SUPPLEMENTARY MATERIALS: Binary Component Decomposition Part I: The Positive-Semidefinite Case*

Richard Kueng[†] and Joel A. Tropp[†]

SM1. Dimension reduction for Algorithm 2.1. As it is stated, Algorithm 2.1 requires solving semidefinite programs in an $n \times n$ matrix variable. In this section, we develop an equivalent procedure that optimizes over a much lower-dimensional space of matrices. This approach, which we document in Algorithm SM1.1, has significantly lower resource usage.

Theorem SM1.1 (Efficient sign component decomposition). *Let $\boldsymbol{A} \in \mathbb{H}_n$ be a rank-r correlation matrix that admits a sign component decomposition*

$$\boldsymbol{A} = \sum_{i=1}^{r} \tau_i \boldsymbol{s}_i \boldsymbol{s}_i^{\mathsf{t}} \quad \textit{where} \quad \boldsymbol{s}_i \in \{\pm 1\}^n \textit{ and } (\tau_1, \dots, \tau_r) \in \Delta_r^+.$$

Assume that the family $\mathcal{S} = \{\boldsymbol{s}_1, \dots, \boldsymbol{s}_r\}$ of sign components is Schur independent. Then Algorithm SM1.1 computes the sign component decomposition up to trivial symmetries. That is, the output is the unordered set of pairs $\{(\tau_i, \xi_i \boldsymbol{s}_i) : 1 \le i \le r\}$, where $\xi_i \in \{\pm 1\}$ are signs. This algorithm can be implemented with arithmetic cost $\mathcal{O}(n^3 \mathrm{polylog}(r))$.

Up to logarithmic factors in r, the running time for Algorithm SM1.1 matches the cost of computing a full eigenvalue decomposition of a dense $n \times n$ symmetric matrix.

Proof. Let $\mathcal{F} = \mathrm{conv}\{\boldsymbol{s}\boldsymbol{s}^{\mathsf{t}} : \boldsymbol{s} \in \mathcal{S}\} \subset \mathcal{E}_n$ be the simplicial face of the elliptope that contains the matrix $\boldsymbol{A}$. Let $\boldsymbol{Q} = \mathrm{orth}(\boldsymbol{A}) \in \mathbb{R}^{n \times r}$ be a matrix with orthonormal columns that span the range of $\boldsymbol{A}$. In particular, $\boldsymbol{P} = \boldsymbol{Q}\boldsymbol{Q}^{\mathsf{t}}$ is the orthogonal projector onto the range of $\boldsymbol{A}$. We can use these matrices to compress all of the optimization problems that arise in Algorithm 2.1.

We begin with the random optimization problem (5.2). It is not hard to check that the feasible set of (5.2) can be rewritten as follows. Let $\boldsymbol{q}_j^{\mathsf{t}} \in \mathbb{R}^r$ be the ith row of the matrix $\boldsymbol{Q}$. Then

$$\mathcal{F} = \{\boldsymbol{X} : \mathrm{trace}(\boldsymbol{P}\boldsymbol{X}) = n \text{ and } \boldsymbol{X} \in \mathcal{E}_n\} = \{\boldsymbol{Q}\boldsymbol{Y}\boldsymbol{Q}^{\mathsf{t}} : \boldsymbol{q}_j^{\mathsf{t}} \boldsymbol{Y} \boldsymbol{q}_j = 1 \text{ for each } j \text{ and } \boldsymbol{Y} \succcurlyeq \boldsymbol{0}\}.$$

Indeed, recall that $\boldsymbol{P}\boldsymbol{s} = \boldsymbol{s}$ for $\boldsymbol{s} \in \mathcal{S}$. Each feasible point $\boldsymbol{X}$ for (5.2) belongs to the face $\mathcal{F}$, so it must satisfy $\boldsymbol{X} = \boldsymbol{P}\boldsymbol{X}\boldsymbol{P}$. Expanding the orthogonal projectors, we obtain the parameterization $\boldsymbol{X} = \boldsymbol{Q}\boldsymbol{Y}\boldsymbol{Q}^{\mathsf{t}}$ where $\boldsymbol{Y} = \boldsymbol{Q}^{\mathsf{t}}\boldsymbol{X}\boldsymbol{Q} \in \mathbb{H}_r$. Moreover, $\boldsymbol{X}$ is psd. According to the conjugation rule (Fact 1.1), this is equivalent to demanding $\boldsymbol{Y} \succcurlyeq \boldsymbol{0}$ Finally, the diagonal contraints $\mathbf{e}_j^{\mathsf{t}} \boldsymbol{X} \mathbf{e}_j = 1$ translate directly into the conditions $\boldsymbol{q}_j^{\mathsf{t}} \boldsymbol{Y} \boldsymbol{q}_j = 1$ for each index j.

We can conjugate the last display by the orthonormal matrix $\boldsymbol{Q}$ to see that

$$\tilde{\mathcal{F}} = \mathrm{conv}\{\boldsymbol{Q}^{\mathsf{t}}\boldsymbol{s}\boldsymbol{s}^{\mathsf{t}}\boldsymbol{Q} : \boldsymbol{s} \in \mathcal{S}\} = \{\boldsymbol{Y} \in \mathbb{H}_r : \boldsymbol{q}_j^{\mathsf{t}} \boldsymbol{Y} \boldsymbol{q}_j = 1 \text{ for each } j \text{ and } \boldsymbol{Y} \succcurlyeq \boldsymbol{0}\}.$$

*Supplementary material for SIMODS MS#M127861.
https://doi.org/10.1137/19M1278612

[†]Department of Computing + Mathematical Sciences, California Institute of Technology, Pasadena, CA 91125-5000 USA (rkueng@caltech.edu, jtropp@cms.caltech.edu).

Moreover, the set $\tilde{\mathcal{F}}$ on the left-hand side is a simplex. As a consequence, we can draw a standard normal vector $\boldsymbol{g} \in \mathbb{R}^r$ and solve the optimization problem

$$\underset{\boldsymbol{Y} \in \mathbb{H}_r}{\text{maximize}} \quad \boldsymbol{g}^{\mathsf{t}} \boldsymbol{Y} \boldsymbol{g} \quad \text{subject to} \quad \boldsymbol{q}_j^{\mathsf{t}} \boldsymbol{Y} \boldsymbol{q}_j = 1 \text{ for each } j \text{ and } \boldsymbol{Y} \succcurlyeq \mathbf{0}. \tag{SM1.1}$$

According to Lemma 5.1, the unique solution will be a matrix $\boldsymbol{Y}_\star = \boldsymbol{Q}^{\mathsf{t}} \boldsymbol{s} \boldsymbol{s}^{\mathsf{t}} \boldsymbol{Q}$ for some $\boldsymbol{s} \in \mathcal{S}$.

The deflation step of Algorithm 2.1 can also be mapped down to the simplex $\tilde{\mathcal{F}}$. We just need to solve

$$\underset{\zeta \in \mathbb{R}}{\text{maximize}} \quad \zeta \quad \text{subject to} \quad \zeta(\boldsymbol{Q}^{\mathsf{t}} \boldsymbol{A} \boldsymbol{Q}) + (1-\zeta) \boldsymbol{Y}_\star \succcurlyeq \mathbf{0}.$$

As before, Lemma 5.2 ensures that this procedure extracts the rank-one component $\boldsymbol{Y}_\star$ from $\boldsymbol{Q}^{\mathsf{t}} \boldsymbol{A} \boldsymbol{Q}$, decreasing the rank by one.

Next, let us propose a further simplification to the random optimization problem (SM1.1) by noticing that the equality constraints are always redundant. First, rewrite the constraints as

$$\operatorname{trace}(\boldsymbol{q}_j \boldsymbol{q}_j^{\mathsf{t}} \boldsymbol{Y}) = 1 \quad \text{for } j = 1, \dots, n.$$

Of course, each constraint matrix $\boldsymbol{q}_j \boldsymbol{q}_j^{\mathsf{t}} \in \mathbb{H}_r$. In fact, the constraint matrices also satisfy additional affine constraints. For each $\boldsymbol{s} \in \mathcal{S}$, we calculate that

$$\operatorname{trace}(\boldsymbol{q}_j \boldsymbol{q}_j^{\mathsf{t}} \boldsymbol{Q}^{\mathsf{t}} \boldsymbol{s} \boldsymbol{s}^{\mathsf{t}} \boldsymbol{Q}) = \operatorname{trace}(\boldsymbol{Q}^{\mathsf{t}} \mathbf{e}_j \mathbf{e}_j^{\mathsf{t}} \boldsymbol{Q} \boldsymbol{Q}^{\mathsf{t}} \boldsymbol{s} \boldsymbol{s}^{\mathsf{t}} \boldsymbol{Q}) = \operatorname{trace}(\mathbf{e}_j \mathbf{e}_j^{\mathsf{t}} \boldsymbol{P} \boldsymbol{s} \boldsymbol{s}^{\mathsf{t}} \boldsymbol{P}) = \operatorname{trace}(\mathbf{e}_j \mathbf{e}_j^{\mathsf{t}} \boldsymbol{s} \boldsymbol{s}^{\mathsf{t}}) = \langle \mathbf{e}_j, \, \boldsymbol{s} \rangle^2 = 1.$$

Therefore, the constraint matrices lie in an affine subspace of $\mathbb{H}_r$ with dimension $\binom{r+1}{2} - r = \binom{r}{2} + 1$. We can select a maximal linearly independent subset of $\{\boldsymbol{q}_j \boldsymbol{q}_j^{\mathsf{t}} : j = 1, \dots, n\}$ and enforce this smaller family of constraints.

To conclude, observe that Algorithm SM1.1 involves $2(r-1)$ semidefinite programs with variable $\boldsymbol{X} \in \mathbb{H}_r$. The number of affine constraints in each SDP is bounded by $\binom{r}{2} + 2$. Standard interior point solvers [SMAHO98] can solve such problems to fixed accuracy in time $\mathcal{O}(r^{6.5})$. This bound can ostensibly be improved to $\mathcal{O}(r^5 \text{polylog}(r))$ using a method proposed in the theoretical algorithms literature [SMLSW15, Table 2].

Finally, we must account for the cost of computing a basis for the range of the input matrix $\boldsymbol{A}$ and lifting the solution from the lower-dimensional space back to the original sign components. For Algorithm SM1.1, this leads to a total runtime of $\mathcal{O}(n^3 \text{polylog}(r))$ using the theoretical method. This bound relies on the restriction (2.3) that $r = \mathcal{O}(\sqrt{n})$ for Schur independence. ■

REFERENCES

[AHO98] F. Alizadeh, J.-P. A. Haeberly, and M. L. Overton. Primal-dual interior-point methods for semidefinite programming: convergence rates, stability and numerical results. *SIAM J. Optim.*, 8(3):746–768, 1998.

[LSW15] Y. T. Lee, A. Sidford, and S. C. W. Wong. A faster cutting plane method and its implications for combinatorial and convex optimization. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 1049–1065, Oct 2015.

Algorithm SM1.1 *Efficient sign component decomposition* (2.1) *of a matrix with Schur independent components.* Implements the procedure in Theorem SM1.1.

Input: Rank-r correlation matrix $\boldsymbol{A} \in \mathbb{H}_n$ that satisfies (3.2)
Output: Sign components $\{\tilde{\boldsymbol{s}}_1, \dots, \tilde{\boldsymbol{s}}_r\} \subset \{\pm 1\}^n$ and convex coefficients $\tilde{\boldsymbol{\tau}} \in \Delta_r^+$ where $\boldsymbol{A} = \sum_{i=1}^r \tilde{\tau}_i \tilde{\boldsymbol{s}}_i \tilde{\boldsymbol{s}}_i^{\mathsf{t}}$

function EfficientSignComponentDecomposition($\boldsymbol{A}$)
  $[n, \sim] \leftarrow$ `size`($\boldsymbol{A}$) and $r \leftarrow \operatorname{rank}(\boldsymbol{A})$
  $\boldsymbol{Q} \leftarrow$ `orth`($\boldsymbol{A}$) ▷ Orthonormal basis for the range of $\boldsymbol{A}$
  $\boldsymbol{M} \leftarrow \boldsymbol{Q}^{\mathsf{t}} \boldsymbol{A} \boldsymbol{Q}$ ▷ Compress the input matrix
  Use RRQR to find a maximal independent set of constraints:

$$\underset{J \subseteq \{1,\dots,n\}}{\text{maximize}} \quad |J| \quad \text{subject to} \quad \{\boldsymbol{q}_j \boldsymbol{q}_j^{\mathsf{t}} : j \in J\} \text{ is linearly independent}$$

  for $i = 1$ to $(r-1)$ **do**
    $\boldsymbol{g} \leftarrow$ `randn`($r, 1$) ▷ Draw a random direction
    Find the solution $\boldsymbol{Y}_\star$ to the semidefinite program ▷ Step 1

$$\underset{\boldsymbol{Y} \in \mathbb{H}_r}{\text{maximize}} \quad \boldsymbol{g}^{\mathsf{t}} \boldsymbol{Y} \boldsymbol{g} \quad \text{subject to} \quad \boldsymbol{q}_j^{\mathsf{t}} \boldsymbol{Y} \boldsymbol{q}_j = 1 \text{ for } j \in J \text{ and } \boldsymbol{Y} \succcurlyeq \boldsymbol{0}$$

    Factorize the rank-one matrix $\boldsymbol{Y}_\star = \boldsymbol{Q}^{\mathsf{t}} \tilde{\boldsymbol{s}}_i \tilde{\boldsymbol{s}}_i^{\mathsf{t}} \boldsymbol{Q}$ ▷ Extract a sign component
    Find the solution $\zeta_\star$ to the semidefinite program ▷ Step 2

$$\underset{\zeta \in \mathbb{R}}{\text{maximize}} \quad \zeta \quad \text{subject to} \quad \zeta \boldsymbol{M} + (1-\zeta) \boldsymbol{Y}_\star \succcurlyeq \boldsymbol{0}$$

    $\boldsymbol{M} \leftarrow \zeta_\star \boldsymbol{M} + (1-\zeta_\star) \boldsymbol{Y}_\star$ ▷ Step 3
  Factorize the rank-one matrix $\boldsymbol{M} = \boldsymbol{Q}^{\mathsf{t}} \tilde{\boldsymbol{s}}_r \tilde{\boldsymbol{s}}_r^{\mathsf{t}} \boldsymbol{Q}$ ▷ $\operatorname{rank}(\boldsymbol{M}) = 1$ in final iteration
  Find the unique solution $\tilde{\boldsymbol{\tau}} \in \mathbb{R}^r$ to the linear system ▷ Step 4

$$\boldsymbol{M} = \sum_{i=1}^r \tilde{\tau}_i \boldsymbol{Q}^{\mathsf{t}} \tilde{\boldsymbol{s}}_i \tilde{\boldsymbol{s}}_i^{\mathsf{t}} \boldsymbol{Q}$$