad U \in \{0,1\}^{n\times k}. \tag{6}$$

Substituting $Z = U(U^\top U)^{-1}U^\top$ gives the equivalent [11] problem

$$\min_Z \langle B^\top B, I - Z\rangle \quad \text{s. t.} \quad Ze = e,\ \langle Z, I\rangle = k,\ Z^2 = Z,\ Z \in \mathcal{S}^n \cap \mathbb{R}^{n\times n}_+ \tag{7}$$

where $\mathcal{S}^n$ denotes the linear space of symmetric $n \times n$ matrices.
Even though (7) looks much simpler than its original formulation (3), it is still intractable and nonconvex due to the constraint $Z^2 = Z$. However, this can be relaxed to $Z \in \mathcal{S}^n_+$ (semidefinite matrix cone) which yields a tractable *semidefinite program (SDP)*. In this context, $B^\top B$ plays the role of a similiarity measure which is the only data-dependent information for the algorithm.

### 2.2 Why This Approach Does Not Generalize

A key property of (7) is dealing with the inherent symmetries of (3), which is necessary for any convex relaxation. To see why this is an issue, consider the convexified set $\mathcal{U}_{n,k} = \{U \in [0,1]^{n\times k} \,|\, Ue = e\}$. For any $U \in \mathcal{U}_{n,k}$ and for any $\pi \in \mathfrak{S}_k$ ($\mathfrak{S}_k$: symmetric group on $[k]$), let $U_\pi$ be the result of permuting the columns of $U$ according to $\pi$. Then $U_\pi \in \mathcal{U}_{n,k}$ and, by convexity, $\sum_{\pi\in\mathfrak{S}_k} U_\pi = \frac{1}{k}J \in \mathcal{U}_{n,k}$ where $J = ee^\top$ is the matrix of all ones. It follows for any symmetric convex function $f$ ($f(U) = f(U_\pi)$ for all $\pi \in \mathfrak{S}_k$) that $\frac{1}{k}J$ is an optimal but

useless solution, because every point can be assigned to every cluster at the same cost.

The two key properties of (7) are that the objective is asymmetric and that the feasible set can be easily convexified. Intuitively, the symmetry variant question of (3), "which points belong to which cluster", is reduced to a weighted version of the symmetry invariant question, "which points belong to the same cluster", since $u_{rj}u_{sj}$ in $Z_{rs} = \sum_{j\in[k]}(u_{rj}u_{sj})(\sum_{i\in[n]} u_{ij})^{-1}$ denotes whether $r$ and $s$ are in cluster $j$ at the same time. This also allows to extract the clusters at the end.

Now consider generalizations of problem (1) of the form

$$\min_{u,x} \sum_{i\in[n]} \sum_{j\in[k]} u_{ij}\|f(x_j, A_i) - b_i\|^2 \quad \text{s. t.} \quad Ue = e, \quad U \in \{0,1\}^{n\times k} \tag{8}$$

for some differentiable function $f$ and data $(A_i, b_i)$, $i \in [n]$. If one wants to generalize the approach of Section 2.1 accordingly, then the prototypes have to be eliminated and the objective has to be reduced to an asymmetric convex function in the remaining variables. Assume cluster $j$ is indexed by $S \subseteq [n]$, that is the assignment variables are fixed and the constraints obsolete. Then, taking derivatives gives the optimality condition

$$0 = \sum_{i\in S} \big\langle \nabla_x f(x, A_i)\big|_{x=x_j}, f(x_j, A_i) - b_i \big\rangle. \tag{9}$$

Depending on $f$, (9) is arbitrarily hard to solve for $x_j$ in closed form. The simplest generalization takes the form (2), that is $f(x_j, A_i) = A_i x_j$. Then (9) becomes

$$x_j(U) = \Big( \sum_{i\in[n]} u_{ij} A_i^\top A_i \Big)^\dagger \Big( \sum_{i\in[n]} u_{ij} A_i^\top b_i \Big). \tag{10}$$

Unfortunately, taking the pseudo-inverse $(\dots)^\dagger$ of a linearly parametrized matrix is highly nonlinear. In particular, even if we could assume that the matrices $\sum_{i\in S} A_i^\top A_i$ admit an ordinary matrix inverse, then $x_j(U)$ in (10) would be a multivariate rational function in $U$ whose coefficients strongly depend on the specific given data. Without further assumptions, there is neither an easy way to see the range of possible values for the coefficients of $U$ after substituting $x_j$ by (10) nor an easy way to estimate the approximation quality of the corresponding convex hull. These facts motivate our approach presented in the subsequent section.

## 3 Joint Approach to Clustering and Regression

### 3.1 Problem, Problem Reformulation

In this section, we consider problem (2) in the form

$$\min_{u,x} \sum_{i\in[n]} \sum_{j\in[k]} u_{ij}\|A_i x_j - b_i\|^2 \tag{11a}$$

$$\text{s.t. } Ue = e, \quad U \in \{0,1\}^{n\times k}, \quad \{x_j\}_{j\in[k]} \subseteq P \tag{11b}$$

where $P \subseteq \mathbb{R}^d$ is a polytope, $\{A_i\} \subseteq \mathbb{R}^{l\times d}$ and $\{b_i\} \subseteq \mathbb{R}^l$. This is equal to (2) if we know a polytope $P$ containing the optimal solution. We will assume this for now, showing examples where we can construct $P$ in closed form in Section 5.

**Problem Reformulation.** Since $P$ is a polytope, $P = \operatorname{conv}(\{v_s\}_{s\in[m]})$ for the columns $v_s$ of some matrix $V \in \mathbb{R}^{d\times m}$. By Caratheodory's theorem [12, Thm. 2.29], we can thus assume that there is a $\lambda^j \in \mathbb{R}^m_+$ where $\langle \lambda^j, e\rangle = 1$ and $|\operatorname{supp}(\lambda^j)| \leq d+1$ such that $x_j = V\lambda^j$.
Using this substitution and applying that $1 = \langle \lambda^j, e\rangle$, one easily checks that

$$\|A_i x_j - b_i\|^2 = \sum_{r,s\in[m]} \lambda^j_r \lambda^j_s (v_r^\top A_i^\top A_i v_s - (b_i^\top A_i)(v_r + v_s) + \|b_i\|^2). \tag{12}$$

Setting $W_i$, $i \in [n]$, with $(W_i)_{rs} := (v_r^\top A_i^\top A_i v_s - (b_i^\top A_i)(v_r + v_s) + \|b_i\|^2)_{rs}$ and $\Lambda^j := \lambda_j \lambda_j^\top$, yields $\|A_i x_j - b_i\|^2 = \langle \Lambda^j, W_i\rangle := \operatorname{tr}(\Lambda^j W_i)$ and the reformulation

$$\min_{u,\Lambda} \sum_{i\in[n]} \sum_{j\in[k]} u_{ij} \langle \Lambda^j, W_i\rangle \tag{13a}$$

$$\text{s.t. } Ue = e, \quad U \in \{0,1\}^{n\times k},$$

$$\langle \lambda^j, e\rangle = 1, \quad \lambda^j \geq 0, \quad \|\lambda^j\|_0 \leq d+1, \quad \Lambda^j = \lambda^j {\lambda^j}^\top. \tag{13b}$$

The constraints (13b) can be equivalently expressed in terms of $\Lambda^j$ by demanding

$$\langle \Lambda^j, J\rangle = 1, \quad \operatorname{rank}(\Lambda^j) = 1, \quad \Lambda^j \in \mathcal{CP}^m, \quad \|\operatorname{diag}(\Lambda^j)\|_0 \leq d+1 \tag{14}$$

where $\mathcal{CP}^m := \{M \in \mathcal{S}^m : M = \sum \mu_i \mu_i^\top, \mu_i \in \mathbb{R}^m_+\}$ is the cone of completely positive matrices [2].

### 3.2 Convex Relaxation

In order to get a convex relaxation we have to convexify both the objective and the feasible set. We even go one step further and linearize the objective.

**Linearizing the Objective.** Setting $\Lambda_i(U) := \sum_{j\in[k]} u_{ij}\Lambda^j$, we get

$$\sum_{i\in[n]} \langle \sum_{j\in[k]} u_{ij}\Lambda^j, W_i\rangle = \sum_{i\in[n]} \langle \Lambda_i(U), W_i\rangle, \tag{15}$$

where the variables $U$ model $\Lambda_i(U) \in \{\Lambda^j\}_{j\in[k]}$, which is invariant under permutations of $(\Lambda^1, \ldots, \Lambda^k)$. This implies that relaxing the condition $\Lambda_i(U) \in \{\Lambda^j\}_{j\in[k]}$ without introducing symmetry is a good first step to get a tractable relaxation with a linear objective.
To proceed, we derive some properties of $\{\Lambda^j\}_{j\in[k]}$. Consider the sets

$$\begin{aligned} \mathcal{N}^m_{\nu,d} :&= \{\Lambda \in \mathcal{CP}^m \colon \langle \Lambda, J\rangle = \nu,\ \operatorname{rank}(\Lambda) \in [\nu],\ \|\operatorname{diag}(\Lambda)\|_0 \leq \nu(d+1)\} \\ &= \nu \cdot \mathcal{N}^m_{1,d} \end{aligned} \tag{16}$$

where $\nu \cdot \mathcal{N}^m_{1,d}$ denotes the Minkowski-sum of $\nu$ copies of $\mathcal{N}^m_{1,d}$. In particular, we have

$$\sum_{j \in S} \Lambda^j \in \mathcal{N}^m_{|S|,d} \quad \text{for all } S \subseteq [k]. \tag{17}$$

It follows that for every feasible, integral assignment $U$, we have

$$\Lambda_i(U) \in \mathcal{N}^m_{1,d}, \quad \Lambda^* := \sum_{j \in [k]} \Lambda^j \in \mathcal{N}^m_{k,d} \quad \text{and} \quad \Lambda^* - \Lambda_i(U) \in \mathcal{N}^m_{k-1,d}. \tag{18}$$

Thus, replacing $\Lambda_i(U)$ by a variable $\Lambda_i$ defines an asymmetric linear objective function for the relaxation

$$\min_{\Lambda} \sum_{i \in [n]} \langle \Lambda_i, W_i \rangle \quad \text{s.t.} \quad \Lambda^* \in \mathcal{N}^m_{k,d}, \quad \Lambda_i \in \mathcal{N}^m_{1,d}, \quad \Lambda^* - \Lambda_i \in \mathcal{N}^m_{k-1,d}. \tag{19}$$

The only relaxation made so far concerns condition $\Lambda^* - \Lambda_i \in \mathcal{N}^m_{k-1,d}$ that cannot *strictly* enforce the set $\{\Lambda_i\}_{i \in [n]}$ to only have $k$ distinct members. While some problem structure is lost, this is necessary to remove the symmetry.

**Relaxing the feasible region.** Optimizing over the set $\mathcal{N}^m_{\nu,d}$ is intractable. The rank-constraint as well as the bounded support make the problem non-convex and very hard in practice. Furthermore, even though $\mathcal{CP}^m$ is a convex cone, separation over $\mathcal{CP}^m$ is NP-hard [5], so this is intractable as well.
Since we are interested in a tractable convex relaxation, we apply standard relaxations for these conditions. To this end, define the sets

$$\mathcal{M}^m_{\nu,d} := \{\Lambda \in \mathcal{S}^m_+ \cap \mathbb{R}^{m \times m}_+ \colon \langle \Lambda, J \rangle = \nu, \operatorname{tr}(\Lambda) \geq \frac{\nu}{d+1}\} = \nu \cdot \mathcal{M}^m_{1,d}, \tag{20a}$$

$$\mathcal{K} := \mathcal{S}^m_+ \cap \mathbb{R}^{m \times m}_+. \tag{20b}$$

**Theorem 1.** *$\mathcal{M}^m_{\nu,d}$ is convex, tractable and $\mathcal{N}^m_{\nu,d} \subseteq \mathcal{M}^m_{\nu,d}$.*

*Proof.* $\mathcal{CP}^m \subseteq \mathcal{K}$ follows from the definition. Furthermore, $\mathcal{N}^m_{\nu,d} \subseteq \mathcal{M}^m_{\nu,d}$ is implied by $\mathcal{N}^m_{1,d} \subseteq \mathcal{M}^m_{1,d}$, so consider $\nu = 1$. For $\Lambda \in \mathcal{N}^m_{1,d}$, by definition, there exists $\lambda$ such that $\Lambda = \lambda\lambda^\top$, $\lambda \geq 0$, $\langle \lambda, e \rangle = 1$ and $\|\lambda\|_0 \leq d+1$. We have $\operatorname{tr}(\Lambda) = \|\lambda\|_2^2$ and one can verify that under these constraints a minimizer of this term is given by any vector $\lambda^*$ where $\|\lambda^*\|_0 = d+1$ and $\lambda^*_i = \frac{1}{d+1}$ for all $i \in \operatorname{supp}(\lambda^*)$. This gives the desired lowerbound on $\operatorname{tr}(\Lambda)$. □

As a direct corollary, we get the *tractable, convex relaxation* of our problem

$$\min_{\Lambda} \sum_{i \in [n]} \langle \Lambda_i, W_i \rangle \quad \text{s.t.} \quad \Lambda^* \in \mathcal{M}^m_{k,d},\ \Lambda_i \in \mathcal{M}^m_{1,d},\ \Lambda^* - \Lambda_i \in \mathcal{M}^m_{k-1,d}. \tag{21}$$

Again, this relaxation loses some structure of the problem but is necessary to achieve tractability.

### 3.3 Extension to Disjunctive Programming

In (11), we required the prototypical model parameters to be contained in a polytope: $\{x_j\}_{j\in[k]} \subseteq P$. We can generalize $P$ to a finite union of (not necessarily disjoint) polytopes $\mathcal{P} = \bigcup_{t\in T} P_t$ with some additional work. Let $V_t$ be the matrix which has the vertices of $P_t$ as columns. Then $x \in \mathcal{P}$ is equivalent to $x = \sum_{t\in T} V_t\lambda_t$ for a vector $\lambda^\top = (\lambda_1^\top, \dots, \lambda_{|T|}^\top)$ such that

$$\langle \lambda, e\rangle = 1, \quad \lambda \geq 0, \quad \|\lambda\|_0 \leq d+1, \quad \{(\lambda_r = 0) \vee (\lambda_s = 0)\}_{r,s\in T} \tag{22}$$

where $\vee$ denotes the logical *or*. Adding $\{(\lambda_r = 0) \vee (\lambda_s = 0)\}_{r,s\in T}$ to (13b) then results in a *disjunctive program* [7].

Now observe that $(\lambda_r = 0) \vee (\lambda_s = 0)$ implies $\lambda_r\lambda_s^\top = 0$ for any $r, s \in T$, so the matrix $\Lambda = \lambda\lambda^\top = (\lambda_r\lambda_s^\top)_{r,s\in T}$ is block diagonal. Since $\Lambda \geq 0$, this can be encoded by a 0/1-matrix $\Omega$ as a single linear constraint $\langle \Lambda, \Omega\rangle = 0$, where $J - \Omega$ shares the block structure of $\Lambda$.
Using the rank condition one can show that adding $\{(\lambda_r = 0) \vee (\lambda_s = 0)\}_{r,s\in T}$ to (13b) is equivalent to adding $\langle \Lambda^j, \Omega\rangle = 0$ to (14). Following Section 3.2 we can relax this constraint for (21) to $\langle \Lambda^*, \Omega\rangle = 0$, which implies $\langle \Lambda_i, \Omega\rangle = 0$ for all $i \in [n]$. Hence, we showed

**Proposition 1.** *Let $\Omega \in \{0,1\}^{m\times m}$ be symmetric and $\operatorname{tr}(\Omega) = 0$. Then adding $\langle \Lambda^*, \Omega\rangle = 0$ to (21) entails that the solution $x_j$ of (11) can be written as a convex combination of $\{v_i\}_{i\in S_j}$ where $[m] \supseteq S_j \not\supseteq \{r, s\}$ for all $\omega_{rs} = 1$. In particular, if $J - \Omega$ is block diagonal, then $\langle \Lambda^*, \Omega\rangle = 0$ implies that $P = \bigcup_{t\in T} P_t$, where each $P_t$ is the convex hull of columns $V_t$ indexed by a diagonal block.*

Note that, while the relaxation in $\Lambda$ is convex, the recovery of $\lambda$ from $\Lambda$ will in general not preserve convexity. Depending on how we recover $\lambda$, the relaxation does not necessarily model a convex space in the $x$ variables, which makes this approach viable. Now observe, however, that for the objective function, convex combinations of rank-1 matrices are in general "bad" since, by linearity and for any convex combination $\Lambda = \sum_{i\in S} \mu_i\lambda_i\lambda_i^\top$, we have $\langle W, \Lambda\rangle = \sum_{i\in S} \mu_i\langle W, \lambda_i\lambda_i^\top\rangle \geq \min_{i\in S}\langle W, \lambda_i\lambda_i^\top\rangle$. Setting $\omega_{rs} = 1$ cuts off rank-1 matrices $\Lambda$ corresponding to $\lambda$ with $\lambda_r, \lambda_s > 0$. As a consequence, optimization will favor rank-1 matrices with either $\lambda_r = 0$ or $\lambda_s = 0$ instead of approximating the cut off matrix, which shows that Prop. 1 extends problem (11) and its relaxation in a reasonable way.

### 3.4 Algorithm

While the computation of (21) is straight forward using any SDP-solver, rounding the solution afterwards requires some care. The easiest way is to use spectral clustering. To this end, define a similarity matrix $H$ by setting $H_{rs} = 1 - \langle \Lambda_r, \Lambda_s\rangle/(\|\Lambda_r\|_2 \cdot \|\Lambda_s\|_2)$, which yields a value in $[0, 1]$ corresponding to the angle between $\Lambda_r$ and $\Lambda_s$ in $\mathbb{R}^{m\times m}$.

**Algorithm 1.1**: $k$-Cluster Relaxation

**Data**: $\{(A_i, b_i)\}_{i\in[n]} \subseteq \mathbb{R}^{l\times d} \times \mathbb{R}^l, V \in \mathbb{R}^{d\times m}, k \in \mathbb{N}, \Omega \in \{0,1\}^{m\times m}$
**Result**: assignments $U$ and centroids $\{x_j\}_{j\in[k]}$
compute $W_i \leftarrow (v_r^\top A_i^\top A_i v_s - (b_i^\top A_i)(v_r + v_s) + \|b_i\|^2)_{rs}$ for $i \in [n]$;
solve (21) subject to $\langle \Lambda^*, \Omega\rangle = 0$ for $\{\Lambda_i\}_{i\in[n]}$;
compute similiarity matrix $H \leftarrow (1 - \langle \Lambda_r, \Lambda_s\rangle/(\|\Lambda_r\|_2 \cdot \|\Lambda_s\|_2))_{rs}$;
compute the assignment $U$ by spectral clustering using $H$;
compute centroids $\{x_j\}_{j\in[k]}$ using $U$;
**return** $(U, \{x_j\}_{j\in[k]})$;

## 4 Analysis

Inspecting the relaxed problem formulation (21) reveals the following: The objective function is separable in terms of the variables $\Lambda_i$, and the right-most constraint that has to be satisfied *simultaneously* for all $\Lambda_i$, $i \in [n]$, fuses this local information. In this section we derive conditions that characterize when this latter condition is sufficiently weak so that the relaxation must fail. Conversely, the more these conditions are not satisfied, the more likely the relaxation will return a useful result. Our theoretical findings will be illustrated in Section 5.

Specifically, we derive values of $(k, m, d)$ so that we can choose $\Lambda^* \in \mathcal{M}^m_{k,d}$ such that $\Lambda^* - \Lambda_i \in \mathcal{M}^m_{k-1,d}$ will be satisfied *for all* choices of $\Lambda_i \in \mathcal{M}^m_{1,d}$. Our corresponding main result is stated below as Theorem 2.
Condition $\Lambda^* - \Lambda_i \in \mathcal{M}^m_{k-1,d}$ is equivalent to

$$\mathrm{tr}(\Lambda^*) - \mathrm{tr}(\Lambda_i) \geq \frac{k-1}{d+1}, \qquad \Lambda^* - \Lambda_i \in \mathcal{K}. \tag{23}$$

Note that since $\mathrm{tr}(\Lambda_i) \leq \langle \Lambda_i, J\rangle = 1$ is sharp, we infer that $\mathrm{tr}(\Lambda^*) \geq \frac{d+k}{d+1}$ is necessary for the first condition to hold.
As for the second condition, let $A \leq_{\mathcal{K}} B$ denote the inclusion $B - A \in \mathcal{K}$. Then we need an upper bound of $\mathcal{M}^m_{1,d}$ with respect to the partial order $\leq_{\mathcal{K}}$, given by the following Lemma.

**Lemma 1 ($\leq_{\mathcal{K}}$-Upper Bound of $\mathcal{M}^m_{1,d}$).** *$I + \frac{1}{4}J$ is a $\leq_{\mathcal{K}}$-upper bound for $\mathcal{M}^m_{1,d}$.*

Our main result is

**Theorem 2 (Decoupling Condition).** *There is a matrix $\Lambda^* \in \mathcal{M}^m_{k,d}$ such that $(\Lambda^* - \Lambda) \in \mathcal{M}^m_{k-1,d}$ for all $\Lambda \in \mathcal{M}^m_{1,d}$ if there are $\alpha, \beta \in \mathbb{R}$ such that $k = \beta \cdot m^2 + \alpha \cdot m$, $\beta \geq \frac{1}{4}$ and*

$$\alpha \geq \max\{1, \frac{\beta}{d}(m - (d+1)) + \frac{1}{m}\}. \tag{24}$$

*In particular, for fixed $m, d$ there is a minimal value $k^*(m, d) \in \mathbb{N}$ that satisfies these conditions, and the conditions can be satisfied for all $k \geq k^*(m, d)$.*

*Proof.* We fix $m, d$ and derive conditions on $k$. By symmetry of $\mathcal{M}^m_{k,d}$ we can assume that $\Lambda^* = \alpha I + \beta J$ where $\beta \geq \frac{1}{4}$ and $\alpha \geq 1$ by Lemma 1. It follows that $k = \langle \Lambda^*, J \rangle = \beta \cdot m^2 + \alpha \cdot m$ and $\operatorname{tr}(\Lambda^*) = (\alpha + \beta)m$.
Together with $\operatorname{tr}(\Lambda^*) \geq \frac{d+k}{d+1}$ from (23), we have $(\alpha+\beta)m = \operatorname{tr}(\Lambda^*) \geq \frac{d+k}{d+1} > \frac{k}{d+1}$, where we can substitute $k = \beta \cdot m^2 + \alpha \cdot m$ and rewrite it as $\alpha \geq \frac{\beta}{d}(m-(d+1)) + \frac{1}{m}$. Since $m, d$ is fixed, the inequalities bound $\alpha, \beta$ and thus $k$ from below. Therefore a minimal value $k^*(m, d) \in \mathbb{N}$ that satisfies these conditions exists. □

## 5 Experiments

All examples have been carried out in Matlab using the SDPT3 [15, 16] package.
**Euclidean Clustering.** By choosing $A_i = I$ we recover (3), where (4) tells us to use $P \supseteq \operatorname{conv}(\{b_i\}_{i \in [n]})$. Using any simplex containing all the points is a coarsest approximation, but yields in general bad results.
Fig. 2 is tied to Thm. 2 - while $k$ is fixed, $k^*(m, d)$ and the quality increase from top to bottom as a consequence of additional polytopes separating the local solutions: When $\Lambda^r$, $\Lambda^s$ are optimal centroids, then (21) has $\langle J, \Lambda^r \wedge \Lambda^s \rangle$ excess weight to shift around in $\Lambda^*$, where $\wedge$ denotes the componentwise minimum. Refining $P$, $\Lambda^r \wedge \Lambda^s$ decreases, thus improving the quality. Given that the optimal solution is already covered, adding disjoint polytopes does not negatively impact the quality of the output, as can be seen in the bottom row of Fig. 2.
**Hyperplane Clustering.** By choosing $b_i = 0$ for all $i \in [n]$ and choosing $A_i = a_i$ as row vectors, problem (2) translates into finding normal vectors $x_j$ of $k$ hyperplanes such that every data point $a_i$ lies on exactly one hyperplane. To exclude the degenerated solution 0 we need an appropriate $P$ for (11).
Without loss of generality we can assume that the $x_j$ are unit vectors belonging to the "upper" half-sphere $S^{d-1} \cap H$, where $H = \{x \in \mathbb{R}^d \mid x_1 \geq 0\}$. The coarsest polytope approximation $P$ is then given by the union of the facets of $C_d$ in $H$, where $C_d$ is the cross polytope $C_d = \operatorname{conv}\{\pm e_l \mid l \in [d]\}$ of dimension $d$. This yields $V = \left(\begin{smallmatrix} 0 & 1 & 0 \\ -I & 0 & I \end{smallmatrix}\right)$ and $\omega_{(e_l, -e_l)} = 1$ for all $1 \neq l \in [d]$.
Ideally, $P$ corresponds to a disjoint union of polytopes each including one $\Lambda^j$. Figure 3 shows that one may need to use separate copies of the same vertices.

## 6 Conclusion

We introduced a novel mathematical model to deal with the affine subspace clustering problem. Our analysis shows why it works reasonably well. Experiment show that it is attractive to use the algorithm with an oversegmentation of the set of feasible solutions, with the focus on separating local solutions. This cannot be achieved using sparsity regularization. Prior knowledge can be used to speed up the algorithm, but is not necessary. Automatically balancing this trade off based on the data in an efficient way is a subject for future work.

**Acknowledgement:** Authors gratefully acknowledge support by the DFG, grant GRK 1653.

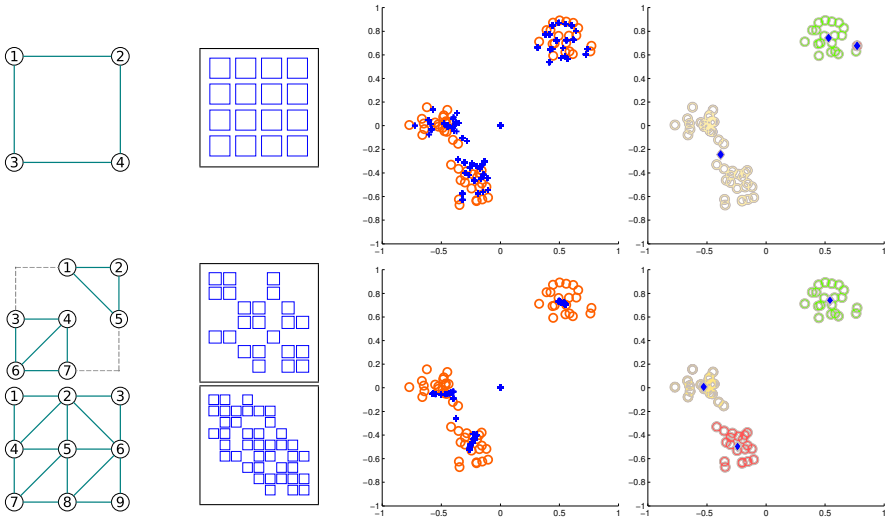

**Fig. 2.** Euclidean Clustering on data spread around three points in 2d, k=3. *Left:* Partial cover of $[-1,1]^2$ given by $V$ and corresponding block structure of $\Lambda^*$ given by $\Omega$. *Middle:* Orange data points and blue centroids extracted from $\Lambda_i$. *Right:* Clustered data points and blue centroids given by our algorithm. *Top:* Naive cover by a single square where $\Omega = 0$. *Bottom:* Optimal choice of $P$ and oversegmentation yield the same result.

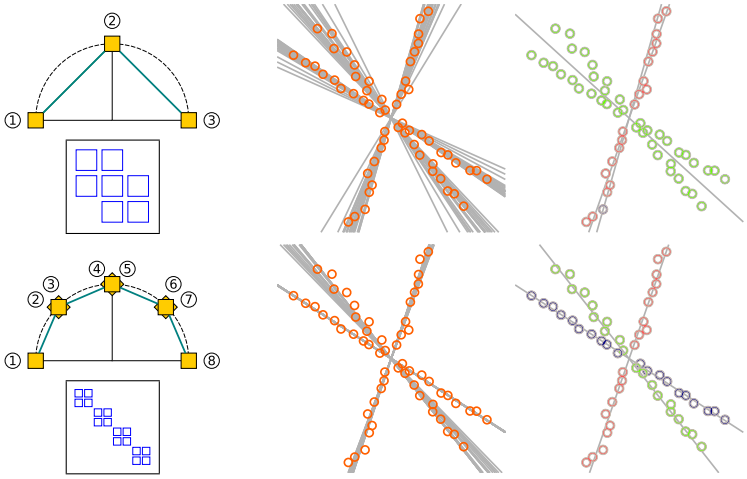

**Fig. 3.** Hyperplane Clustering on three Lines in 2d, k=3. *Left:* Polytope approximation of $S^1 \cap H$ given by $V$ and corresponding block structure of $\Lambda^*$ given by $\Omega$. *Middle:* Orange data points and grey centroids extracted from $\Lambda_i$. *Right:* Clustered data points and grey centroids given by our algorithm. *Top:* Coarsest approximation given by the facets of $C_2$ in $H$. *Bottom:* Oversegmentation where separate copies of the same vertices needed to be used to get the proper result.

## References


1. Belkin, M., Niyogi, P.: Laplacian Eigenmaps for dimensionality reduction and data representation. Neural Computation 15(6), 1373–1396 (2003)
2. Berman, A., Shaked-Monderer, N.: Completely Positive Matrices. World Sci. Publ. (2003)
3. Carin, L., Baraniuk, R., Cevher, V., Dunson, V., Jordan, M., Sapiro, G., Wakin, M.: Learning Low-Dimensional Signal Models. IEEE Signal Proc. Mag. 28(2), 39–51 (2011)
4. Chen, G., Lerman, G.: Foundations of a Multi-way Spectral Clustering Framework for Hybrid Linear Modeling. Found. Comp. Math. 9, 517–558 (2009)
5. Dickinson, P., Gijben, L.: On the computational complexity of membership problems for the completely positive cone and its dual. Computational Optimization and Applications 57(2), 403–415 (2014)
6. Elhamifar, E., Vidal, R.: Sparse Subspace Clustering: Algorithm, Theory, and Applications. IEEE Trans. Patt. Anal. Mach. Intell. 35(11), 2765–2781 (2013)
7. Grossmann, I.E., Lee, S.: Generalized convex disjunctive programming: Nonlinear convex hull relaxation. Computational Optimization and Applications 26(1), 83–100 (2003)
8. Hofman, T., Buhmann, J.: Pairwise Data Clustering by Deterministic Annealing. IEEE Trans. Patt. Anal. Mach. Intell. 19(1), 1–14 (1997)
9. Ma, Y., Yang, A., Derksen, H., Fossum, R.: Estimation of Subspace Arrangements with Applications in Modeling and Segmenting Mixed Data. SIAM Review 50(3), 413–458 (2008)
10. du Merle, O., Hansen, P., Jaumard, B., Mladenović, N.: An interior points algorithm for minimum sum-of-squares clustering. SIAM J. Sci. Comput. 21(4), 1485–1505 (2000)
11. Peng, J., Wei, Y.: Approximating $K$-means-type Clustering via Semidefinite Programming. SIAM J. Optimization 18(1), 186–205 (2007)
12. Rockafellar, R., Wets, R.J.B.: Variational Analysis. Springer, 2nd edn. (2009)
13. Rose, K.: Deterministic Annealing for Clustering, Compression, Classification, Regression, and Related Optimization Problems. Proc. IEEE 86(11), 2210–2239 (1998)
14. Singh, V., Mukherjee, L., Peng, J., Xu, J.: Ensemble clustering using semidefinite programming with applications. Mach. Learning 79(1-2), 177–200 (2010)
15. Toh, K.C., Todd, M.J., Tütüncü, R.H.: SDPT3 — a MATLAB software package for semidefinite programming (Dec 1996)
16. Tütüncü, R.H., Toh, K.C., Todd, M.J.: Solving semidefinite-quadratic-linear programs using sdpt3. Math. Program. 95(2), 189–217 (2003)
17. Xing, E., Jordan, M.: On Semidefinite Relaxation for Normalized k-cut and Connections to Spectral Clustering. Technical Report UCB/CSD-03-1265, EECS Department, University of California, Berkeley (June 2003)