



Efficient, certifiably optimal clustering with applications to latent variable graphical models

Carson Eisenach¹ · Han Liu²

Received: 11 December 2017 / Accepted: 7 February 2019 / Published online: 16 February 2019
© Springer-Verlag GmbH Germany, part of Springer Nature and Mathematical Optimization Society 2019

Abstract

Motivated by the task of clustering either d variables or d points into K groups, we investigate efficient algorithms to solve the Peng–Wei (P–W) K -means semi-definite programming (SDP) relaxation. The P–W SDP has been shown in the literature to have good statistical properties in a variety of settings, but remains intractable to solve in practice. To this end we propose FORCE, a new algorithm to solve this SDP relaxation. Compared to off-the-shelf interior point solvers, our method reduces the computational complexity of solving the SDP from $\tilde{O}(d^7 \log \epsilon^{-1})$ to $\tilde{O}(d^6 K^{-2} \epsilon^{-1})$ arithmetic operations for an ϵ -optimal solution. Our method combines a primal first-order method with a dual optimality certificate search, which when successful, allows for early termination of the primal method. We show for certain *variable clustering problems* that, with high probability, FORCE is guaranteed to find the optimal solution to the SDP relaxation and provide a certificate of exact optimality. As verified by our numerical experiments, this allows FORCE to solve the P–W SDP with dimensions in the hundreds in only tens of seconds. For a variation of the P–W SDP where K is not known a priori a slight modification of FORCE reduces the computational complexity of solving this problem as well: from $\tilde{O}(d^7 \log \epsilon^{-1})$ using a standard SDP solver to $\tilde{O}(d^4 \epsilon^{-1})$.

Mathematics Subject Classification 90C22 · 90C35 · 90C90 · 62H30

✉ Carson Eisenach
eisenach@princeton.edu

Han Liu
hanliu@northwestern.edu

¹ Department of Operations Research and Financial Engineering, Princeton University, Princeton, NJ 08544, USA

² Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL 60208, USA

1 Introduction

Clustering a set of objects optimally according to some similarity measure is a central task of statistics and machine learning. These problems arise everywhere from the analysis of medical imaging data to search result groupings on Google. Such tasks can be broadly categorized as either: *data clustering*, where we partition of d points in \mathbb{R}^p into K clusters, or *variable clustering*, where we consider n samples of a random variable $X \in \mathbb{R}^d$ and group the variables into K groups of size at least m . In many actual use cases the purpose of clustering is to recover some underlying *ground truth*, a partition $G^* = \{G_1^*, \dots, G_K^*\}$; the optimization objective and similarity measure are chosen such that the optimal partitioning corresponds to the ground-truth.

For data clustering, one classical formulation is K -means:

$$\operatorname{argmin}_G \sum_{s=1}^K \sum_{i \in G_s} \|x_i - \mu_s\|_2^2, \text{ subject to } \mu_s = \frac{1}{|G_s|} \sum_{i \in G_s} x_i, \quad (1)$$

This formulation, roughly speaking, can also be applied to variable clustering by treating $\operatorname{Cov}(X)$ as a measure of “distances” between d points [1]. Because (1), and combinatorial optimization in general, is NP-hard [2,3], fast algorithms that have been proposed to solve clustering problems are not guaranteed to produce an optimal solution to the original problem [4–8].

This becomes a major issue in certain scenarios, like post-selection inference, where first a statistical model is selected, e.g. through variable clustering, and then an inferential procedure is applied. Nearly optimal clusterings are insufficient for this purpose because incorrect model selection will invalidate the results of subsequent inferences; for such applications recovery of the optimal clustering is required. Applications where variable clustering and statistical inference questions arise include the analysis of stock pricing, fMRI, and gene expression data.

One particularly interesting class of algorithms leverage a *convex relaxation* to find an approximate solution, followed by a rounding step [9]. Though this may not always give an optimal solution to the original problem, significant progress has been made on understanding when such relaxations are *tight*—that the optimal solution to the relaxed and original problems coincide [1,8,10,11]. Motivated by recent developments in cluster based graphical models, in particular the G -Latent model (see Sect. 2.1) where each cluster of variables corresponds to a latent generator [1,12,13], we study efficient algorithms for exact cluster recovery.

Bunea et al. [1] show that the Peng–Wei (P–W) SDP relaxation (see Sect. 2.1) of (1) is tight with high probability¹ for G -Latent models and introduce a procedure to recover G^* based on solving this SDP. Similarly recent work [10,14] has studied when convex relaxations are tight in the data clustering setting. In this setting, it is again the P–W SDP which has the strongest statistical guarantees [10,11,15,16].

Despite the attractive theoretical properties of the P–W SDP for a variety of clustering problems, efficiently solving it in practice remains a significant challenge: to

¹ If an event occurs with probability $q(d)$ for dimension d , it is said to occur with high probability if $q(d) \geq 1 - C/d$ for all d sufficiently large.

find a δ -optimal solution, standard SDP solvers have worst-case $\tilde{O}(d^7 \log \delta^{-1})$ running time due to a large number of constraints.^{2,3} In this paper we introduce FORCE (First-ORder CErtifiably Optimal Clustering), an algorithm to solve the P–W SDP. The difficulty in solving NP-hard problems, such as K -means, derives from the integer structure of their solutions. The underlying insight is that for clustering problems, when we expect the convex relaxation to be tight, the integer structure of the optimal solution can actually be leveraged to *help solve* the clustering problem. The FORCE algorithm consists of two components: a first-order method to solve the P–W SDP and a dual solution construction used to certify the optimality of a primal solution. The idea is that if we have an algorithm to quickly construct a dual solution at G^* and an interior point method to solve an SDP relaxation \mathcal{P} , then while solving \mathcal{P} we can periodically “round” the current iterate and search for a matching dual solution. If the primal and dual objective values match, the algorithm can terminate early.

We summarize our main contributions below:

1. **FORCE primal step and convergence analysis** A first-order algorithm for the P–W SDP based on a variant of Renegar’s Smoothed Scheme (RSS) [17]. By converting the SDP to an eigenvalue maximization problem, we obtain a substantially improved computational complexity because we can reduce the *effective dimension* of the problem from $\mathcal{O}(d^2)$ to $\mathcal{O}(d)$. This allows us to reduce the number of arithmetic operations required to approximately solve the P–W SDP from $\tilde{O}(d^7 \log \delta^{-1})$ to $\tilde{O}(d^6 K^{-2} \epsilon^{-1})$.
2. **Dual certificate and probabilistic guarantees for variable clustering** We introduce a novel dual certificate for the P–W SDP that is tailored to variable clustering and easy to compute. We show that for clustering in G -Latent models, this certificate is guaranteed to exist with high probability (w.h.p.) at a nearly the minimax optimal cluster separation rate required for recovery of G^* .
3. **Extensions to unknown K** We extend FORCE to a P–W SDP variant recently considered for variable clustering when K is not known [1]. Theoretical guarantees translate almost 1-to-1 from the case when K is fixed, except now the FORCE primal step requires $\tilde{O}(d^4 \epsilon^{-1})$ arithmetic operations to obtain an ϵ -optimal solution.

Remark 1 We make no claims as to the statistical properties of the dual certificate for other generative models for the clustering data—e.g. for the stochastic block model or stochastic ball model. In general, the design of an appropriate dual certificate is closely linked to the data generating distribution. In any case the primal step is still applicable—to the best of our knowledge our proposed method is the most efficient algorithm to date for solving the P–W SDP—and in practice the dual certificate may be useful even if it is not guaranteed to exist w.h.p., but this is beyond the scope of our work.

Remark 2 Our theoretical analysis of the statistical properties of the proposed dual certificate also provides an alternative proof of the tightness of the P–W SDP for

² Note the switch to δ , with which we denote an additive error; ϵ , used to quantify the error of FORCE, more properly corresponds to a type of relative additive error (Sect. 3).

³ The notation \tilde{O} is used to suppress poly-log factors of d .

variable clustering in G -Latent models, at nearly the same cluster separation rate as in the literature [1]. This proof differs from Bunea et al. [1] in that it is more constructive in nature since it analyzes the properties of an explicit dual solution construction. It also shows that instances are perfectly recoverable using and can be proven optimal for the P–W SDP at nearly the same cluster separation rate.

Notation Throughout, d denotes dimension and K , the number of clusters. Clusterings, either of data points or variables, are denoted by $G = \{G_1, \dots, G_K\}$, where G_i is a single cluster or variable group. Hats, i.e. \hat{G} , always indicate quantities estimated from data and stars, i.e. G^* , always denote ground truths. For a $n \times n$ matrix \mathbf{M} , $\|\mathbf{M}\|_2$ denotes the largest eigenvalue of \mathbf{M} and $\|\mathbf{M}\|_\infty$ is the matrix ℓ_∞ norm. $\|\mathbf{M}\|_{\max} = \max_{i,j} |M_{i,j}|$ and $\|\mathbf{M}\|_{\min} = \min_{i,j} M_{i,j}$. Let S and S' be subsets of $[n]$. Then $\mathbf{M}_{S,S'}$ refers to the sub-matrix of \mathbf{M} with entries whose row index is in S and column index is in S' . The notation \tilde{O} is used to suppress poly-log factors of the dimension d . The function $\lambda(\mathbf{M})$ maps a matrix \mathbf{M} to the set of its eigenvalues. Similarly $\lambda_{\min}(\mathbf{M})$ and $\lambda_{\max}(\mathbf{M})$ map \mathbf{M} to its minimum and maximum eigenvalue, respectively. We define $\text{dvec}(\mathbf{M}) := \text{diag}(\text{vec}(\mathbf{M}))$, mapping a matrix \mathbf{M} to a diagonal matrix with the vectorized matrix \mathbf{M} on the main diagonal. Constants (either absolute or in K and d) introduced by our results are denoted by c_j ; the notation p_i is reserved only for the constants from Lemma 1.

2 Preliminaries

2.1 Background

Peng–Wei SDP The Peng–Wei SDP [8] is defined as

$$\max_{\mathbf{U}} \langle -\mathbf{D}, \mathbf{U} \rangle \quad \text{s.t.} \quad \mathbf{U} \in \mathcal{C} := \{\mathbf{U} : \mathbf{U} \geq 0; \mathbf{U}\mathbf{1} = \mathbf{1}; \text{tr}(\mathbf{U}) = K; \mathbf{U} \succeq 0\}. \quad (2)$$

For the data clustering problem, \mathbf{D} is defined by $D_{i,j} = \|x_i - x_j\|_2^2$. A solution is called “integer” if $U_{ij} = \frac{1}{|G_a|}$ if $i, j \in G_a$ and 0 otherwise, and it is said to correspond to the partition G . This is also called the “partnership matrix” of the clustering solution G , which we denote by $B(G)$. It can be shown that the dual SDP to (2) is

$$\begin{aligned} & \underset{y_{a,b}, y_a, y_T}{\text{minimize}} && 2 \sum_{a=1}^d y_a + K y_T \\ & \text{subject to} && \sum_{a=1}^d y_a \mathbf{R}_a + y_T \mathbf{I} \succeq -\mathbf{D} + \sum_{a \leq b} y_{a,b} \mathbf{I}_{a,b} \\ & && y_{a,b} \geq 0 \quad \text{for all } a \leq b, \end{aligned} \quad (3)$$

where the matrices $\mathbf{I}_{a,b}$ and \mathbf{R}_a are defined by $\mathbf{I}_{ab} = \frac{1}{2} (\mathbf{e}_a \mathbf{e}_b^T + \mathbf{e}_b \mathbf{e}_a^T)$ for all $a < b$, $\mathbf{I}_{aa} = \frac{1}{2} \mathbf{e}_a \mathbf{e}_a^T$, and $\mathbf{R}_a = \mathbf{1} \mathbf{e}_a^T + \mathbf{e}_a^T \mathbf{1}$.

Variable clustering in G-latent models The G -Latent model assumes the observed variables $\mathbf{X} = (X_1, \dots, X_d) \in \mathbb{R}^d$ can be partitioned into K unknown clusters $G^* = \{G_1^*, \dots, G_K^*\}$ such that variables in the same cluster share similar behavior. We denote $m := \min_i |G_i^*|$ and assume that $m \geq 3$. Further we also assume there exists a latent mean-zero random vector $\mathbf{Z} \in \mathbb{R}^K$ with covariance matrix $\text{Cov}(\mathbf{Z}) = \mathbf{C}^*$, such that $\mathbf{X} = \mathbf{A}\mathbf{Z} + \mathbf{E}$, for a zero mean error vector \mathbf{E} with independent entries. The $d \times K$ assignment matrix \mathbf{A} is defined as $A_{jk} = \mathbb{I}\{j \in G_k^*\}$. We denote $\text{Cov}(\mathbf{E}) = \mathbf{\Gamma}^*$, a diagonal matrix with entries $\Gamma_{jj}^* = \gamma_j^*$ for any $1 \leq j \leq d$. We also assume that the noise \mathbf{E} is independent of \mathbf{Z} . We assume that $\mathbf{Z} \sim N(0, \mathbf{C}^*)$ and $\mathbf{E} \sim N(0, \mathbf{\Gamma}^*)$, which implies $\mathbf{X} \sim N(0, \mathbf{\Sigma}^*)$ with $\mathbf{\Sigma}^* = \mathbf{A}\mathbf{C}^*\mathbf{A}^T + \mathbf{\Gamma}^*$. To be able to recover clusters, the latent variables cannot be too highly correlated, and we can define a distance between components of \mathbf{Z} as

$$\Delta(\mathbf{C}^*) =: \min_{j < k} \mathbb{E}(\mathbf{Z}_j - \mathbf{Z}_k)^2 > 0.$$

To recover the true group partition G^* , Bunea et al. [1] propose using (2) with $\mathbf{D} = \widehat{\mathbf{\Gamma}} - \widehat{\mathbf{\Sigma}}$, a penalized covariance matrix estimator (we refer to this as the PECOK estimator). Because a priori the group structure is unknown, an estimator $\widehat{\mathbf{\Gamma}}$ of $\mathbf{\Gamma}^*$ is somewhat involved so we omit the details here. For our purposes, we are only concerned with its rate of convergence in the max-norm. Bunea et al. [1] show that Lemma 1 holds when $\mathbf{X}_1, \dots, \mathbf{X}_n$ are generated from a G -Latent Model. The bound on $\Delta(\mathbf{C}^*)$ is shown to be minimax optimal.

Lemma 1 *There exist constants $p_0 - p_2$ (in K and d) such that if $\log d \leq p_0 n$, then with probability at least $1 - p_2/d^3$,*

$$\|\widehat{\mathbf{\Gamma}} - \mathbf{\Gamma}^*\|_\infty \leq p_1 \|\mathbf{\Gamma}^*\|_\infty \sqrt{\log d/n} =: \delta_{n,d}. \quad (4)$$

Furthermore, if

$$\Delta(\mathbf{C}^*) \gtrsim \|\mathbf{\Gamma}^*\|_\infty \left(\sqrt{\frac{\log d}{nm}} + \sqrt{\frac{\log d}{nm^2}} + \frac{d}{nm} + \frac{\log d}{n} \right) + \frac{\delta_{n,d}}{m},$$

then with probability at least $1 - p_3/d$ the optimizer to (2) is $\mathbf{X}^* = B(G^*)$ for some constants p_0, p_3 .

2.2 Related work

Solving the SDP An obvious approach is to simply solve the P-W SDP relaxation using standard second-order convex optimization methods (see Boyd and Vandenberghe [18, Ch. 11] for some examples). One well known approach to quickly solving certain SDPs is the matrix multiplicative weights (MMW) algorithm [19]. For the P-W SDP, the MMW algorithm requires $\tilde{O}(K^2 d^2 \alpha^{-2} \epsilon^{-2})$ arithmetic operations to find an ϵ -optimal⁴ solution and where α is related to a lower bound on the optimal value of a

⁴ In this context, ϵ is a multiplicative error.

rescaled version of the SDP. Typically, we have $\alpha = \mathcal{O}(d^{-1})$ giving a computational complexity of $\tilde{\mathcal{O}}(K^2 d^4 \epsilon^{-2})$.⁵

Another possibility is to solve the SDP using the Alternating Direction Method of Multipliers (ADMM) [20]. Recent work [15] takes this approach for a related SDP relaxation applied to the *bi-clustering problem*, but the focus there is on the statistical properties of the SDP relaxation not on deriving an algorithm with convergence guarantees. Like our approach, ADMM requires $\mathcal{O}(d^3)$ arithmetic operations per update, but there is no guarantee on its convergence rate. Instead, in this paper we convert the SDP into an equivalent eigenvalue maximization problem using a technique due to Renegar [17], which allows us to achieve better worst-case runtime bounds than existing methods. This is described in more detail in the next section.

Optimality certificates for data clustering In proving the tightness of (2) it is standard to derive an empirically testable condition on an instance of the clustering problem [10, 11, 16]. To do this, recent work on convex relaxations of K -means for data clustering takes a *dual optimality certificate* approach [10, 11, 16]. In general the dual optimality certificate approach is: (a) find an appropriate convex relaxation (denoted \mathcal{P}) and its dual (denoted \mathcal{D}) of (1), (b) given a candidate solution to \mathcal{P} construct a solution to \mathcal{D} with matching objective value, (c) derive a deterministic condition that can be checked on an instance of \mathcal{P} and proposed solution to \mathcal{P} that is sufficient for the construction in (b) to exist. The deterministic condition found in step (c) can then be analyzed to find the necessary assumptions on the data generating distribution to give the following guarantee: *with high probability a random instance of \mathcal{P} will satisfy the condition at the optimal solution G^* to \mathcal{P}* . To use the condition from step (c), all that remains is a way to “quickly” find optimal solutions to \mathcal{P} and then test the condition at the proposed optimal solution.

The dual solutions used in Awasthi and Bandeira [10], Iguchi et al. [11] differ from each other mainly in their choice of assignment to $y_{a,b}$ (likewise for our proposed certificate). The choice of $y_{a,b}$ in turn determines what testable condition one can derive and then leverage to prove tightness results and certify optimal clusterings. Unfortunately, Iguchi et al. [11] only offer a fast algorithm for the $K = 2$ case, and their method cannot be directly applied to variable clustering since it operates directly on the data points to be clustered, not merely the matrix \mathbf{D} . These certificates (and ours) benefit from Lemma 2 characterizing solutions to (3).

Lemma 2 (Theorem 4 [16]) *The following are equivalent: (a) \mathbf{B}^* is an optimal solution to (2), (b) every solution to (3) satisfies $y_{a,b} = 0$ for $a, b \in G_i^*$ and $\mathbf{Q}_{G_i^*, G_i^*} \mathbf{1} = 0$ for all i , and (c) every solution to (3) satisfies $\mathbf{y}_{G_i^*} = \mathbf{L}_{G_i^*, G_i^*}^{-1} (-\mathbf{D}_{G_i^*, G_i^*} \mathbf{1} - \mathbf{y}_T \mathbf{1})$. \mathbf{L} is a block-diagonal matrix determined by G^* , where the diagonal blocks are defined as $\mathbf{L}_{G_i^*, G_i^*} = |G_i^*| \mathbf{I} + \mathbf{1}\mathbf{1}^T$ and the off-diagonal blocks are zero.*

Other clustering approaches Spectral clustering [6, 21] is another approach, but these methods are tailored towards data clustering and are provably suboptimal [1] in terms of

⁵ We also did implement a MMW algorithm for P–W SDP, but found it did not converge in practice; we suspect this is due to the presence d^2 equality constraints since at each iteration of MMW these are not satisfied, but we did not investigate this further.

exact recovery in variable clustering. Heuristic approaches such as Lloyd's Algorithm [4] and CLINK [5] are fast, but in general do not find global optima.

Comparison to stochastic block model Variable clustering of data generated by the stochastic block model (SBM) has been heavily studied in recent years using the P–W SDP (and other related SDPs). In SBM, one wants to recover the true partition of d nodes using an observed $d \times d$ adjacency matrix where each entry is modeled as an independent Bernoulli random variable. Similar recovery guarantees to those described for the G -Latent model exist for SBM and use similar proof techniques—see Li et al. [22] for a recent survey [15,23,24]. An effective algorithm for solving the P–W SDP could therefore also benefit clustering in this regime as well.

3 The FORCE algorithm

In this section we first present the primal step, followed by the dual certificate and then a convergence guarantee for the P–W SDP on any instance \mathcal{D} .

3.1 Primal step

Because we consider clustering in the high-dimensional setting, a fast algorithm to solve (2) is critical. While second-order methods have an appealing iteration complexity, the per iteration cost is prohibitive for (2) because the cost of each iteration depends not only on the dimension d but also on the number of constraints—in (2), this is $\mathcal{O}(d^2)$. First-order methods, by contrast, may have a higher iteration complexity, but a lower per-iteration cost.

Algorithmic framework Informally, RSS [17] can be described as Nesterov's accelerated gradient method [25] and smoothing [26,27] applied to an eigenvalue maximization problem that is closely linked to the SDP of interest. Specifically, consider an SDP in standard form

$$\begin{aligned} & \underset{\mathbf{U}}{\text{minimize}} \langle \mathbf{D}, \mathbf{U} \rangle \\ & \text{s.t. } \mathbf{U} \in \mathcal{C} := \{\mathbf{U} : \langle \mathbf{A}_i, \mathbf{U} \rangle = b_i \text{ for } i = 1, \dots, p; \mathbf{U} \succeq 0\}, \end{aligned} \quad (5)$$

where $\mathbf{A}_i \in \mathcal{S}^{n \times n}$, $\mathbf{D} \in \mathcal{S}^{n \times n}$ and $b_i \in \mathbb{R}$; denote the optimal value of (5) by u^* . To apply RSS, we must specify as input any strictly feasible solution \mathbf{F} to (5).⁶ Given \mathbf{F} , a projection can be defined from \mathbf{F} onto the border of the positive semi-definite cone by $P_{\mathbf{F}}(\mathbf{U}) = \mathbf{F} + \frac{1}{1 - \lambda_{\min, \mathbf{F}}(\mathbf{U})}(\mathbf{U} - \mathbf{F})$, where $\lambda_{\min, \mathbf{F}}(\mathbf{U}) = \lambda_{\min}(\mathbf{F}^{-1/2}\mathbf{U}\mathbf{F}^{-1/2})$. $P_{\mathbf{F}}(\mathbf{U})$ lies at the intersection of the line segment between \mathbf{F} and \mathbf{U} and the positive semi-definite cone. Clearly if $\mathbf{U} \in \mathcal{S}_+^{n \times n}$ then $P_{\mathbf{F}}(\mathbf{U}) \in \mathcal{S}_+^{n \times n}$. Now, let $u_0 \in \mathbb{R}$ satisfying $u_0 < \langle \mathbf{D}, \mathbf{F} \rangle$. Renegar [17, Theorem 2.2] shows that if \mathbf{V}^* is a global optimum for

⁶ Actually Renegar [17] works in the setting $\mathbf{F} = \mathbf{I}$; what we present here is a slightly modified version and later we use the results of the corresponding, adjusted theoretical analysis.

$$\begin{aligned} & \underset{\mathbf{V}}{\text{maximize}} \lambda_{\min, F}(\mathbf{V}) \\ & \text{s.t. } \mathbf{V} \in \mathcal{C}_\lambda := \{\mathbf{V} : \langle \mathbf{A}_i, \mathbf{V} \rangle = b_i \text{ for } i = 1, \dots, p; \langle \mathbf{D}, \mathbf{V} \rangle = u_0\} \end{aligned} \quad (6)$$

then $P_{\mathbf{F}}(\mathbf{V}^*)$ is optimal for (5). In addition, if \mathbf{U}^* is optimal for (5), then $\mathbf{V}^* = \mathbf{F} + \frac{\langle \mathbf{D}, \mathbf{F} \rangle - u_0}{\langle \mathbf{D}, \mathbf{F} \rangle - u^*} (\mathbf{U}^* - \mathbf{F})$ is optimal for (6). To obtain faster convergence, Nesterov's smoothing technique can be applied and the objective function in (6) can be replaced by

$$f_{\mu, \mathbf{F}}(\mathbf{V}) = -\mu \log \sum_j \exp \left(-\lambda_j (\mathbf{F}^{-1/2} \mathbf{V} \mathbf{F}^{-1/2}) / \mu \right), \quad (7)$$

giving the smoothed problem

$$\begin{aligned} & \underset{\mathbf{V}}{\text{maximize}} f_{\mu, \mathbf{F}}(\mathbf{V}) \\ & \text{s.t. } \mathbf{V} \in \mathcal{C}_\lambda := \{\mathbf{V} : \langle \mathbf{A}_i, \mathbf{V} \rangle = b_i \text{ for } i \in [p]; \langle \mathbf{D}, \mathbf{V} \rangle = u_0\}. \end{aligned} \quad (8)$$

RSS internally applies Nesterov's accelerated projected gradient descent algorithm [28] to (8) several times through careful selection of initial iterates and after at most

$$T \leq 2R \|\mathbf{F}^{-1}\|_2^2 \sqrt{\log d} \left(\frac{1}{\epsilon} + \log_{5/4} \left(\frac{\langle \mathbf{D}, \mathbf{F} \rangle - u^*}{\langle \mathbf{D}, \mathbf{F} \rangle - u_0} \right) \right), \quad (9)$$

updates, the matrix \mathbf{U}_T output by RSS satisfies

$$\frac{\langle \mathbf{D}, \mathbf{U}_T \rangle - u^*}{\langle \mathbf{D}, \mathbf{F} \rangle - u^*} \leq \epsilon. \quad (10)$$

We direct the reader to Renegar [17, Theorem 7.2] for additional details. To summarize—applying RSS to an SDP requires strictly feasible \mathbf{F} , feasible \mathbf{U}_0 such that $\langle \mathbf{D}, \mathbf{U} \rangle < \langle \mathbf{D}, \mathbf{F} \rangle$, efficient computation of $\nabla f_{\mu, \mathbf{F}}$ and efficient computation of $\mathcal{P}_{\mathcal{C}_\lambda^\perp}$, the projection of the gradient onto $\mathcal{C}_\lambda^\perp = \{\mathbf{U} | \langle \mathbf{A}_i, \mathbf{U} \rangle = 0, \langle \mathbf{D}, \mathbf{U} \rangle = 0\}$.

Conversion to an eigenvalue maximization problem

First, we introduce the augmented variables

$$\mathbf{U}' = \left[\begin{array}{c|c} \mathbf{U} & \mathbf{0} \\ \hline \mathbf{0} & \text{dvec}(\mathbf{U}_C) \end{array} \right], \quad \mathbf{I}'_{a,b} = \left[\begin{array}{c|c} \mathbf{I}_{a,b} & \mathbf{0} \\ \hline \mathbf{0} & \frac{-1}{2} \text{diag}(\mathbf{e}_{a,b}) \end{array} \right], \quad (11)$$

where \mathbf{U}_C is a $d \times d$ matrix of slack-variables—likewise, $\text{dvec}(\mathbf{U}_C)$ is $d^2 \times d^2$ —and $\mathbf{e}_{a,b}$ denotes the d^2 -dimensional vector of 0s with 1s in only the $((a-1)d+b)$ th and $((b-1)d+a)$ th positions. We also define the variables \mathbf{R}'_a , \mathbf{I}' , and \mathbf{D}' as $(d^2+d) \times (d^2+d)$ matrices with upper left block equal to \mathbf{R}_a , \mathbf{I} and \mathbf{D} , respectively, and zero elsewhere. Up to the sign of the optimal value, (2) can thus be expressed as

$$\begin{aligned}
& \underset{\mathbf{U}'}{\text{minimize}} \langle \mathbf{D}', \mathbf{U}' \rangle, \\
& \text{s.t. } \mathbf{U}' \in \mathcal{C} := \left\{ \mathbf{U}' : \begin{array}{l} \langle \mathbf{I}'_{ab}, \mathbf{U}' \rangle = 0 \text{ for } a \leq b; \langle \mathbf{R}'_a, \mathbf{U}' \rangle = 2 \text{ for all } a; \\ \langle \mathbf{I}', \mathbf{U}' \rangle = K; \quad \mathbf{U}' \succeq 0. \end{array} \right\} \quad (12)
\end{aligned}$$

Given a strictly feasible solution \mathbf{F} and \mathbf{U}_0 such that $\langle -\mathbf{D}, \mathbf{F} \rangle < \langle -\mathbf{D}, \mathbf{U}_0 \rangle = -u_0$ to (2), we construct the pair

$$\mathbf{F}' = \left[\begin{array}{c|c} \mathbf{F} & \mathbf{0} \\ \hline \mathbf{0} & \text{dvec}(\mathbf{F}) \end{array} \right], \quad \mathbf{U}'_0 = \left[\begin{array}{c|c} \mathbf{U}_0 & \mathbf{0} \\ \hline \mathbf{0} & \text{diag}(\text{vec}(\mathbf{U}_0)) \end{array} \right]$$

necessary to apply RSS to (12). Finally, turning (12) into an equivalent eigenvalue maximization problem and applying Nesterov's smoothing gives

$$\begin{aligned}
& \underset{\mathbf{V}'}{\text{maximize}} f_{\mu, \mathbf{F}'}(\mathbf{V}'), \\
& \text{s.t. } \mathbf{V}' \in \mathcal{C}_\lambda := \left\{ \mathbf{V}' : \begin{array}{l} \langle \mathbf{I}'_{ab}, \mathbf{V}' \rangle = 0 \text{ for } a \leq b; \langle \mathbf{R}'_a, \mathbf{V}' \rangle = 2 \text{ for all } a; \\ \langle \mathbf{I}', \mathbf{V}' \rangle = K; \quad \langle \mathbf{D}', \mathbf{V}' \rangle = u_0. \end{array} \right\} \quad (13)
\end{aligned}$$

Importantly, we note that

$$\lambda(\mathbf{F}'^{-\frac{1}{2}} \mathbf{V}' \mathbf{F}'^{-\frac{1}{2}}) = \lambda(\mathbf{F}^{-\frac{1}{2}} \mathbf{V} \mathbf{F}^{-\frac{1}{2}}) \bigcup \{X_{i,j}/F_{i,j}^{-1}\}. \quad (14)$$

Constraint set projection

Formally, we can define the projection of \mathbf{V}' onto $\mathcal{C}_\lambda^\perp$ as

$$\underset{\mathbf{V}'' \in \mathcal{C}_\lambda^\perp}{\text{minimize}} \|\mathbf{V}' - \mathbf{V}''\|_{\mathbf{F}'}^2. \quad (15)$$

To obtain a closed form for $\mathcal{P}_{\mathcal{C}_\lambda^\perp}(\mathbf{V}')$, we utilize the fact that \mathbf{V}' has the specific structure given in (11) to solve the KKT optimality conditions [20, p. 267]; for (15), they are given below as (16). Because a projection onto a convex set has a unique minimizer, it will suffice to find any such point.

By deriving the KKT conditions and then simplifying, we obtain the following system of $d + 2$ equations in $d + 2$ unknowns (v and λ):

$$\begin{aligned}
\sum_{b=1}^d V_{ab} + \sum_{b=1}^d (V_C)_{ab} &= \sum_{b=1}^d v_b^* + d v_a^* + v_T^* + \left[\sum_{b=1}^d D_{ab} \right] \lambda^* \quad \text{for } a \in [d] \\
\text{tr}(\mathbf{V}) + \text{tr}(\mathbf{V}_C) &= \sum_{a=1}^d v_a^* + d v_T^* + \text{tr}(\mathbf{D}) \lambda^* \\
\text{tr}(\mathbf{D}\mathbf{V}) + \text{tr}(\mathbf{D}\mathbf{V}_C) &= 2 \sum_{a=1}^d \left[\sum_{b=1}^d D_{ab} \right] v_a^* + \text{tr}(\mathbf{D}) v_T^* + \text{tr}(\mathbf{D}\mathbf{D}) \lambda^*. \quad (16)
\end{aligned}$$

Solving (16), we get the projected matrix

$$\mathcal{P}_{\mathcal{C}_\lambda^\perp}(\mathbf{V}') = \begin{bmatrix} \mathbf{V}_* & \mathbf{0} \\ \mathbf{0} & \text{dvec}(\mathbf{V}_*) \end{bmatrix}, \quad \mathbf{V}_* = \frac{1}{2} \left[\mathbf{V} + \mathbf{V}_C - \sum_{a=1}^d \mathbf{R}_a v_a^* - v_T^* \mathbf{I} - \lambda^* \mathbf{D} \right]. \quad (17)$$

Remark 3 The last two sections highlight how the *effective* dimension of the problem is reduced by conversion to an eigenvalue maximization problem. The d^2 slack variables do not affect the cost of computing the projection $\mathcal{P}_{\mathcal{C}_\lambda^\perp}$. Likewise (14) shows that the cost of evaluating $f_{\mu, \mathbf{F}'}$ is dominated by that of computing the eigenvalues of the upper $d \times d$ diagonal block.

Existence of a strictly feasible solution

Unlike for the SDP's considered by Renegar [17], \mathbf{I} is not feasible for (2) as $K < d$, $\text{tr}(\mathbf{I}) = d \neq K$. We also note that the intuitive idea to find a possibly suboptimal clustering \widehat{G} and use $\mathbf{F} = B(\widehat{G})$ is not possible because strict feasibility for (2) requires all $\mathbf{F}_{ij} > 0$.

Nonetheless, there are valid choices of \mathbf{F} . Consider matrices of the form $\mathbf{F} = a\mathbf{I} + b\mathbf{1}\mathbf{1}^T$, where $a, b > 0$. Such matrices clearly satisfy $\mathbf{F}_{ij} > 0$ and $\mathbf{F} \succ 0$, so all that remains is to choose a and b such that $\langle \mathbf{F}, \mathbf{I} \rangle = K$ and $\mathbf{F}\mathbf{1} = \mathbf{1}$. Multiplying these expressions out, simplifying and solving the resulting system of equations gives $a = \frac{K-1}{d-1}$ and $b = \frac{d-K}{d^2-d}$. Lemma 3 summarizes the properties of \mathbf{F} .

Lemma 3 *Given d and K , define*

$$\mathbf{F}_{d,K} := \frac{K-1}{d-1} \mathbf{I} + \frac{d-K}{d^2-d} \mathbf{1}\mathbf{1}^T.$$

$\mathbf{F}_{d,K}$ is strictly feasible for (2) and $\|\mathbf{F}^{-1}\|_2 = \frac{d-1}{K-1}$.

Proof The first claim follows by the previous discussion and the second follows immediately from Lemma 10. \square

3.2 FORCE algorithm: dual step

Because all instances of (2) are strictly feasible, as shown in Lemma 3, then for any primal optimal solution there exists a dual solution such that its objective value is exactly equal to the primal. Unlike the primal problem, however, the dual does not lend itself easily to mapping a clustering onto a feasible solution for the SDP.

Let $\widehat{G} = \{\widehat{G}_1, \dots, \widehat{G}_K\}$ be the candidate clustering for which we want to find a dual solution. Because the goal is to certify optimality, consider $\widehat{G} = G^*$. Without loss of generality we can assume that the variables are ordered according to G^* , so that $\mathbf{B}^* = B(G^*)$ is block-diagonal. Denote by $d^* = \langle \mathbf{D}, \mathbf{B}^* \rangle$ and $\mathbf{Q} := \sum_{a=1}^d y_a \mathbf{R}_a + y_T \mathbf{I} + \mathbf{D} - \sum_{a \leq b} y_{a,b} \mathbf{I}_{a,b}$. Complementary slackness gives that for $a \in G_i^*$ and

$b \in G_i^*$, $y_{a,b} = 0$. Thus if we can “eliminate” the off-diagonal blocks in \mathbf{Q} , finding a dual solution should be very straightforward; this motivates Property 1.

Property 1 (*Large diagonal blocks property*) An instance \mathbf{D} of a clustering problem satisfies the Large Diagonal Blocks Property if there exists a feasible dual solution with value d^* such that the variables $y_{a,b}$ can be chosen to make off-diagonal blocks of the matrix \mathbf{Q} equal to $\mathbf{0}$.

Intuitively, we expect that in the variable clustering setting Property 1 will frequently hold. Because $-\mathbf{D}$ is an estimate of a covariance matrix for a generative model with block covariance structure, the diagonal blocks should dominate the off-diagonal blocks. What remains then is to search over assignments to y_a and y_T . In light of Lemma 2, the FORCE dual solution construction can be viewed as a function of y_T :

$$y_{G_i^*}(\mathbf{D}, y_T) = \mathbf{L}_i(-\mathbf{D}_i \mathbf{1} - y_T \mathbf{1}), \quad y_{a,b}(\mathbf{D}, y_T) = \begin{cases} 0, & \text{if } a = b \\ y_a + y_b + D_{a,b}, & \text{o/w,} \end{cases} \quad (18)$$

where $\mathbf{L}_i = \mathbf{L}_{G_i^*, G_i^*}^{-1}$ and $\mathbf{D}_i = \mathbf{D}_{G_i^*, G_i^*}$. By performing binary search over y_T , we obtain such a feasible dual solution if and only if Property 1 is satisfied. Computation of (18) is straightforward using Lemma 4 below.

Lemma 4 *Let \mathbf{L} be defined as above as in Sect. 3.2. Then \mathbf{L} is invertible and its inverse is block-diagonal, given by $\mathbf{L}_{G_i^*, G_i^*}^{-1} = \frac{1}{|G_i^*|} \mathbf{I} - \frac{1}{2|G_i^*|^2} \mathbf{1}\mathbf{1}^T$. Furthermore, $\lambda_{\max}(\mathbf{L}_{G_i^*, G_i^*}^{-1}) = |G_i^*|^{-1}$.*

Proof Using the Sherman-Morrison formula we can obtain the first claim, from which the second follows immediately. \square

We set the search interval for y_T to be $[0, C]$ for some C that can be selected at runtime. In practice to select the bound C , we will see from the proof of Theorem 2 in Sect. 4 can select

$$C = 2\|\widehat{\mathbf{F}}\|_{\infty} \left(\frac{d}{n} + \sqrt{\frac{d}{n}} \right).$$

Under the conditions of the theorem, there exists with high probability (tending to 1 as $d \rightarrow \infty$) a $y_T \in [0, C]$ such that the corresponding dual certificate is a feasible solution for (3). We note that in the statement of Theorem 2 there is a constant c_1 which above we have absorbed into the probability term.

3.3 Convergence rate of FORCE

Denoting by O_C a rounding oracle (e.g. Lloyd’s Algorithm or CLINK), O_C a certificate oracle that returns a dual feasible tuple $(\mathbf{y}_a, \mathbf{y}_{a,b}, y_T)$, and h the dual certificate search frequency, we can combine the primal update and dual certificate giving FORCE as Algorithm 1. On its own, the FORCE Primal Step offers an improved theoretical

guarantee over second-order interior point methods for (2). By appropriately choosing the dual certificate search frequency h , the convergence rate properties of the primal step transfer to FORCE. These results are summarized as Theorem 1.

Algorithm 1 First-Order Certifiable Clustering (FORCE)

Input: $0 < \epsilon < 1, \mathbf{D}, h, \mathbf{U}_0, \mathbf{F}$

Output: \widehat{G}

Run RSS with inputs $\epsilon, \mathbf{D}, \mathbf{U}_0, \mathbf{F}$ for T steps, denoting the iterate at time s by \mathbf{V}_s

for each update $s \in [T]$ such that $s \bmod h == 0$ **do**

$\mathbf{U}_s \leftarrow P_{\mathbf{F}}(\mathbf{V}_s), \widehat{G}_s \leftarrow O_R(\mathbf{U}_s)$

$(\mathbf{y}_a, \mathbf{y}_{a,b}, \mathbf{y}_T) \leftarrow O_C(\widehat{G}_s)$

If $\sum_{a=1}^d \mathbf{y}_a + K \mathbf{y}_T == \langle -\mathbf{D}, \mathbf{U}_s \rangle$, then **return** \widehat{G}_s

end for

return $O_R(P_{\mathbf{F}}(\mathbf{V}_T))$

Theorem 1 Let C and h be selected such that $C/h \leq 1$. Then, Algorithm 1 terminates after $\widetilde{\mathcal{O}}(d^6 K^{-2} \epsilon^{-1})$ arithmetic operations, giving an ϵ -optimal solution.

Proof We start by showing that the claim holds for RSS applied to (2). Note that for any \mathbf{U} and $\mathbf{V} \in \mathcal{C}$, $\|\mathbf{U} - \mathbf{V}\|_F \leq \sqrt{2}d$. For $\mathbf{F}_{d,K}$, applying Lemma 3 gives $\|\mathbf{F}_{d,K}^{-1}\|_2 = \frac{d-1}{K-1}$. The iteration complexity of RSS, (9), gives that the number of gradient updates required is at most

$$T = \left(2\sqrt{2\log d}\right) \frac{d(d-1)^2}{(K-1)^2} \left(\frac{1}{\epsilon} + \log_{5/4} \left(\frac{\langle \mathbf{D}, \mathbf{F} \rangle - u^*}{\langle \mathbf{D}, \mathbf{F} \rangle - u_0}\right)\right).$$

From (14), computing the gradient of $f_{\mu, \mathbf{F}}$ requires $\mathcal{O}(d^3)$ arithmetic operations and from (17) we see that projecting the gradient likewise requires $\mathcal{O}(d^3)$ operations. Therefore the running time of RSS is bounded by $\widetilde{\mathcal{O}}(d^6 K^{-2} \epsilon^{-1})$.

All that remains is to determine the cost of each query to the oracles O_R and O_C . Using CLINK as O_R , $\mathcal{O}(d^2)$ arithmetic operations are required per query. For O_C , we observe that at most $\mathcal{O}(C \log C)$ iterations of binary search are required. By pre-computing the transformations for $\mathbf{y}_{G_i^*}$, which requires at most $\mathcal{O}(d^3)$ arithmetic operations, each iteration of the search requires computing only the minimum eigenvalue of a d -dimensional matrix. This gives an overall bound of $\widetilde{\mathcal{O}}(Cd^3)$ on the number of arithmetic operations for O_C . Because there are at most T/h calls to O_C and we have that $C/h \leq 1$, the additional cost of all calls to O_C is $\widetilde{\mathcal{O}}(d^6 K^{-2} \epsilon^{-1})$, concluding the proof. \square

Remark 4 In Sect. 6, we will see that the dependence on K in practice is better than Theorem 1 predicts. One reason is that the computational complexity depends on $\|\mathbf{V}_0 - \mathbf{V}^*\|_F$ —the distance between the initial and optimal iterates. Our analytical bound on this quantity may be overly pessimistic because in practice we can use a heuristic clustering to construct \mathbf{V}_0 . For smaller K , the heuristic typically outputs a solution that is “closer” to optimal (indeed Fig. 3, Sect. 6, confirms this intuition).

4 Theoretical properties of the dual certificate

In the previous section, (18) defined the FORCE dual certificate in terms of y_T . In this section, we state and prove Theorem 2 showing that for variable clustering in G -Latent models, the certificate (18) exists at G^* w.h.p. whenever the cluster separation metric ΔC^* is above a minimal threshold. Our approach is in keeping with the literature on analyzing statistical properties of SDP relaxations, and we use similar proof strategies [10, 11, 15, 16]. Theorem 2 also shows that the P–W SDP is tight for G -Latent models since the SDP must be tight whenever the certificate exists.

Theorem 2 *Consider the variable clustering setting under the G -Latent model and assume $\log d \leq p_0 n$, where p_0 is the constant from Lemma 1. There exist constants c_1 , c_2 and c_3 such that if*

$$\Delta C^* \geq c_1 \|\Gamma^*\|_\infty \left(\sqrt{\frac{\log d}{nm}} + \sqrt{\frac{d}{nm^2}} + \frac{d}{nm} \right) + c_2 \sigma \sqrt{\frac{\log d}{n}},$$

then with probability at least $1 - c_3/d$ the FORCE Dual Certificate exists at G^ , where $\sigma = \max_i C_{i,i}^* + \|\Gamma^*\|_\infty$.*

4.1 General properties

Denoting by (\mathbf{D}, G^*) an instance of (2), we now characterize the factors that determine when Property 1 is satisfied—when, for each i , y_T can be selected such that (a) for all a and b , $y_{a,b}(\mathbf{D}, y_T) \geq 0$, and (b) that $\mathbf{Q}_i(\mathbf{D}, y_T) := \mathbf{D}_{G_i^*, G_i^*} + \sum_{a \in G_i^*} y_a \mathbf{R}_a + y_T \mathbf{I}$ is positive semidefinite. Importantly, problem (b) requires studying the behavior of points or variables only within the same group, greatly simplifying the analysis. Lemma 5 characterizes the behavior of the minimal eigenvalue of \mathbf{Q}_i .

Lemma 5 *Using the notation and quantities introduced above $\lambda_{\min}(\mathbf{Q}_i(\mathbf{D}, y_T)) = y_T + \min\{-y_T, \lambda_{\min}(\mathbf{Q}_i^\perp(\mathbf{D}))\}$, where*

$$\mathbf{Q}_i^\perp(\mathbf{D}) := \frac{(\mathbf{1}^T \mathbf{D}_{G_i^*, G_i^*} \mathbf{1}) \mathbf{1} \mathbf{1}^T}{|G_i^*|^2} - \frac{\mathbf{1} \mathbf{1}^T \mathbf{D}_{G_i^*, G_i^*} + \mathbf{D}_{G_i^*, G_i^*} \mathbf{1} \mathbf{1}^T}{|G_i^*|} + \mathbf{D}_{G_i^*, G_i^*}.$$

Proof To demonstrate the result, we first find an expression of the minimal eigenvalue of $\mathbf{Q}_i(\mathbf{D}, y_T)$ in terms of y_T and $\mathbf{D}_{G_i^*, G_i^*}$. Then we can apply Lemma 6 to obtain the result. One way to express the minimum eigenvalue is

$$\operatorname{argmin}_{\mathbf{v} \in \mathcal{S}^{|G_i^*|-1}} \underbrace{\mathbf{v}^T \mathbf{Q}_i(\mathbf{D}, y_T) \mathbf{v}}_{(i)}.$$

Now, for any $\mathbf{v} \in \mathcal{S}^{|G_i^*|-1}$ we can expand (i) as

$$\begin{aligned}
\text{(i)} &= \sum_{a=1}^{|G_i^*|} \sum_{b=1}^{|G_i^*|} v_a v_b Q_i(\mathbf{D}, y_T)_{a,b} \\
&= \sum_{a=1}^{|G_i^*|} v_a^2 y_T + \sum_{a=1}^{|G_i^*|} \sum_{b=1}^{|G_i^*|} v_a v_b (y_a + y_b) + \sum_{a=1}^{|G_i^*|} \sum_{b=1}^{|G_i^*|} v_a v_b D_{a,b} \\
&= y_T + \underbrace{\mathbf{v}^T \mathbf{D}_{G_i^*, G_i^*} \mathbf{v}}_{\text{(ii.a)}} + 2 \underbrace{\sum_{a=1}^{|G_i^*|} \sum_{b=1}^{|G_i^*|} v_a v_b y_a}_{\text{(ii.b)}}.
\end{aligned} \tag{19}$$

Via some algebra we obtain

$$\text{(ii.b)} = \sum_{a=1}^{|G_i^*|} v_a y_a \sum_{b=1}^{|G_i^*|} v_b = \sum_{a=1}^{|G_i^*|} v_a y_a \mathbf{v}^T \mathbf{1} = \mathbf{v}^T \mathbf{1} y_{G_i^*}^T \mathbf{v}.$$

From 4.1 above we see that the object of interest is now $\mathbf{1} y_{G_i^*}^T$, a $|G_i^*| \times |G_i^*|$ matrix. Recall that $\mathbf{y}_{G_i^*}^T$ is ultimately a function of y_T and \mathbf{D} . Fortunately, we already have explicit expressions for these quantities. In particular,

$$\begin{aligned}
\mathbf{1} y_{G_i^*}^T &= \mathbf{1} \left(-\mathbf{1}^T y_T - \mathbf{1}^T \mathbf{D}_{G_i^*, G_i^*} \right) \mathbf{L}_{G_i^*, G_i^*}^{-1} \\
&= -y_T \mathbf{1} \mathbf{1}^T \mathbf{L}_{G_i^*, G_i^*}^{-1} - \mathbf{1} \mathbf{1}^T \mathbf{D}_{G_i^*, G_i^*} \mathbf{L}_{G_i^*, G_i^*}^{-1} \\
&= -y_T \frac{1}{|G_i^*|} \mathbf{1} \mathbf{1}^T + \frac{1}{2|G_i^*|} \mathbf{1} \mathbf{1}^T - \frac{1}{|G_i^*|} \mathbf{1} \mathbf{1}^T \mathbf{D}_{G_i^*, G_i^*} + \frac{1}{2|G_i^*|^2} \mathbf{1} \mathbf{1}^T \mathbf{D}_{G_i^*, G_i^*} \mathbf{1} \mathbf{1}^T \\
&= -\frac{y_T}{2|G_i^*|} \mathbf{1} \mathbf{1}^T - \frac{1}{|G_i^*|} \mathbf{1} \mathbf{1}^T \mathbf{D}_{G_i^*, G_i^*} + \underbrace{\frac{1}{2|G_i^*|^2} \mathbf{1} \mathbf{1}^T \mathbf{D}_{G_i^*, G_i^*} \mathbf{1} \mathbf{1}^T}_{\text{(iii)}}.
\end{aligned} \tag{20}$$

In 20, observe that (iii) = $\frac{1}{2|G_i^*|^2} (\mathbf{1}^T \mathbf{D}_{G_i^*, G_i^*} \mathbf{1}) \mathbf{1} \mathbf{1}^T$. Plugging this back into 20 gives that

$$\mathbf{1} y_{G_i^*}^T = \frac{1}{2|G_i^*|^2} \left(\mathbf{1}^T \mathbf{D}_{G_i^*, G_i^*} \mathbf{1} - |G_i^*| y_T \right) \mathbf{1} \mathbf{1}^T - \frac{1}{|G_i^*|} \mathbf{1} \mathbf{1}^T \mathbf{D}_{G_i^*, G_i^*}. \tag{21}$$

We can substitute 21 into 4.1, yielding that

$$\begin{aligned}
\text{(ii.b)} &= \mathbf{v}^T \left(\frac{(\mathbf{1}^T \mathbf{D}_{G_i^*, G_i^*} \mathbf{1} - |G_i^*| y_T) \mathbf{1} \mathbf{1}^T}{|G_i^*|^2} - \frac{\mathbf{1} \mathbf{1}^T \mathbf{D}_{G_i^*, G_i^*}}{|G_i^*|} - \frac{\mathbf{D}_{G_i^*, G_i^*} \mathbf{1} \mathbf{1}^T}{|G_i^*|} \right) \mathbf{v} \\
&= \mathbf{v}^T \left(\mathbf{Q}_i^\perp(\mathbf{D}) - \frac{y_T}{|G_i^*|} \mathbf{1} \mathbf{1}^T - \mathbf{D}_{G_i^*, G_i^*} \right) \mathbf{v}.
\end{aligned}$$

Substituting back into (i), we get that

$$\lambda_{\min}(\mathbf{Q}_i(\mathbf{D}, y_T)) = y_T + \lambda_{\min}\left(-\frac{y_T}{|G_i^*|}\mathbf{1}\mathbf{1}^T + \mathbf{Q}_i^\perp(\mathbf{D})\right)$$

which is nearly the desired result. To proceed, we can see that $\frac{y_T}{|G_i^*|}\mathbf{1}\mathbf{1}^T$ and $\mathbf{Q}_i^\perp(\mathbf{D})$ lie in orthogonal spaces. This is a deterministic statement and does not depend on any particular clustering instance. Indeed, we can check that

$$\mathbf{1}^T \mathbf{Q}_i^\perp(\mathbf{D}) \mathbf{1} = 0$$

This is good, because then their respective eigenspaces are orthogonal giving

$$\lambda_{\min}(\mathbf{Q}_i(\mathbf{D}, y_T)) = y_T + \min\left\{-y_T, \lambda_{\min}(\mathbf{Q}_i^\perp(\mathbf{D}))\right\}. \quad \square$$

4.2 Properties under the G-latent model

Now the setup is that we have n samples of a d -dimensional random vector, denoted by $\mathbf{X} \in \mathbb{R}^{n \times d}$, \mathbf{D} is the PECOK penalized covariance estimator (Sect. 1). By writing $\mathbf{y}_{G_i^*}(\mathbf{X}, y_T)$ as a function of $\mathbf{X} \in \mathbb{R}^{n \times d}$, it is easy to observe that $\mathbb{E}[\mathbf{y}_{G_i^*}(\mathbf{X}, y_T)] \approx \frac{1}{2}(C_{i,i}^* - |G_i^*|^{-1}y_T)\mathbf{1}$ and therefore $\mathbb{E}[\mathbf{Q}_i(\mathbf{X}, y_T)] \approx y_T(\mathbf{I} - |G_i^*|^{-1}\mathbf{1}\mathbf{1}^T)$.⁷ From Lemma 5, whether or not the FORCE construction succeeds depends on how quickly $\mathbf{Q}_i(\mathbf{X}, y_T)$ concentrates about its mean (in terms of spectral norm) which in turn determines if y_T can be chosen small enough to ensure that the corresponding $y_{a,b}$ are feasible. Accordingly, the final two ingredients needed to prove Theorem 2 are Lemma 6 which controls the spectral radius of $\mathbf{Q}_i^\perp(\mathbf{X})$ and Lemma 7 which bounds $y_{a,b}$ in terms of y_T .

Lemma 6 Assume that $\log d \leq p_0 n$. Then,

$$\|\mathbf{Q}_i^\perp(\mathbf{X})\|_2 \leq c_1 \|\mathbf{\Gamma}^*\|_\infty \left(\frac{d}{n} + \sqrt{\frac{d}{n}} \right),$$

with probability at least $1 - \frac{c_2}{d^2}$ where c_1 and c_2 are constants.

Lemma 7 Let i and j be in $[K]$ and $i \neq j$. Define $y'_{a,b}(\mathbf{X}, y_T) := D_{a,b} + y_a + y_b$ for all $a \in G_i^*$ and $b \in G_j^*$. Under the assumption $\log d \leq p_0 n$,

$$y'_{a,b} \geq \frac{1}{2}\Delta(\mathbf{C}^*) - \frac{1}{m}y_T - c_1 \|\mathbf{\Gamma}^*\|_\infty \sqrt{\frac{\log d}{nm}} - c_2 \sigma \sqrt{\frac{\log d}{n}},$$

⁷ The equalities are inexact because we make no assumptions on the mean of $\hat{\mathbf{\Gamma}}$, only its convergence rate.

with probability at least $1 - c_3/d^3$, where c_1, c_2 and c_3 are constants, $\sigma = \max_i C_{i,i}^* + \|\mathbf{\Gamma}^*\|_\infty$.

Proof of Theorem 2

Now that we have all the necessary lemmas, we can prove the main result. First, we select

$$y'_T := \max_i \|\mathbf{Q}_i^\perp(X)\|_2,$$

ensuring that all $\mathbf{Q}_i(X)$ are positive semidefinite. By Lemma 6 and taking the union bound over all $i \in [K]$,

$$y'_T \leq c'_1 \|\mathbf{\Gamma}^*\|_\infty \left(\frac{d}{n} + \sqrt{\frac{d}{n}} \right),$$

with probability at least $1 - c'_2/d$.

Furthermore, by taking the union bound over all a and b not in the same group and using Lemma 7,

$$\min y'_{a,b} \geq \frac{1}{2} \Delta(\mathbf{C}^*) - \frac{1}{2m} y_T - \frac{1}{2m} y_T - c''_2 \|\mathbf{\Gamma}^*\|_\infty \sqrt{\frac{\log d}{nm}} - c''_3 \sigma \sqrt{\frac{\log d}{n}}$$

with probability at least $1 - c''_1/d$. Therefore, there exist constants c_1, c_2 and c_3 such that if we take $y_T = y'_T$ and

$$\Delta \mathbf{C}^* \geq c_1 \|\mathbf{\Gamma}^*\|_\infty \left(\sqrt{\frac{\log d}{nm}} + \sqrt{\frac{d}{nm^2}} + \frac{d}{nm} \right) + c_2 \sigma \sqrt{\frac{\log d}{n}},$$

then with probability at least $1 - c_3/d$, $\min_{a,b} y_{a,b} \geq 0$, demonstrating dual feasibility. Thus with probability at least $1 - c_3/d$, y'_T gives a feasible solution to (3), concluding the proof of the theorem. \square

5 Extension of FORCE to unknown K

The motivation and insight behind the FORCE algorithm remains the same when K is unknown, so we do not repeat the full discussion given in Sect. 3. When K is not known a priori, it can sometimes be estimated simultaneously by exchanging the trace constraint for an appropriately chosen trace penalty. In the variable clustering setting, Bunea et al. [1] show that (22) recovers the optimal solution to (1) without requiring K to be known a priori at the same cluster separation rate as the setting where K is known.

K-means adaptive SDP

We refer to

$$\underset{\mathbf{U}}{\text{maximize}} \langle -\mathbf{D} - \widehat{\kappa}\mathbf{I}, \mathbf{U} \rangle \text{ s.t. } \mathbf{U} \in \mathcal{C} := \{\mathbf{U} : \mathbf{U} \succeq 0; \mathbf{U}\mathbf{1} = \mathbf{1}; \mathbf{U} \succeq 0\}, \quad (22)$$

as the K -means Adaptive SDP due to its use in *adaptively* selecting the number of clusters and finding the optimal clustering simultaneously. The trace penalty is defined by a data driven tuning parameter $\widehat{\kappa}$, the selection of which is given later in (29). It is beyond the scope of this work to consider the theoretical properties of (22) for data clustering and the remainder of this section focuses on the variable clustering setting.

Like the case when K is known, the dual SDP has the form

$$\begin{aligned} &\underset{y_{a,b}, y_a}{\text{minimize}} \quad 2 \sum_{a=1}^d y_a \\ &\text{subject to} \quad \sum_{a=1}^d y_a \mathbf{R}_a + \widehat{\kappa}\mathbf{I} \succeq -\mathbf{D} + \sum_{a \leq b} y_{a,b} \mathbf{I}_{a,b} \\ &\quad y_{a,b} \geq 0 \quad \text{for all } a \leq b. \end{aligned} \quad (23)$$

Conversion to eigenvalue maximization

The conversion to standard form and an eigenvalue maximization problem is nearly identical to the case when K is known, so the derivation is omitted. Using the notation from Sect. 3, in standard form (22) becomes

$$\begin{aligned} &\underset{\mathbf{U}'}{\text{minimize}} \quad \langle \mathbf{D}' + \widehat{\kappa}\mathbf{I}', \mathbf{U}' \rangle, \\ &\text{s.t. } \mathbf{U}' \in \{\mathbf{U}' : \langle \mathbf{I}'_{ab}, \mathbf{U}' \rangle = 0 \text{ for } a \leq b; \langle \mathbf{R}'_a, \mathbf{U}' \rangle = 2 \text{ for all } a; \mathbf{U}' \succeq 0\}, \end{aligned} \quad (24)$$

with corresponding smoothed eigenvalue maximization problem

$$\begin{aligned} &\underset{\mathbf{V}'}{\text{maximize}} \quad f_{\mu, \mathbf{F}'}(\mathbf{V}'), \\ &\text{s.t. } \mathbf{V}' \in \mathcal{C}_\lambda := \left\{ \mathbf{V}' : \langle \mathbf{I}'_{ab}, \mathbf{V}' \rangle = 0 \text{ for } a \leq b; \langle \mathbf{D}' + \widehat{\kappa}\mathbf{I}', \mathbf{V}' \rangle = u_0; \right. \\ &\quad \left. \langle \mathbf{R}'_a, \mathbf{V}' \rangle = 2 \text{ for all } a. \right\} \end{aligned} \quad (25)$$

Constraint projection

As in the case when K is known, we must derive the projection onto $\mathcal{C}_\lambda^\perp$. Denoting by $\mathbf{H} := \mathbf{D}' + \widehat{\kappa}\mathbf{I}'$ and solving the KKT conditions, we get the following system of $d+1$ equations:

$$\sum_{b=1}^d V_{ab} + \sum_{b=1}^d (V_C)_{ab} = \sum_{b=1}^d v_b^* + d v_a^* + \left[\sum_{b=1}^d H_{ab} \right] \lambda^* \quad \text{for } a \in [d]$$

$$\text{tr}(\mathbf{H}\mathbf{V}) + \text{tr}(\mathbf{H}\mathbf{V}_C) = 2 \sum_{a=1}^d \left[\sum_{b=1}^d H_{ab} \right] v_a^* + \text{tr}(\mathbf{H}\mathbf{H})\lambda^*. \quad (26)$$

Solving (26), we can get the projected matrix

$$\mathcal{P}_{C_\lambda^\perp}(\mathbf{V}') = \left[\begin{array}{c|c} \mathbf{V}_* & \mathbf{0} \\ \hline \mathbf{0} & \text{dvec}(\mathbf{V}_*) \end{array} \right], \quad \mathbf{V}_* = \frac{1}{2} \left[\mathbf{V} + \mathbf{V}_C - \sum_{a=1}^d \mathbf{R}_a v_a^* - \lambda^* (\mathbf{D} + \hat{\kappa} \mathbf{I}) \right]. \quad (27)$$

Existence of a feasible solution

Clearly for (22) $\mathbf{F} = \mathbf{I}$ is feasible, but unfortunately it is not strictly feasible so a different choice of \mathbf{F} is required. Unlike in the case when K is known, there is no trace constraint and therefore we can find an \mathbf{F} such that for any d , $c_1^{-1} \leq \lambda_{\min}(\mathbf{F}) \leq \lambda_{\max}(\mathbf{F}) \leq c_1$, for some $c_1 \geq 1$. In particular, we can choose

$$\mathbf{F} := \frac{1}{2} \mathbf{I} + \frac{1}{2d} \mathbf{1}\mathbf{1}^T,$$

which clearly is strictly feasible for (22). Using the Sherman-Morrison formula, we obtain that

$$\mathbf{F}^{-1} = 2\mathbf{I} - \frac{1}{d} \mathbf{1}\mathbf{1}^T.$$

Furthermore, it is easy to see that $\frac{1}{2} \leq \lambda_{\min}(\mathbf{F}) \leq \lambda_{\max}(\mathbf{F}) \leq 2$. This shows that in the case where K is unknown, we pay only a factor of 4 penalty for \mathbf{I} not being strictly feasible. This is a sharp contrast to the fixed K case, where the penalty is much higher.

5.1 FORCE dual step

In order to find a dual certificate, we first characterize the form of optimal solutions to (23). Lemma 8 characterizes all primal, dual optimal pairs for (22), just as Lemma 2 does for the case where K is known a priori.

Lemma 8 *The following are equivalent: (a) \mathbf{B}^* is an optimal solution to (2), (b) every solution to (3) satisfies $y_{a,b} = 0$ for $a, b \in G_i^*$ and $\mathbf{Q}_{G_i^*, G_i^*}^* \mathbf{1} = 0$ for all i , and (c) every solution to (3) satisfies $\mathbf{y}_{G_i^*}^* = \mathbf{L}_{G_i^*, G_i^*}^{-1} (-\mathbf{D}_{G_i^*, G_i^*}^* \mathbf{1} - \hat{\kappa} \mathbf{1})$.*

Proof The proof of Lemma 8 follows from complementary slackness and by rearranging a system of linear equations. For more details, we direct the reader to Iguchi et al. [16, Theorem 4]. \square

Now, observe that $\hat{\kappa}$ plays the same role in (23) as y_T does in (3). Therefore the results and intuition regarding the dual construction still hold, but now we do not

search over y_T . Instead we just invert a linear system and check feasibility. The dual solution to (23) corresponding to G^* is

$$y_{G_i^*}(\mathbf{D}) = \mathbf{L}_i(-\mathbf{D}_i \mathbf{1} - \widehat{\kappa} \mathbf{1}), \quad y_{a,b}(\mathbf{D}) = \begin{cases} 0, & \text{if } a = b \\ y_a + y_b + D_{a,b}, & \text{o/w,} \end{cases} \quad (28)$$

where $\mathbf{L}_i = \mathbf{L}_{G_i^*, G_i^*}^{-1}$ and $\mathbf{D}_i = \mathbf{D}_{G_i^*, G_i^*}$. Just as the case when K is known, we can use the explicit dual solution construction (28) to certify the optimality.

5.2 The FORCE algorithm

Algorithm 1 requires only minor modification to be applied to (22). First, we apply RSS to (22) instead of (2). Second we replace the certificate oracle O_C with one based on (23). Finally, we replace the rounding oracle O_R with a procedure that can simultaneously cluster the projected iterate and select K . One such approach is to choose $K = \text{round}(\text{tr}(P_F(\mathbf{V}_s)))$ and then proceed by applying either CLINK or Lloyd's algorithm using the selected K . This approach is justified in that when (22) is tight, the optimal solution is $B(G^*)$ [which satisfies $\text{tr}(B(G^*)) = K$]. Thus once the iterate is sufficiently close to optimal, this method will of course recover G^* . In practice, however, one could consider using CLINK for the clustering step to obtain solutions for all K in only $\mathcal{O}(d^2)$ arithmetic operations. By plotting the mean-squared error (MSE) of each clustering solution against K , the elbow method can be used to select the number of clusters [29,30].

5.3 Theoretical results

Mirroring our results for fixed K , Theorem 3 gives a worst-case bound on the computational complexity of FORCE for (22). Proofs of the results in this section are nearly identical to those in Sects. 3 and 4.

Theorem 3 *For any certificate search frequency h , Algorithm 1 applied to solving (22) terminates in at most $\tilde{\mathcal{O}}(d^4 \epsilon^{-1})$ arithmetic operations, giving an ϵ -optimal solution.*

We next address how to choose $\widehat{\kappa}$ in practice. The choice is driven by the following consideration: When does the dual certificate exist and when is the SDP relaxation tight? These questions are intimately connected, so it is not surprising that our choice of $\widehat{\kappa}$ is similar to previous work on the tightness of (22) [1]. Specifically, we select

$$\widehat{\kappa} := 5 \|\widehat{\mathbf{F}}\|_\infty \left(\frac{d}{n} + \sqrt{\frac{d}{n}} \right) \quad (29)$$

for variable clustering in G -Latent models when K is unknown. This choice is driven purely by the dual certificate construction, and the derivation of it can be found in the proof of Theorem 4, in “Appendix A”. We note that the choice of constant in $\widehat{\kappa}$ could be altered, but we do not explore whether or not some other choice is preferable.

Importantly $\hat{\kappa}$ is data-driven in the sense that its selection requires *no knowledge* of the parameters of the generating distribution.

Theorem 4 *Consider the variable clustering setting under the G-Latent model and assume $\log d \leq p_0 n$, where p_0 is the constant from Sect. 2.1. If $\hat{\kappa} = 5\|\hat{\Gamma}\|_\infty (d/n + \sqrt{d/n})$, there exist constants c_1, c_2 and c_3 such that if*

$$\Delta C^* \geq c_1 \|\Gamma^*\|_\infty \left(\sqrt{\frac{\log d}{nm}} + \sqrt{\frac{d}{nm^2}} + \frac{d}{nm} \right) + c_2 \sigma \sqrt{\frac{\log d}{n}},$$

then with probability at least $1 - c_3/d$ the FORCE Dual Certificate exists at G^ , where $\sigma = \max_i C_{i,i}^* + \|\Gamma^*\|_\infty$.*

Remark 5 The additional cost of constraining K to be fixed is imposed directly by the trace constraint. It is somewhat surprising that we should obtain a significantly better worst-case complexity bound, for certain K , when we have *less information* about the structure of the problem at hand. For this reason we suspect it may not be impossible to obtain the same worst case bound if we impose a fixed K in the problem formulation.

The adaptive formulation, (22), can also be applied to data clustering, and we suspect the FORCE algorithm may have strong theoretical properties in that setting when K is unknown, but that analysis is beyond the scope of this work.

6 Numerical results

We evaluate FORCE by validating Theorem 2 empirically, comparing the FORCE primal step to other methods for solving (2), and comparing the performance of FORCE with clustering heuristics. Note that the third evaluation captures a combination of the properties of (2) and of FORCE, since it is an inexact solver for the SDP. We focus on the case where K is known due to space constraints, but similar results can be obtained for K unknown (see “Appendix C”).

Implementation details

We implement FORCE in R and because FORCE is not a traditional primal-dual algorithm and does not make dual updates, we use an early stopping rule as the termination condition. Specifically, for a given s and δ , if at any iteration t ,

$$\max_{u \in [t-s+1, t]} \frac{f_{\mu, \mathbf{F}}(\mathbf{V}_u) - f_{\mu, \mathbf{F}}(\mathbf{V}_{t-s})}{f_{\mu, \mathbf{F}}(\mathbf{V}_{t-s})} < \delta,$$

then the algorithm terminates. For all experiments we use $(s, \delta) = (100, 10^{-4})$. An adaptive restart rule is used for the accelerated PGD weighting coefficients [31]. In practice, we also found that the warm-start step of RSS was unnecessary to achieve good performance, and the following simple heuristic gave at least as good results in

terms of the final output: let $\mathbf{U}_0 = \frac{1}{d}B(\mathcal{K}(\mathbf{D}, K)) + \frac{d-1}{d}\mathbf{F}$, then perform accelerated PGD on $f_{\mu, \mathbf{F}}$ starting with initial iterate $\mathbf{V}_0 := \mathbf{U}_0$ for a fixed number of iterations N to obtain $\mathbf{U}_1 := P_{\mathbf{F}}(\mathbf{V}_N)$. The matrix \mathbf{U}_1 is then used in place of the original warm-start step of RSS. We found that this heuristic produced in practice a matrix \mathbf{U}_1 satisfying the warm-start requirements of RSS.

Benchmarking framework

To benchmark the algorithms we use a Dell XPS 9570 with an i7-8750H processor. All algorithms are limited to 6 computational threads and the R build is linked against Intel's MKL BLAS implementation to ensure a fair comparison with MATLAB.

We compare FORCE with several alternatives. Primarily this shows how several alternative algorithms scale. We compare against a MATLAB implementation using MOSEK [32] as the solver, a MATLAB implementation using SDPNAL+ [33], and an ADMM algorithm to solve (2) due to Ames [15].⁸ For short, we refer to these algorithms as MOSEK, SDPNAL+, and ADMM respectively. MOSEK and SDPNAL+ are run using the default options and ADMM is run using the same options as in Ames [15]. FORCE refers to Algorithm 1 and FORCE-P denotes just the primal step of FORCE with no dual certificate search.

Generative model

Recall that the generating distribution of a G -Latent model with d observed variables and K latent factors can be described in terms of $(G^*, \Theta^*, \Gamma^*)$. We first select a graph structure for \mathbf{Z} and then once the graph structure is constructed, the latent precision matrix is defined as $\Theta^* = \rho \mathbf{W} + (|\lambda_{\min}(\mathbf{W})| + 0.2)\mathbf{I}$, where \mathbf{W} is the adjacency matrix of the generated graph. We take $\Gamma^* = \gamma \mathbf{I}$ for some constant γ to be specified later. Because we work in the high-dimensional regime, we generate $n = d$ samples for each simulation.

Throughout we use the scale-free generative model to construct the dependency structure amongst the latent variables \mathbf{Z} . It is a model for network data, whose degree distribution follows a power law and we generate the graph one node at a time, starting with a 2 node chain. For nodes $s \in \{3, \dots, K\}$, node s is added and one edge is added between s and one of the $s - 1$ previous nodes. At each step, if k_i denotes the current degree of node i in the graph, the probability that node t and node i are connected is $p_i = k_i / (\sum_i k_i)$. By construction, such a graph always has K edges.⁹

Dual certificate

To assess the effect of noise on the existence of the dual certificates, (3) and (23), we select two designs (one for $d = 250$ and $d = 500$) and vary the level of γ . Figure 1 contains the results and we can observe a sharp phase transition as γ increases. Interestingly only slightly less noise is required for the certificate to exist when K is

⁸ The authors have made the code available on-line at <http://bpames.people.ua.edu/software.html>.

⁹ Similar results can be obtained for other graph structures, such as Band or Hub graphs.

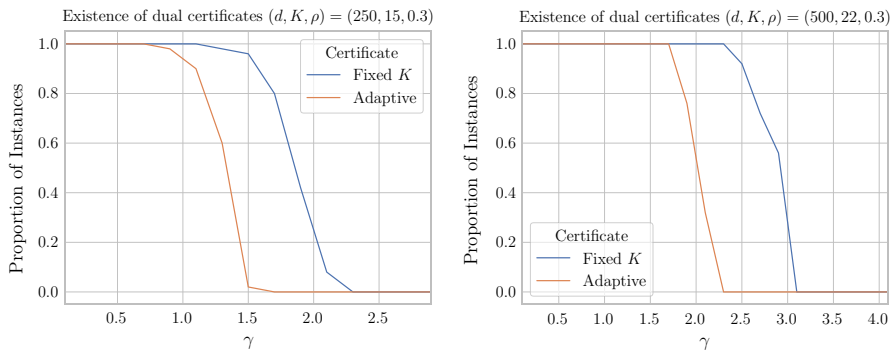


Fig. 1 Proportion of randomly generated instances for which a certificate exists at G^*

not known versus when K is fixed a priori. Another interpretation of Fig. 1 is that it shows the sharp phase transition under which the P–W SDP is tight for G -Latent models as a function of noise γ .

6.1 FORCE versus other algorithms for solving the P–W SDP

Low-dimensional problem sizes The goal of the simulations in lower dimensions is to evaluate the scaling of the various alternatives. We vary both d and the level of noise γ , evaluating six different settings. For each setting, we generate 100 random instances and run each algorithm. Relative error is measured in terms of the objective value of (2) and we assume $v^* = \langle -\mathbf{D}, B(G^*) \rangle$.

Table 1 gives the results, showing that even for $d = 120$ MOSEK (a traditional interior point solver) is computationally expensive. Note that in some cases FORCE achieves an error of 0—this occurs because FORCE performs a search for a dual certificate and finds the *exact* optimal solution. In addition, we observe the FORCE-P typically has a higher relative error than the other methods. This is to be expected because there are two sources of error: both optimization error and approximation error (introduced by the smoothing). We also can observe that the precision of the primal solution is not critical because despite FORCE-P having lower precision, FORCE frequently finds the exact optimal solution. Figure 3 below also indicates that the accuracy of FORCE-P achieved in these experiments is sufficient for exact recovery with high probability (even without performing dual certificate search).

High-dimensional problem sizes For higher dimensional setups, $d = 500$, we find that both MOSEK and SDPNAL+ require too much memory and computational resources to run the simulations on our benchmarking platform (a high-end consumer PC), and therefore we compare only ADMM, FORCE and FORCE-P. We compare both high ($\gamma = 3.0$) and low ($\gamma = 1.0$) noise setups for $K = 9, 22, 50, 100$ which range from $\mathcal{O}(\log d)$ to $\mathcal{O}(d)$. For each design, 100 random instances were generated and the results are reported in Table 2. To compute relative error we assume $v^* = \langle -\mathbf{D}, B(G^*) \rangle$.

Table 1 Benchmark results for low dimensional designs comparing FORCE and FORCE-P with MOSEK, SDPNAL+ and ADMM

| (d, k, ρ, γ) | $(50, 5, 0.3, 0.3)$ | | $(50, 5, 0.3, 1.0)$ | |
|------------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| Algorithm | Rel. Err. | Time (s) | Rel. Err. | Time (s) |
| MOSEK | 2.31×10^{-8} | 4.13×10^{-1} | 3.82×10^{-8} | 4.24×10^{-1} |
| SDPNAL+ | 5.55×10^{-7} | 3.97×10^{-1} | 4.91×10^{-7} | 3.87×10^{-1} |
| ADMM | 2.10×10^{-7} | 2.08×10^{-2} | 3.07×10^{-7} | 3.29×10^{-2} |
| FORCE | 0.00×10^0 | 1.31×10^{-3} | 1.10×10^{-8} | 1.97×10^{-2} |
| FORCE-P | 6.86×10^{-3} | 9.18×10^{-2} | 9.26×10^{-3} | 1.06×10^{-1} |
| (d, k, ρ, γ) | $(85, 9, 0.3, 0.3)$ | | $(85, 9, 0.3, 1.0)$ | |
| Algorithm | Rel. Err. | Time (s) | Rel. Err. | Time (s) |
| MOSEK | 1.04×10^{-8} | 3.42×10^0 | 2.56×10^{-8} | 3.58×10^0 |
| SDPNAL+ | 7.77×10^{-7} | 1.38×10^0 | 2.20×10^{-6} | 1.37×10^0 |
| ADMM | 1.42×10^{-7} | 7.48×10^{-2} | 2.69×10^{-7} | 9.14×10^{-2} |
| FORCE | 0.00×10^0 | 2.65×10^{-2} | 0.00×10^0 | 8.85×10^{-2} |
| FORCE-P | 7.60×10^{-3} | 3.54×10^{-1} | 9.96×10^{-3} | 4.12×10^{-1} |
| (d, k, ρ, γ) | $(120, 12, 0.3, 0.3)$ | | $(120, 12, 0.3, 1.0)$ | |
| Algorithm | Rel. Err. | Time (s) | Rel. Err. | Time (s) |
| MOSEK | 2.17×10^{-8} | 1.83×10^1 | 3.96×10^{-8} | 1.83×10^1 |
| SDPNAL+ | 7.69×10^{-8} | 3.28×10^0 | 1.09×10^{-6} | 3.01×10^0 |
| ADMM | 1.07×10^{-7} | 4.41×10^{-2} | 1.43×10^{-7} | 4.97×10^{-2} |
| FORCE | 0.00×10^0 | 4.62×10^{-2} | 0.00×10^0 | 1.84×10^{-1} |
| FORCE-P | 7.25×10^{-3} | 7.18×10^{-1} | 1.12×10^{-2} | 8.60×10^{-1} |

When they converge, all three methods exhibit similar running times. However for designs closer to the threshold above which exact recovery is possible, ADMM often does not converge (in all cases, this was caused by the sequence of updates diverging). By comparison, FORCE and FORCE-P converge on all instances encountered during benchmarking. Table 2 shows that FORCE always has 0 error, i.e. that it achieves perfect recovery. From this we conclude FORCE-P finds a solution to (2) that is “close-enough” to the optimal solution that by rounding and finding a dual certificate, FORCE achieves exact recovery.

Table 2 also reveals that as K increases, it takes longer for FORCE to solve (2). While this runs contrary to the predictions of Theorem 1, it does align with our intuition as the effective sample size per group is decreasing in K . For a discussion, see Remark 4 in Sect. 3.

Scaling We compare the runtimes of FORCE, FORCE-P and ADMM versus d . In these experiments we vary d from 150 to 500 and set $(K, \rho, \gamma) = (d/10, 0.3, 1.0)$. These settings are chosen (in light of the results in Table 2) to ensure ADMM converges most of the time. The results are displayed as Fig. 2, and in the log-log plot, we see

Table 2 Benchmark results for high dimensional designs comparing ADMM, FORCE and FORCE-P

| Alg. | (d, k, ρ, γ) | Rel. Err. | Rel. Err. $ \mathcal{F}$ | Conv. (%) | Time (s) |
|---------|------------------------|-----------------------|--------------------------|-----------|-----------------------|
| ADMM | (500, 9, 0.3, 1.0) | 3.71×10^{-7} | 3.71×10^{-7} | 100.0 | 2.39×10^0 |
| FORCE | | 0.00×10^0 | 0.00×10^0 | 100.0 | 3.20×10^{-1} |
| FORCE-P | | 9.12×10^{-3} | 9.12×10^{-3} | 100.0 | 1.77×10^1 |
| ADMM | (500, 9, 0.3, 3.0) | 5.38×10^{-7} | 5.38×10^{-7} | 96.0 | 3.41×10^0 |
| FORCE | | 0.00×10^0 | 0.00×10^0 | 100.0 | 1.29×10^0 |
| FORCE-P | | 2.39×10^{-2} | 2.40×10^{-2} | 100.0 | 2.34×10^1 |
| ADMM | (500, 22, 0.3, 1.0) | 1.86×10^{-7} | 1.86×10^{-7} | 100.0 | 3.24×10^0 |
| FORCE | | 0.00×10^0 | 0.00×10^0 | 100.0 | 4.03×10^0 |
| FORCE-P | | 1.70×10^{-2} | 1.70×10^{-2} | 100.0 | 2.34×10^1 |
| ADMM | (500, 22, 0.3, 3.0) | 7.99×10^{-7} | 7.99×10^{-7} | 56.0 | 5.90×10^0 |
| FORCE | | 0.00×10^0 | 0.00×10^0 | 100.0 | 8.48×10^0 |
| FORCE-P | | 2.29×10^{-2} | 2.29×10^{-2} | 100.0 | 1.99×10^1 |
| ADMM | (500, 50, 0.3, 1.0) | 2.96×10^{-8} | 2.96×10^{-8} | 96.0 | 3.13×10^0 |
| FORCE | | 0.00×10^0 | 0.00×10^0 | 100.0 | 1.14×10^1 |
| FORCE-P | | 1.69×10^{-2} | 1.72×10^{-2} | 100.0 | 2.37×10^1 |
| ADMM | (500, 50, 0.3, 3.0) | 5.84×10^{-8} | 5.84×10^{-8} | 64.0 | 3.33×10^0 |
| FORCE | | 0.00×10^0 | 0.00×10^0 | 100.0 | 1.46×10^1 |
| FORCE-P | | 2.11×10^{-2} | 1.99×10^{-2} | 100.0 | 2.45×10^1 |
| ADMM | (500, 100, 0.3, 1.0) | 1.32×10^{-8} | 1.32×10^{-8} | 20.0 | 3.53×10^0 |
| FORCE | | 0.00×10^0 | 0.00×10^0 | 100.0 | 1.56×10^1 |
| FORCE-P | | 1.16×10^{-2} | 1.08×10^{-2} | 100.0 | 2.65×10^1 |
| ADMM | (500, 100, 0.3, 3.0) | N/A | N/A | 0.0 | N/A |
| FORCE | | 0.00×10^0 | N/A | 100.0 | 2.57×10^1 |
| FORCE-P | | 2.08×10^{-2} | N/A | 100.0 | 2.88×10^1 |

 \mathcal{F} is the event ADMM converges on a problem instance

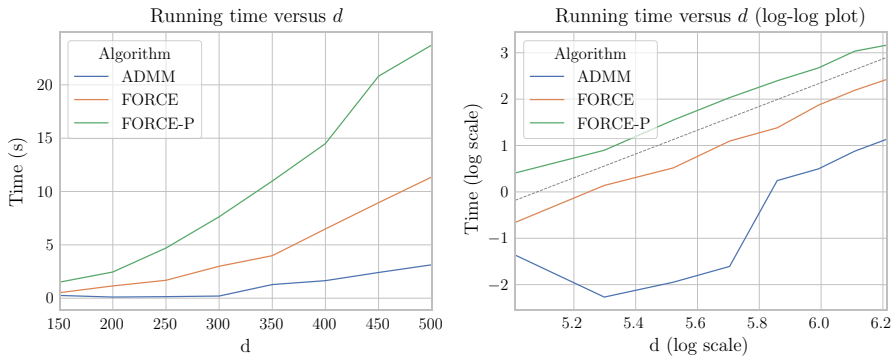


Fig. 2 Growth of running time as d increases

that all three algorithms appear to display a similar growth rate once d is sufficiently large.

Using the runtimes for FORCE we perform an OLS regression of $\log(T)$ on $\log(d)$, giving the relation $\log(T) \propto 2.56 \log(d)$. In practice, therefore, we observe a computational complexity that grows with $d^{2.56}$ for fixed ϵ . A line with this slope is displayed on the log-log plot for comparison. In contrast, Theorem 1 gives a worst-case computational complexity bound of $\tilde{O}(d^4)$ for fixed ϵ , which in light of these results, seems overly pessimistic.

6.2 FORCE and the P–W SDP versus heuristic methods

Lastly, we compare FORCE applied to the P–W SDP to heuristic methods to cluster the data. Heuristic methods are typically fast, and if they were to offer similar performance in practice, it may not make sense to solve the P–W SDP using FORCE or any other algorithm. As we show in the experiments described below, this is not the case. We compare against Lloyd’s algorithm with kmeans++ initialization as this gave better results than either CLINK or Lloyd’s algorithm with random initialization. We consider the design $(d, K, \rho) = (500, 22, 0.3)$ and study the effect of γ on the performance gap of FORCE and the P–W SDP versus heuristic methods.

First we compare clustering applied \mathbf{V}_T , the final iterate output by FORCE-P, to clustering applied to either $\mathbf{D} = \hat{\mathbf{\Sigma}} - \hat{\mathbf{\Gamma}}$ or $\hat{\mathbf{\Sigma}}$. Denoting by $\mathcal{K}(\mathbf{M}, K)$ the algorithm that takes matrix \mathbf{M} and runs Lloyd’s algorithm with kmeans++ initialization returning a partition \hat{G} . The metrics used to evaluate the output are $d_1(\hat{G}, G^*) = \mathbb{I}[\hat{G} = G^*]$ and $d_2(\hat{G}, G^*) = n^{-1} \sum_{i=1}^K \max_j |\hat{G}_i \cap G_j^*|$, which captures the number of correctly assigned variables.

Row one in Fig. 3 shows $\mathbb{E}[d_i(\mathcal{K}(\mathbf{M}, K), G^*)]$ plotted against γ for $\mathbf{M} = \hat{\mathbf{\Sigma}}, \hat{\mathbf{\Sigma}} - \hat{\mathbf{\Gamma}}, P_F(\mathbf{V}_T)$; the expectation is both with respect to the generating model and the randomness of \mathcal{K} . For each level of γ , 50 random instances were generated, and because \mathcal{K} is a random algorithm, it is run multiple times on each instance. The average across both instances and runs of \mathcal{K} is reported. One trend of particular importance is that as the level of noise increases, the expected exact recovery rate for either of the alternative candidate heuristics goes to zero.

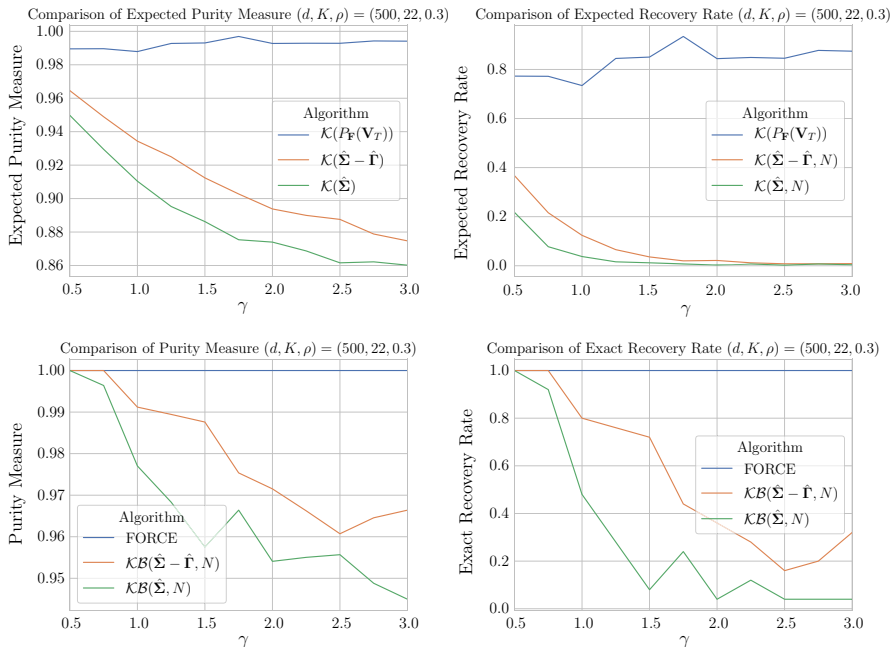


Fig. 3 Comparison of FORCE with heuristic methods, demonstrating that as noise increases heuristics alone cannot provide high quality clusterings

A natural follow-up question is whether or not, despite the *expected* recovery rate going to zero as γ increases, if we run \mathcal{K} many times using $\mathbf{M} = \hat{\mathbf{S}}$, $\hat{\mathbf{S}} - \hat{\mathbf{F}}$ and select the best clustering found, can we do just as well as FORCE? If yes, then running FORCE (and indeed solving the P–W SDP relaxation in general) offers little benefit over running \mathcal{K} many times and then attempting to certify the best clustering found. To answer this question, we denote by $\mathcal{KB}(\mathbf{M}, K, N)$ the algorithm which runs \mathcal{K} , defined above, N times and returns the best clustering found in terms of SDP objective value. We compare this to the output of FORCE and choose N to be the maximum of 100 and the number of times FORCE calls a clustering algorithm as a sub-routine on that problem instance. The results in terms of $\mathbb{E}[d_i(\mathcal{KB}(\mathbf{M}, K, N), G^*)]$ are plotted versus gamma in row two of Fig. 3; as before, 50 random instances were generated for each level of γ . Examining the plots we can conclude that solving the SDP not only improves the percentage of points clustered correctly on average, but that it is essential to achieving exact recovery. Using heuristic methods alone cannot achieve the same performance as FORCE or other algorithms that leverage the P–W SDP relaxation.

7 Conclusion

Motivated by the variable clustering problem, we proposed a new algorithm, FORCE, to solve the P–W SDP which has strong statistical properties in many clustering regimes. FORCE consists of a primal first-order method based on Renegar’s method

[17] and a novel dual certificate construction. We show that for G -Latent models satisfying a minimal cluster separation condition, FORCE is guaranteed with high probability to both recover the true latent structure G^* and provide a certificate of having done so. We extended our results to a variant of the P–W SDP where K is not known a priori.

One interesting consequence of our certificate existence theorems, Theorems 2 and 4, is that they show for G -Latent models, the SDPs (2) and (22) are tight with high probability for $\Delta(C^*)$ sufficiently large. Indeed we recover nearly the same minimal cluster separation rate as Bunea et al. [1], with the distinction that our proof is constructive in nature.

Our numerical studies clearly indicate the success of FORCE in the variable clustering setting. In our simulation studies, only one other method, ADMM, was able to scale to high dimensions, and it often did not converge in high noise designs. Our studies also verified that solving the P–W SDP was essential to achieve high quality clusterings as noise increased (see Fig. 3). In future work it would be of interest to study the properties of the FORCE dual certificate under other generating distributions for variable and data clustering. The FORCE algorithm is available in the R package GFORCE on CRAN.

Appendix A: Proofs omitted in Sect. 3.2

First we have a lemma regarding the concentration of the noise terms \mathbf{E} about their mean. Sometimes rather than state these concentration results in terms of d , we state them in terms of $t \geq d$ to allow for more precise control of constants in our main theorems. We let \mathcal{E} denote the event that $\|\hat{\mathbf{\Gamma}} - \mathbf{\Gamma}^*\|_\infty \leq p_1 \|\mathbf{\Gamma}^*\|_{\max} \sqrt{\frac{\log d}{n}}$.

Lemma 9 *Under the notation and assumptions from previous sections, if $t \geq d$ then*

$$\left| \sum_{j=1}^n \mathbf{1}^T \mathbf{E}_{G_i^*}^j \mathbf{E}_{G_i^*}^{jT} \mathbf{1} - \mathbf{1}^T \mathbf{\Gamma}_{G_i^*, G_i^*}^* \mathbf{1} \right| \leq c_0 \|\mathbf{\Gamma}^*\|_\infty \sqrt{|G_i^*|^2 n \log t},$$

with probability at least $1 - \frac{2}{t}$, where $c_0 = c'(1 + \sqrt{p_0})$ is a constant that depends only on p_0 and the absolute constant c' from Proposition 11. Similarly with probability at least $1 - \frac{2}{t}$, for $a \in G_i^*$,

$$\left| \sum_{j=1}^n \mathbf{1}^T \mathbf{E}_{G_i^*}^j \mathbf{E}_a^j - \gamma_a^* \right| \leq c_0 \|\mathbf{\Gamma}^*\|_\infty \sqrt{|G_i^*| n \log t},$$

Proof To obtain the result, we observe that

$$\sum_{j=1}^n \mathbf{1}^T \mathbf{E}_{G_i^*}^j \mathbf{E}_{G_i^*}^{jT} \mathbf{1} - \mathbf{1}^T \mathbf{\Gamma}_{G_i^*, G_i^*}^* \mathbf{1}$$

is a quadratic form of a $n|G_i^*|$ -dimensional Gaussian random vector with independent entries. In particular, if we define \mathbf{M} to be block diagonal with the i th $n \times n$ diagonal block as $(\mathbf{\Gamma}_{G_i^*, G_i^*}^*)^{1/2} \mathbf{1}\mathbf{1}^T (\mathbf{\Gamma}_{G_i^*, G_i^*}^*)^{1/2}$, then we can apply Corollary 1 with matrix \mathbf{M} . Because $\|\mathbf{M}\|_2 \leq \|\mathbf{\Gamma}^*\|_\infty |G_i^*|$ and $\|\mathbf{M}\|_F \leq \|\mathbf{\Gamma}^*\|_\infty |G_i^*| \sqrt{n}$, applying the corollary gives

$$\left| \sum_{j=1}^n \mathbf{1}^T \mathbf{E}_{G_i^*, G_i^*}^j \mathbf{E}_{G_i^*, G_i^*}^{jT} \mathbf{1} - \mathbf{1}^T \mathbf{\Gamma}_{G_i^*, G_i^*}^* \mathbf{1} \right| \leq c' \|\mathbf{\Gamma}^*\|_\infty \left(\sqrt{|G_i^*|^2 n \log t} + |G_i^*| \log t \right),$$

with probability at least $1 - \frac{2}{t}$. Using the assumption $\log d \leq p_0 n$ gives the desired result. The proof of the second statement follows similarly, taking instead the diagonal blocks of \mathbf{M} as $(\mathbf{\Gamma}_{G_i^*, G_i^*}^*)^{1/2} \mathbf{1}_a \mathbf{1}_a^T (\mathbf{\Gamma}_{G_i^*, G_i^*}^*)^{1/2}$, giving $\|\mathbf{M}\|_2 \leq \|\mathbf{\Gamma}^*\|_\infty \sqrt{|G_i^*|}$ and $\|\mathbf{M}\|_F \leq \|\mathbf{\Gamma}^*\|_\infty \sqrt{n|G_i^*|}$. \square

Proof of Lemma 6

Step 1 For notation, c_i will be used to denote absolute constants. The first step is to decompose $\mathbf{Q}_i^\perp(X)$. Recall that under the G-Latent model, $\mathbf{D} = -\widehat{\mathbf{\Sigma}} + \widehat{\mathbf{\Gamma}}$. Substituting that into the expression for $\mathbf{Q}_i^\perp(X)$ gives

$$\begin{aligned} \mathbf{Q}_i^\perp(X) = & \underbrace{-\frac{1}{|G_i^*|^2} \left(\mathbf{1}^T \widehat{\mathbf{\Sigma}}_{G_i^*, G_i^*} \mathbf{1} \right) \mathbf{1}\mathbf{1}^T + \frac{1}{|G_i^*|} \left(\mathbf{1}\mathbf{1}^T \widehat{\mathbf{\Sigma}}_{G_i^*, G_i^*} + \widehat{\mathbf{\Sigma}}_{G_i^*, G_i^*} \mathbf{1}\mathbf{1}^T \right) - \widehat{\mathbf{\Sigma}}_{G_i^*, G_i^*}}_{(i)} \\ & + \underbrace{\frac{1}{|G_i^*|^2} \left(\mathbf{1}^T \widehat{\mathbf{\Gamma}}_{G_i^*, G_i^*} \mathbf{1} \right) \mathbf{1}\mathbf{1}^T - \frac{1}{|G_i^*|} \left(\mathbf{1}\mathbf{1}^T \widehat{\mathbf{\Gamma}}_{G_i^*, G_i^*} + \widehat{\mathbf{\Gamma}}_{G_i^*, G_i^*} \mathbf{1}\mathbf{1}^T \right) + \widehat{\mathbf{\Gamma}}_{G_i^*, G_i^*}}_{(ii)}. \end{aligned}$$

For (i), we recall that by the definition of the G-Latent model that

$$\widehat{\mathbf{\Sigma}}_{G_i^*, G_i^*} = \frac{1}{n} \sum_{j=1}^n \mathbf{X}_{G_i^*}^j \mathbf{X}_{G_i^*}^{jT} = \sum_{j=1}^n (\mathbf{Z}_i^j + \mathbf{E}_{G_i^*}^j)(\mathbf{Z}_i^j + \mathbf{E}_{G_i^*}^j)^T.$$

Plugging this into (i) and simplifying gives us that

$$(i) = \frac{1}{n} \sum_{j=1}^n \left(-\frac{\mathbf{1}^T \mathbf{E}_{G_i^*}^j \mathbf{E}_{G_i^*}^{jT} \mathbf{1}}{|G_i^*|^2} \mathbf{1}\mathbf{1}^T + \frac{\mathbf{1}^T \mathbf{E}_{G_i^*}^j}{|G_i^*|} \left(\mathbf{1} \mathbf{E}_{G_i^*}^{jT} + \mathbf{E}_{G_i^*}^j \mathbf{1}^T \right) - \mathbf{E}_{G_i^*}^j \mathbf{E}_{G_i^*}^{jT} \right).$$

Now we see that, again, the expression for $\mathbf{Q}_i^\perp(X)$ has eight terms. We first show that each concentrates to its mean at the desired rate, and then use the triangle inequality to obtain the final result. Fortunately, we can subtract the mean for each of the 8 terms

to the expression for $\mathbf{Q}_i^\perp(X)$ as the means for (i) are offset by the means for (ii). To give the new decomposition of $\mathbf{Q}_i^\perp(X)$ explicitly,

$$\begin{aligned} \mathbf{Q}_i^\perp(X) = & - \underbrace{\sum_{j=1}^n \frac{\mathbf{1}^T \mathbf{E}_{G_i^*}^j \mathbf{E}_{G_i^*}^{jT} \mathbf{1}}{n|G_i^*|^2} \mathbf{1} \mathbf{1}^T}_{(i).a} + \underbrace{\sum_{j=1}^n \frac{\mathbf{1}^T \mathbf{E}_{G_i^*}^j}{n|G_i^*|} \mathbf{1} \mathbf{E}_{G_i^*}^{jT}}_{(i).b} + \underbrace{\sum_{j=1}^n \frac{\mathbf{1}^T \mathbf{E}_{G_i^*}^j}{n|G_i^*|} \mathbf{E}_{G_i^*}^j \mathbf{1}^T}_{(i).c} - \underbrace{\frac{1}{n} \sum_{j=1}^n \mathbf{E}_{G_i^*}^j \mathbf{E}_{G_i^*}^{jT}}_{(i).d} \\ & + \underbrace{\frac{1}{|G_i^*|^2} \left(\mathbf{1}^T \widehat{\mathbf{F}}_{G_i^*, G_i^*} \mathbf{1} \right) \mathbf{1} \mathbf{1}^T}_{(ii).a} - \underbrace{\frac{1}{|G_i^*|} \mathbf{1} \mathbf{1}^T \widehat{\mathbf{F}}_{G_i^*, G_i^*}}_{(ii).b} + \underbrace{\frac{1}{|G_i^*|} \widehat{\mathbf{F}}_{G_i^*, G_i^*} \mathbf{1} \mathbf{1}^T}_{(ii).c} + \underbrace{\widehat{\mathbf{F}}_{G_i^*, G_i^*}}_{(ii).d}. \quad (30) \end{aligned}$$

Step 2 For the term (i).a, we can directly apply Lemma 9. Doing so, it follows immediately that with probability at least $1 - \frac{2}{t}$

$$\left\| \sum_{j=1}^n \frac{\mathbf{1}^T \mathbf{E}_{G_i^*}^j \mathbf{E}_{G_i^*}^{jT} \mathbf{1}}{n|G_i^*|^2} \mathbf{1} \mathbf{1}^T - \frac{1}{|G_i^*|^2} \left(\mathbf{1}^T \mathbf{F}_{G_i^*, G_i^*}^* \mathbf{1} \right) \mathbf{1} \mathbf{1}^T \right\|_2 \leq c_0 \|\mathbf{F}^*\|_\infty \sqrt{\frac{\log t}{n}}.$$

For the term (i).c (and so by symmetry (i).b), we observe that has the form $\mathbf{u} \mathbf{v}^T$ and that $\|\mathbf{u} \mathbf{v}^T\|_2 = \|\mathbf{u}\|_2 \|\mathbf{v}\|_2$. Therefore, we can apply Lemma 9 and obtain that with probability at least $1 - 2|G_i^*|/t^2$,

$$\left\| \sum_{j=1}^n \frac{\mathbf{1}^T \mathbf{E}_{G_i^*}^j}{n|G_i^*|} \mathbf{E}_{G_i^*}^j \mathbf{1}^T - \frac{1}{|G_i^*|} \mathbf{1} \mathbf{1}^T \mathbf{F}_{G_i^*, G_i^*}^* \right\|_2 \leq c_0 \|\mathbf{F}^*\|_\infty \sqrt{\frac{2 \log t}{n}}.$$

Step 3 Now we control the term (i).d, the sample covariance matrix of the errors. We can directly apply Corollary 3 to obtain that with probability at least $1 - 2/t$

$$\begin{aligned} & \left\| \frac{1}{n} \sum_{j=1}^n \mathbf{E}_{G_i^*}^j \mathbf{E}_{G_i^*}^{jT} - \mathbf{F}_{G_i^*, G_i^*}^* \right\|_2 \\ & \leq \|\mathbf{F}^*\|_\infty \left(\frac{|G_i^*|}{n} + 2 \frac{\sqrt{2|G_i^*| \log t}}{n} + 2 \sqrt{\frac{|G_i^*|}{n}} + (2 + \sqrt{p_0}) \sqrt{\frac{2 \log t}{n}} \right) \\ & \leq \|\mathbf{F}^*\|_\infty \left(\frac{d}{n} + (2 + 2\sqrt{2p_0}) \sqrt{\frac{d}{n}} + (2 + \sqrt{p_0}) \sqrt{\frac{2 \log t}{n}} \right). \end{aligned}$$

Step 4 For the terms in (ii), consider first (ii).a. We see that

$$\left\| \left(\mathbf{1}^T \widehat{\mathbf{F}}_{G_i^*, G_i^*} \mathbf{1} \right) \mathbf{1} \mathbf{1}^T - \left(\mathbf{1}^T \mathbf{F}_{G_i^*, G_i^*}^* \mathbf{1} \right) \mathbf{1} \mathbf{1}^T \right\|_{\max} \leq |G_i^*| \|\widehat{\mathbf{F}}_{G_i^*, G_i^*} - \mathbf{F}_{G_i^*, G_i^*}^*\|_\infty$$

Conditional on event \mathcal{E} ,

$$\left\| \frac{1}{|G_i^*|^2} \left(\mathbf{1}^T \widehat{\mathbf{F}}_{G_i^*, G_i^*} \mathbf{1} \right) \mathbf{1}^T - \frac{1}{|G_i^*|^2} \left(\mathbf{1}^T \mathbf{F}_{G_i^*, G_i^*}^* \mathbf{1} \right) \mathbf{1}^T \right\|_{\max} \leq \frac{p_1 \|\mathbf{F}^*\|_{\infty}}{|G_i^*|} \sqrt{\frac{\log d}{n}}.$$

Because the matrices above are a multiple of $\mathbf{1}\mathbf{1}^T$, it follows that

$$\left\| \frac{1}{|G_i^*|^2} \left(\mathbf{1}^T \widehat{\mathbf{F}}_{G_i^*, G_i^*} \mathbf{1} \right) \mathbf{1}^T - \frac{1}{|G_i^*|^2} \left(\mathbf{1}^T \mathbf{F}_{G_i^*, G_i^*}^* \mathbf{1} \right) \mathbf{1}^T \right\|_2 \leq p_1 \|\mathbf{F}^*\|_{\infty} \sqrt{\frac{\log d}{n}}.$$

Next for (ii).b (and (ii).c by symmetry), we can see that

$$\left\| \frac{1}{|G_i^*|} \mathbf{1}^T \widehat{\mathbf{F}}_{G_i^*, G_i^*} - \frac{1}{|G_i^*|} \mathbf{1}^T \mathbf{F}_{G_i^*, G_i^*}^* \right\|_2 = \frac{1}{|G_i^*|} \left\| \mathbf{1}^T \left(\widehat{\mathbf{F}}_{G_i^*, G_i^*} - \mathbf{F}_{G_i^*, G_i^*}^* \right) \right\|_2. \quad (31)$$

Because $\widehat{\mathbf{F}}$ and \mathbf{F}^* are diagonal, we can use event \mathcal{E} and the fact that for matrices of the form $\mathbf{u}\mathbf{v}^T$, $\|\mathbf{u}\mathbf{v}^T\|_2 = \|\mathbf{u}\|_2 \|\mathbf{v}\|_2$, to obtain

$$\left\| \frac{1}{|G_i^*|} \mathbf{1}^T \widehat{\mathbf{F}}_{G_i^*, G_i^*} - \frac{1}{|G_i^*|} \mathbf{1}^T \mathbf{F}_{G_i^*, G_i^*}^* \right\|_2 \leq p_1 \|\mathbf{F}^*\|_{\infty} \sqrt{\frac{\log d}{n}}$$

The same result is immediate for (ii).a by (4). Therefore by combining the above, applying the triangle inequality to (30), using that \mathcal{E} occurs with probability at least $1 - p_2/d^2$, and choosing $t = d^2$, we find that with probability at least $1 - \frac{c_2}{d^2}$

$$\|\mathbf{Q}_i^\perp(\mathbf{X})\|_2 \leq c_1 \|\mathbf{F}^*\|_{\infty} \left(\frac{d}{n} + \sqrt{\frac{d}{n}} + \sqrt{\frac{\log d}{n}} \right),$$

concluding the proof. \square

Proof of Lemma 7

Under the G-Latent model,

$$y'_{a,b}(\mathbf{X}, y_T) = - \underbrace{\widehat{\Sigma}_{a,b}}_{(i)} + \underbrace{y_a(\mathbf{X}, y_T)}_{(ii)} + \underbrace{y_b(\mathbf{X}, y_T)}_{(iii)}$$

Above, we saw that

$$y_a(\mathbf{X}, y_T) = \frac{1}{2|G_i^*|^2} \mathbf{1}^T \mathbf{D}_{G_i^*, G_i^*} \mathbf{1} - \frac{1}{|G_i^*|} \mathbf{D}_{a, G_i^*} \mathbf{1} - \frac{1}{2|G_i^*|} y_T,$$

and likewise for y_b . Below we denote by $\sigma_1 = \max_i C_{i,i}^*$ and $\sigma_2 = \max\{\max_i C_{i,i}^*, \|\mathbf{\Gamma}^*\|_\infty\}$. Following the same decomposition as in Lemma 6, we get that

$$\begin{aligned} y_a(X, y_T) &= -\frac{1}{2|G_i^*|^2} \mathbf{1}^T \widehat{\Sigma}_{G_i^*, G_i^*} \mathbf{1} + \frac{1}{2|G_i^*|^2} \mathbf{1}^T \widehat{\Gamma}_{G_i^*, G_i^*} \mathbf{1} + \frac{1}{|G_i^*|} \widehat{\Sigma}_{a, G_i^*} \mathbf{1} - \widehat{\Gamma}_{a, a} - \frac{1}{2|G_i^*|} y_T \\ &= \underbrace{\frac{1}{n} \sum_{l=1}^n \frac{1}{2} (Z_i^l)^2}_{(ii).a} - \underbrace{\frac{1}{2n|G_i^*|^2} \sum_{l=1}^n (\mathbf{1}^T E_{G_i^*}^l)^2}_{(ii).b} + \underbrace{\frac{1}{n|G_i^*|} \sum_{l=1}^n E_a^l \mathbf{1}^T E_{G_i^*}^l}_{(ii).c} + \underbrace{\frac{1}{n} \sum_{l=1}^n E_a^l Z_i^l}_{(ii).d} \\ &\quad + \underbrace{\frac{1}{2|G_i^*|^2} \mathbf{1}^T \widehat{\Gamma}_{G_i^*, G_i^*} \mathbf{1}}_{(ii).e} - \underbrace{\frac{1}{|G_i^*|} \widehat{\Gamma}_{a, a}}_{(ii).f} - \frac{1}{2|G_i^*|} y_T. \end{aligned}$$

As in the proof of Lemma 6, the means of (ii).b and (ii).c offset the means of (ii).e and (ii).f. To control terms (ii).b and (ii).c, by Lemma 9 with probability at least $1 - 1/t$,

$$\frac{1}{2n|G_i^*|^2} \sum_{j=1}^n \left(\mathbf{1}^T E_{G_i^*}^j E_{G_i^*}^{jT} \mathbf{1} - \mathbf{1}^T \mathbf{\Gamma}_{G_i^*, G_i^*}^* \mathbf{1} \right) \leq \frac{c_0 \|\mathbf{\Gamma}^*\|_\infty}{2} \sqrt{\frac{\log t}{n|G_i^*|^2}}.$$

Likewise, by Lemma 9,

$$\frac{1}{n|G_i^*|} \sum_{i=1}^n \left(E_a E_{G_i^*}^{jT} \mathbf{1} - \gamma_a^* \right) \geq -c_0 \|\mathbf{\Gamma}^*\|_\infty \sqrt{\frac{\log t}{n|G_i^*|}},$$

with probability at least $1 - 1/t$. Conditional on event \mathcal{E} , (4) shows that

$$\begin{aligned} \frac{1}{2|G_i^*|^2} \left(\mathbf{1}^T \widehat{\Gamma}_{G_i^*, G_i^*} \mathbf{1} - \mathbf{1}^T \mathbf{\Gamma}_{G_i^*, G_i^*}^* \mathbf{1} \right) &\geq -p_1 \|\mathbf{\Gamma}^*\|_\infty \sqrt{\frac{\log d}{n|G_i^*|}}, \\ \frac{1}{|G_i^*|} \left(\widehat{\Gamma}_{a, a} - \Gamma_{a, a}^* \right) &\leq p_1 \|\mathbf{\Gamma}^*\|_\infty \sqrt{\frac{\log d}{n|G_i^*|}}. \end{aligned}$$

Lastly, if we denote by $\sigma_1 = \max_i C_{i,i}^*$, term (ii).d can be bounded by using Corollary 1, which gives that

$$\frac{1}{n} \sum_{l=1}^n E_a^l Z_i^l \geq -c_0 \|\mathbf{\Gamma}^*\|_\infty^{1/2} \sigma_1^{1/2} \sqrt{\frac{\log t}{n}}, \quad (32)$$

with probability at least $1 - 1/t$. The same results can be obtained for y_b . For the terms in (i), we expand as before:

$$\widehat{\Sigma}_{a,b} = \underbrace{\frac{1}{n} \sum_{l=1}^n Z_i^l Z_j^l}_{(i).a} + \underbrace{\frac{1}{n} \sum_{l=1}^n E_a^l Z_j^l}_{(i).b} + \underbrace{\frac{1}{n} \sum_{l=1}^n E_b^l Z_i^l}_{(i).c} + \underbrace{\frac{1}{n} \sum_{l=1}^n E_a^l E_b^l}_{(i).d}.$$

Terms (i).b and (i).c can be bounded in the same way as (32). Term (i).d can be bounded by Corollary 1, giving that

$$\frac{1}{n} \sum_{l=1}^n E_a^l E_b^l \geq -c_0 \|\mathbf{\Gamma}^*\|_\infty \sqrt{\frac{\log t}{n}},$$

with probability at least $1 - 1/t$. All that remains is to bound the terms (i).a, (ii).a and (iii).a. Fortunately, these correspond to the population quantity $\Delta \mathbf{C}^*$. Observing that this is just a quadratic form of $2n$ -dimensional Gaussian vector, we can applying Lemma 9. Doing so gives that

$$\frac{1}{2n} \left(\sum_{l=1}^n (Z_i^l)^2 + \sum_{l=1}^n (Z_j^l)^2 - 2 \sum_{l=1}^n Z_i^l Z_j^l \right) \geq \frac{1}{2} (C_{i,i}^* + C_{j,j}^* - C_{i,j}^*) - 2c_0 \sigma_1 \sqrt{\frac{\log t}{n}}$$

with probability at least $1 - 1/t$. Combining all the bounds for (i)-(iii), using that \mathcal{E} occurs with probability at least $1 - p_2/d^3$, and selecting $t = d^3$, we can see that, with probability at least $1 - c_1/d^3$

$$\begin{aligned} y'_{a,b} &\geq \frac{1}{2} (C_{i,i}^* + C_{j,j}^* - 2C_{i,j}^*) - \frac{1}{2|G_i^*|} y_T - \frac{1}{2|G_j^*|} y_T - c_1 \|\mathbf{\Gamma}^*\|_\infty \sqrt{\frac{\log d}{n|G_i^*|}} - c_2 \sigma \sqrt{\frac{\log d}{n}} \\ &\geq \frac{1}{2} \Delta(\mathbf{C}^*) - \frac{1}{2|G_i^*|} y_T - \frac{1}{2|G_j^*|} y_T - c_1 \|\mathbf{\Gamma}^*\|_\infty \sqrt{\frac{\log d}{n|G_i^*|}} - c_2 \sigma \sqrt{\frac{\log d}{n}}. \end{aligned}$$

□

Appendix B: Some technical lemmas

Lemma 10 Let \mathbf{M} be a $d \times d$ real, symmetric matrix of the form

$$\mathbf{M} = a\mathbf{I} + b\mathbf{1}\mathbf{1}^T.$$

where $a, b \in \mathbb{R}$ then \mathbf{M} has eigenvalues $a+b$ with multiplicity 1 and a with multiplicity $d - 1$. If $a, b > 0$, then \mathbf{M} also has the property that

$$\begin{aligned} \mathbf{M}^{1/2} &= \sqrt{a}\mathbf{I} + \frac{\sqrt{a+db} - \sqrt{a}}{d} \mathbf{1}\mathbf{1}^T, \\ \mathbf{M}^{-1} &= \frac{1}{a}\mathbf{I} - \frac{b}{a^2 + abd} \mathbf{1}\mathbf{1}^T, \\ \mathbf{M}^{-1/2} &= \frac{1}{\sqrt{a}}\mathbf{I} - \frac{\sqrt{a+db} - \sqrt{a}}{d\sqrt{a^2 + dab}} \mathbf{1}\mathbf{1}^T. \end{aligned}$$

Proof Using the Sherman-Morrison formula, a matrix of the form $\mathbf{M} = a\mathbf{I} + b\mathbf{1}\mathbf{1}^T$, where $a, b > 0$ has the inverse

$$\mathbf{M}^{-1} = \frac{1}{a}\mathbf{I} - \frac{b}{a^2 + abd}\mathbf{1}\mathbf{1}^T.$$

Because $\mathbf{M} \succ 0$, all eigenvalues are strictly positive and denote by λ_i and q_i the eigenvalues and corresponding eigenvectors. Without loss of generality, let q_i be orthonormal. Then we can write $\mathbf{M} = \sum_i \lambda_i q_i q_i^T$. By the form of \mathbf{M} , clearly $\frac{1}{\sqrt{d}}\mathbf{1}$ is always an eigenvector of \mathbf{M} with eigenvalue $a + db$, so we can take $q_1 = \frac{1}{\sqrt{d}}\mathbf{1}$ and $\lambda_1 = 1$. The remaining q_i span $(\mathbf{1}\mathbf{1}^T)^\perp$ and have corresponding eigenvalues $\lambda_i = a$. Therefore,

$$\mathbf{M}^{1/2} = \frac{\sqrt{a+db}}{\sqrt{d}}\mathbf{1}\mathbf{1}^T + \sum_{i=2}^d \sqrt{a} q_i q_i^T.$$

Because this eigen-decomposition is unique, the above gives

$$\mathbf{M}^{1/2} = \sqrt{a}\mathbf{I} + \frac{\sqrt{a+db} - \sqrt{a}}{d}\mathbf{1}\mathbf{1}^T.$$

Using the expression for \mathbf{M}^{-1} given above, it follows that

$$\mathbf{M}^{-1/2} = \frac{1}{\sqrt{a}}\mathbf{I} - \frac{\sqrt{a+db} - \sqrt{a}}{d\sqrt{a^2 + dab}}\mathbf{1}\mathbf{1}^T.$$

□

The following result for quadratic forms of standard multivariate Gaussian random variables can be found in many forms in the literature (for example, Rudelson and Vershynin [34]).

Lemma 11 (Hanson-Wright inequality for Gaussian random variables) *Let $\mathbf{X} \sim N(0, \mathbf{I})$ be a d -dimensional random vector and let \mathbf{A} be a $d \times d$ matrix in $\mathbb{R}^{d \times d}$. Then*

$$\mathbb{P}\left(|\mathbf{X}^T \mathbf{A} \mathbf{X} - \mathbb{E}[\mathbf{X}^T \mathbf{A} \mathbf{X}]| \geq t\right) \leq 2 \exp\left(-c \min\left\{\frac{t^2}{\|\mathbf{A}\|_F^2}, \frac{t}{\|\mathbf{A}\|_2}\right\}\right),$$

for some absolute constant c .

In particular, the following corollary is useful.

Corollary 1 *Let $\mathbf{X} \sim N(0, \mathbf{I})$ be a d -dimensional random vector and let \mathbf{A} be a $d \times d$ matrix in $\mathbb{R}^{d \times d}$. Then*

$$\mathbb{P}\left(|\mathbf{X}^T \mathbf{A} \mathbf{X} - \mathbb{E}[\mathbf{X}^T \mathbf{A} \mathbf{X}]| \geq \|\mathbf{A}\|_F \sqrt{t} + \|\mathbf{A}\|_2 t\right) \leq 2 \exp(-ct),$$

for some absolute constant c . Equivalently,

$$|\mathbf{X}^T \mathbf{A} \mathbf{X} - \mathbb{E}[\mathbf{X}^T \mathbf{A} \mathbf{X}]| \leq c' \left(\|\mathbf{A}\|_F \sqrt{\log t} + \|\mathbf{A}\|_2 \log t \right)$$

with probability at least $1 - 2/t$ for some absolute constant c' .

Below we are concerned with the rate of concentration in the spectral norm of a sample covariance matrix to its mean: $\|\widehat{\Sigma} - \Sigma^*\|_2$. If we write $\widehat{\Sigma} = \frac{1}{n} \mathbf{X}^T \mathbf{X}$, where \mathbf{X} refers to the $n \times d$ matrix in which the rows are the observations \mathbf{X}_i , we see how such a result is directly applicable to the problem at hand. We repeat the statement of Gordon's Theorem given in Vershynin [35] below as Proposition 1. We use the notation from Vershynin [35] of s_{\min} and s_{\max} to denote the smallest and largest singular values, respectively.

Proposition 1 *Let \mathbf{X} be an $n \times d$ matrix whose entries are independent standard normal random variables. Then*

$$\sqrt{n} - \sqrt{d} \leq \mathbb{E}[s_{\min}(\mathbf{X})] \leq \mathbb{E}[s_{\max}(\mathbf{X})] \leq \sqrt{n} + \sqrt{d}$$

Using the result on sub-Gaussian concentration of a Lipschitz function of independent random variables, we immediately obtain the following corollary (also given in Vershynin [35]).

Corollary 2 *Let \mathbf{X} be an $n \times d$ matrix whose entries are independent standard normal random variables, then for every $t \geq 0$*

$$\sqrt{n} - \sqrt{d} - t \leq s_{\min}(\mathbf{X}) \leq s_{\max}(\mathbf{X}) \leq \sqrt{n} + \sqrt{d} + t$$

with probability at least $1 - 2 \exp(-t^2/2)$.

Proof Observing that the functions s_{\min} and s_{\max} are 1-Lipschitz and using the sub-Gaussian tail bound, the result is immediate from the above. \square

Corollary 3 *Let \mathbf{X}_i , for $i = 1, \dots, n$, be a d -dimensional random vector sampled from $N(0, \Sigma)$. Denoting $\widehat{\Sigma} := n^{-1} \sum_{i=1}^n \mathbf{X}_i \mathbf{X}_i^T$, we have that*

$$\begin{aligned} \lambda_{\min}(\widehat{\Sigma} - \Sigma) &\geq \lambda_{\min}(\Sigma) \left(\frac{d}{n} + \frac{2t\sqrt{d}}{n} + \frac{t^2}{n} - \frac{2(\sqrt{d} + t)}{\sqrt{n}} \right), \\ \lambda_{\max}(\widehat{\Sigma} - \Sigma) &\leq \lambda_{\max}(\Sigma) \left(\frac{d}{n} + \frac{2t\sqrt{d}}{n} + \frac{t^2}{n} + \frac{2(\sqrt{d} + t)}{\sqrt{n}} \right), \end{aligned}$$

with probability at least $1 - 2 \exp(-t^2/2)$.

Proof This follows directly from Corollary 2. \square

Appendix C: Numerical results for unknown K

In this section we consider the exact same setup as in Sect. 6, but now we solve (22) using FORCE. For K unknown we do not have a readily available baseline comparison in high dimensions—prior work considers an ADMM based approach for (2), but not (22)—so we present only the results from FORCE and FORCE-P.

The main differences between the results displayed in Tables 3 and 4 and those in Sect. 6 are that (1) SDPNAL reached its iteration limit in some cases, leading to a higher average error than for K known, and (2) for the setting $(d, k, \rho, \gamma) = (500, 100, 0.3, 3.0)$ the SNR was too low and FORCE did not converge. Otherwise, the results shown below, are very similar to those when K is known, and we refer the reader to the discussion in Sect. 6. The fact that we do not observe any significant difference in empirical performance between K fixed and K unknown—indeed K unknown appears to take longer in practice as intuition suggests—indicates that Theorem 1 either may not be tight or instances on which it achieves the worst-case bound are encountered with low probability.

Table 3 Benchmark results for low dimensional designs comparing FORCE and FORCE-P with MOSEK and SDPNAL+ with unknown K

| (d, k, ρ, γ) | $(50, 5, 0.3, 0.3)$ | | $(50, 5, 0.3, 1.0)$ | |
|------------------------|-----------------------|-----------------------|------------------------|-----------------------|
| Algorithm | Rel. Err. | Time (s) | Rel. Err. | Time (s) |
| MOSEK | 1.14×10^{-8} | 4.31×10^{-1} | 3.88×10^{-9} | 4.68×10^{-1} |
| SDPNAL+ | 3.13×10^{-6} | 1.61×10^0 | 3.69×10^{-6} | 1.50×10^0 |
| FORCE | 0.00×10^0 | 3.17×10^{-4} | 1.13×10^{-8} | 1.80×10^{-1} |
| FORCE-P | 7.72×10^{-3} | 1.67×10^{-1} | 1.57×10^{-2} | 2.70×10^{-1} |
| (d, k, ρ, γ) | $(85, 9, 0.3, 0.3)$ | | $(85, 9, 0.3, 1.0)$ | |
| Algorithm | Rel. Err. | Time (s) | Rel. Err. | Time (s) |
| MOSEK | 3.53×10^{-8} | 4.33×10^0 | 8.29×10^{-9} | 5.32×10^0 |
| SDPNAL+ | 3.18×10^{-3} | 7.91×10^0 | 3.76×10^{-2} | 7.13×10^0 |
| FORCE | 0.00×10^0 | 9.13×10^{-4} | 1.94×10^{-10} | 8.05×10^{-2} |
| FORCE-P | 7.44×10^{-3} | 4.59×10^{-1} | 1.18×10^{-2} | 6.05×10^{-1} |
| (d, k, ρ, γ) | $(120, 12, 0.3, 0.3)$ | | $(120, 12, 0.3, 1.0)$ | |
| Algorithm | Rel. Err. | Time (s) | Rel. Err. | Time (s) |
| MOSEK | 3.49×10^{-8} | 2.25×10^1 | 1.13×10^{-8} | 2.56×10^1 |
| SDPNAL+ | 3.09×10^{-2} | 1.56×10^1 | 3.16×10^{-2} | 1.49×10^1 |
| FORCE | 0.00×10^0 | 1.72×10^{-3} | 0.00×10^0 | 1.67×10^{-3} |
| FORCE-P | 6.92×10^{-3} | 9.61×10^{-1} | 1.20×10^{-2} | 1.01×10^0 |

Table 4 Benchmark results for high dimensional designs comparing FORCE and FORCE-P for K unknown

| Alg. | (d, k, ρ) | $(\rho, \gamma) = (0.3, 1.0)$ | | $(\rho, \gamma) = (0.3, 3.0)$ | |
|---------|----------------|-------------------------------|-----------------------|-------------------------------|--------------------|
| | | Rel. Err. | Time (s) | Rel. Err. | Time (s) |
| FORCE | (500, 9) | 1.73×10^{-16} | 3.60×10^{-2} | 1.82×10^{-16} | 1.00×10^0 |
| FORCE-P | | 8.97×10^{-3} | 2.03×10^1 | 2.82×10^{-2} | 4.89×10^1 |
| FORCE | (500, 22) | 7.76×10^{-17} | 5.56×10^{-2} | 6.74×10^{-17} | 2.39×10^0 |
| FORCE-P | | 9.33×10^{-3} | 2.82×10^1 | 2.74×10^{-2} | 4.69×10^1 |
| FORCE | (500, 50) | 1.27×10^{-16} | 3.79×10^{-2} | 6.08×10^{-3} | 2.15×10^1 |
| FORCE-P | | 9.56×10^{-3} | 1.94×10^1 | 8.17×10^{-2} | 4.20×10^1 |
| FORCE | (500, 100) | 1.73×10^{-16} | 3.38×10^{-2} | N/A | N/A |
| FORCE-P | | 1.08×10^{-2} | 1.66×10^1 | N/A | N/A |

References

- Bunea, F., Giraud, C., Royer, M., Verzelen, N.: PECOK: a convex optimization approach to variable clustering. [arXiv:1606.05100](https://arxiv.org/abs/1606.05100) (2016)
- Dasgupta, S.: The hardness of k-means clustering. University of California, San Diego, Tech. rep. (2008)
- Mahajan, M., Nimbhorkar, P., Varadarajan, K.: The planar k-means problem is NP-hard. *Theor. Comput. Sci.* **442**, 13–21 (2012). <https://doi.org/10.1016/j.tcs.2010.05.034>
- Lloyd, S.P.: Least squares quantization in PCM. *IEEE Trans. Inf. Theory* **28**(2), 129–137 (1982). <https://doi.org/10.1109/TIT.1982.1056489>
- Defays, D.: An efficient algorithm for a complete link method. *Comput. J.* **20**(4), 364–366 (1977)
- Kumar, A., Kannan, R.: Clustering with spectral norm and the k-means algorithm. In: FOCS. [arXiv:1004.1823v1](https://arxiv.org/abs/1004.1823v1) (2010)
- Arthur, D., Vassilvitskii, S.: k-means++: the advantages of careful seeding. In: SODA (2007)
- Peng, J., Wei, Y.: Approximating K-means-type clustering via semidefinite programming. *SIAM J. Optim.* **18**(1), 186–205 (2007). <https://doi.org/10.1137/050641983>
- Vazirani, V.: *Approximation Algorithms*. Springer, Berlin (2001)
- Awasthi, P., Bandeira, A.S.: Relax, no need to round: integrality of clustering formulations. In: ITCS, p. 27. <https://doi.org/10.1145/2688073.2688116>. [arXiv:1408.4045](https://arxiv.org/abs/1408.4045) (2015)
- Iguchi, T., Mixon, D.G., Peterson, J., Villar, S.: Probably certifiably correct k-means clustering. *Mathematical Programming* pp 1–29. <https://doi.org/10.1007/s10107-016-1097-0>. [arXiv:1509.07983](https://arxiv.org/abs/1509.07983) (2016)
- Bunea, F., Giraud, C., Luo, X., Royer, M., Verzelen, N.: Model assisted variable clustering: minimax-optimal recovery and algorithms. [arXiv:1508.01939](https://arxiv.org/abs/1508.01939) (2018)
- Bunea, F., Ning, Y., Wegkamp, M.: Overlapping variable clustering with statistical guarantees. [arXiv:1704.06977v1](https://arxiv.org/abs/1704.06977v1) (2017)
- Bandeira, A.S.: A note on probably certifiably correct algorithms. [arXiv:1509.00824v1](https://arxiv.org/abs/1509.00824v1) (2015)
- Ames, B.P.W.: Guaranteed clustering and biclustering via semidefinite programming. *Math. Program. Ser. A* **147**(1–2), 429–465 (2014). <https://doi.org/10.1007/s10107-013-0729-x>. [arXiv:1202.3663](https://arxiv.org/abs/1202.3663)
- Iguchi, T., Mixon, D.G., Peterson, J., Villar, S.: On the tightness of an SDP relaxation of k-means. [arXiv:1505.04778](https://arxiv.org/abs/1505.04778) (2015)
- Renegar, J.: Efficient first-order methods for linear programming and semidefinite programming. [arXiv:1409.5832](https://arxiv.org/abs/1409.5832) (2014)
- Boyd, S., Vandenberghe, L.: *Convex Optimization*. Cambridge University Press. <https://doi.org/10.1080/10556781003625177>. [arXiv:1111.6189v1](https://arxiv.org/abs/1111.6189v1) (2004)
- Arora, S., Hazan, E., Kale, S.: Fast algorithms for approximate semidefinite programming using the multiplicative weights update method. In: FOCS (2005)

20. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.* **3**(1), 1–122 (2011). <https://doi.org/10.1561/22000000016>. [arXiv:1408.2927](https://arxiv.org/abs/1408.2927)
21. Awasthi, P., Sheffet, O.: Improved spectral-norm bounds for clustering. In: *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pp. 37–49. [arXiv:1206.3204v2](https://arxiv.org/abs/1206.3204v2) (2012)
22. Li, X., Chen, Y., Xu, J.: Convex relaxation methods for community detection. [arXiv:1810.00315](https://arxiv.org/abs/1810.00315) (2018)
23. Abbe, E., Bandeira, A.S., Hall, G.: Exact recovery in the stochastic block model. *IEEE: Trans. Inf. Theory* **62**(1) (2016)
24. Pirinen, A., Ames, B.: Clustering of sparse and approximately sparse graphs by semidefinite programming. [arXiv:1603.05296](https://arxiv.org/abs/1603.05296) (2016)
25. Nesterov, Y.: *Introductory Lectures on Convex Optimization: A Basic Course*, vol. 87. Springer US. <https://doi.org/10.1007/978-1-4419-8853-9>. [arXiv:1011.1669v3](https://arxiv.org/abs/1011.1669v3) (2004)
26. Nesterov, Y.: Smooth minimization of non-smooth functions. *Math. Program. Ser. A* **103**, 127–152 (2005). <https://doi.org/10.1007/s10107-004-0552-5>
27. Nesterov, Y.: Smoothing technique and its applications in semidefinite optimization. *Oper. Res.* **259**(1991), 245–259 (2007)
28. Bubeck, S.: Convex optimization: algorithms and complexity. *Found. Trends Mach. Learn.* **8**(3–4), 231–357 (2015). <https://doi.org/10.1561/22000000050>. [arXiv:1405.4980v2](https://arxiv.org/abs/1405.4980v2)
29. Ketchen, D., Shook, C.: The application of cluster analysis in strategic management research: an analysis and critique. *Strategic Manag. J.* **17**(6), 441–458 (1996)
30. Goutte, C., Toft, P., Rostrup, E., Nielsen, F.Å., Hansen, L.K.: On clustering fMRI time series. *NeuroImage* **9**(3), 298–310 (1999). <https://doi.org/10.1006/NIMG.1998.0391>
31. O'Donoghue, B., Candès, E.: Adaptive restart for accelerated gradient schemes. *Found. Comput. Math.* **15**(3), 715–732 (2015). <https://doi.org/10.1007/s10208-013-9150-3>. [arXiv:1204.3982](https://arxiv.org/abs/1204.3982)
32. Andersen, E.D., Andersen, K.D.: The Mosek interior point optimizer for linear programming: an implementation of the homogeneous algorithm. In: *High Performance Optimization*, Springer, pp. 197–232. https://doi.org/10.1007/978-1-4757-3216-0_8 (2000)
33. Sun, D., Toh, K.C., Yuan, Y., Zhao, X.Y.: SDPNAL+: A Matlab software for semidefinite programming with bound constraints (version 1.0). [arXiv:1710.10604](https://arxiv.org/abs/1710.10604) (2017)
34. Rudelson, M., Vershynin, R.: Hanson-Wright inequality and sub-Gaussian concentration. [arXiv:1306.2872](https://arxiv.org/abs/1306.2872) (2013)
35. Vershynin, R.: Introduction to the non-asymptotic analysis of random matrices. <https://doi.org/10.1017/CBO9780511794308.006>. [arXiv:1011.3027](https://arxiv.org/abs/1011.3027) (2011)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.