

A Missing proofs for the ℓ_∞ -bounded case

First we bound the diameter of the set of possible solutions for minimizing $\text{sdp}_\infty(\mathbf{a})$.

Lemma A.1. *Let P be a sample-target distribution. The optimum $\mathbf{a}^* = \operatorname{argmin}_{\mathbf{a}} \text{sdp}_\infty(\mathbf{a})$ satisfies*

$$\frac{1}{m} \sum_{i=1}^m \|a_i^* - b_i\|^2 \leq \frac{\pi}{2} \operatorname{opt}(P)$$

Proof. Let $D = \{x \in \mathbb{R}^n \mid \|x\|_\infty \leq 1\}$ and $\mathbf{a} = \{a_1, \dots, a_m\}$. Recall that by the positive semidefinite Grothendieck inequality $\text{sdp}_\infty(\mathbf{a})$ is a relaxation of the optimization problem $\max_{x \in D} x^\top M(\mathbf{a})x$, with value at most $\frac{\pi}{2}$ times larger. Thus

$$\min_{\mathbf{a}} \text{sdp}_\infty(\mathbf{a}) \leq \frac{\pi}{2} \min_{\mathbf{a}} \max_{x \in D} x^\top M(\mathbf{a})x = \frac{\pi}{2} \operatorname{opt}(P)$$

Letting $\mathbf{a}^* = \operatorname{argmin}_{\mathbf{a}} \text{sdp}_\infty(\mathbf{a})$ we then have

$$\frac{1}{m} \sum_{i=1}^m \|a_i^* - b_i\|^2 = \langle M(\mathbf{a}^*), I \rangle \leq \max_{X \succeq 0, X_{j,j}=1} \langle M(\mathbf{a}^*), X \rangle = \min_{\mathbf{a}} \text{sdp}_\infty(\mathbf{a}) \leq \frac{\pi}{2} \operatorname{opt}(P)$$

as desired. \square

Next we compute a bound on the gradient of the cost functions used to minimize $\text{sdp}_\infty(\mathbf{a})$ via online gradient descent.

Lemma A.2. *Let $\mathbf{a} = \{a_1, \dots, a_m\}$ with $a_i \in W_i$ for all i . Let $X \succeq 0$ be an $n \times n$ positive semidefinite matrix with $X_{j,j} = 1$ for $j \in \{1, \dots, n\}$. Let $f(\mathbf{a}) = \langle M(\mathbf{a}), X \rangle$. For any \mathbf{a} satisfying*

$$\sum_{i=1}^m \|a_i - b_i\|^2 \leq r^2$$

we have

$$\|\nabla f(\mathbf{a})\| \leq \frac{2nr}{m}$$

Proof. Recalling the definition of $M(\mathbf{a})$ we have

$$f(\mathbf{a}) = \langle M(\mathbf{a}), X \rangle = \frac{1}{m} \sum_{i=1}^m (a_i - b_i)^\top X (a_i - b_i).$$

Therefore the component of $\nabla f(\mathbf{a})$ corresponding to a_i is given by

$$\nabla_{a_i} f(\mathbf{a}) = \frac{2}{m} \Pi_{W_i} X (a_i - b_i).$$

Thus we estimate the norm by

$$\begin{aligned} \|\nabla f(\mathbf{a})\|^2 &= \frac{4}{m^2} \sum_{i=1}^m \|\Pi_{W_i} X (a_i - b_i)\|^2 \\ &\leq \frac{4}{m^2} \sum_{i=1}^m \|X (a_i - b_i)\|^2 \\ &\leq \frac{4}{m^2} \sum_{i=1}^m n^2 \|a_i - b_i\|^2 \end{aligned}$$

where the last line follows from the fact that $\|X\| \leq \operatorname{Tr} X = n$. Plugging in the assumed bound on $\sum_{i=1}^m \|a_i - b_i\|^2$ yields the desired result. \square

The next lemma shows that final steps in Algorithm 1 correspond to projection onto $B_r(\mathbf{b})$, the ball of radius r centered at \mathbf{b} . In particular, since $\mathbf{a} = \{a_1, \dots, a_m\}$ is restricted to the subspace where $a_i \in W_i$ for each i , the projection occurs within this subspace.

Lemma A.3. For $\mathbf{a} = \{a_1, \dots, a_m\}$ let $W \subseteq \mathbb{R}^{nm}$ be the subspace where $a_i \in W_i$ for all i . Let $\beta = \sum_{i=1}^m \|\Pi_{W_i^\perp} b_i\|^2$. For any r such that $B_r(\mathbf{b}) \cap W \neq \emptyset$, the projection $\text{Proj}(\mathbf{a})$ of \mathbf{a} onto $B_r(\mathbf{b})$ restricted to the subspace W is given by:

$$\lambda = \min \left\{ 1, \sqrt{\frac{r^2 - \beta}{\sum_{i=1}^m \|a_i - \Pi_{W_i} b_i\|^2}} \right\}$$

$$\text{Proj}(a_i) \leftarrow \lambda a_i + (1 - \lambda) \Pi_{W_i} b_i.$$

Proof. Note first that $\mathbf{b} = \Pi_W \mathbf{b} + \Pi_{W^\perp} \mathbf{b}$ for any $\mathbf{b} \in \mathbb{R}^{nm}$. The squared distance from \mathbf{b} to W is given by $\beta = \sum_{i=1}^m \|\Pi_{W_i^\perp} b_i\|^2$. Thus if $B_r(\mathbf{b}) \cap W \neq \emptyset$ then $\beta \leq r^2$ and the definition of λ in the lemma statement makes sense.

Next observe that since each $\Pi_{W_i^\perp} b_i$ is orthogonal to all vectors in W_i

$$\sum_{i=1}^m \|a_i - b_i\|^2 = \sum_{i=1}^m \|a_i - \Pi_{W_i} b_i\|^2 + \sum_{i=1}^m \|\Pi_{W_i^\perp} b_i\|^2$$

for any $\mathbf{a} \in W$. Thus the intersection $B_r(\mathbf{b}) \cap W$ is equal to those vectors $\mathbf{a} \in W$ such that $\sum_{i=1}^m \|a_i - \Pi_{W_i} b_i\|^2 \leq r^2 - \beta$. This is precisely the ball of radius $r^2 - \beta$ in W centered at $\Pi_W \mathbf{b}$. Thus, for any \mathbf{a} not already in this ball, the projection is given by moving a λ fraction of the distance along the line from \mathbf{a} to $\Pi_W \mathbf{b}$, exactly as described in the lemma statement. \square

Next we show how the fast interior point SDP solver of [14] can be used to solve an instance of sdp_∞ to the accuracy required for the proof of Theorem 2.3.

Lemma A.4. A positive semidefinite matrix X satisfying $\text{sdp}_\infty(\mathbf{a}) \leq (1 + \frac{\epsilon}{10}) \langle M(\mathbf{a}), X \rangle$ can be compute in time $\tilde{O}(n^{7/2} \log(1/\epsilon))$. Furthermore, if $\langle M(\mathbf{a}), X \rangle \leq \text{sdp}_\infty(\mathbf{a}^*) + \frac{\epsilon}{2}$ then the approximation is additive i.e. $\text{sdp}_\infty(\mathbf{a}) \leq \langle M(\mathbf{a}), X \rangle + \frac{\epsilon}{2}$.

Proof. For matrices $M, C_1, \dots, C_k \in \mathbb{R}^{n \times n}$ and $b_i \in \mathbb{R}$, the interior point solver of [14] solves SDPs of the form

$$\begin{aligned} & \text{Maximize: } \langle M, X \rangle \\ & \text{subject to: } \langle C_i, X \rangle = b_i \\ & X \succeq 0 \end{aligned} \tag{4}$$

to accuracy $(1 + \epsilon)$ in time $\tilde{O}(\sqrt{n}(kn^2 + k^\omega + n^\omega) \log(1/\epsilon))$. In our case $k = n$ as there are n constraints of the form $X_{ii} = 1$ which can be equivalently written as $\langle e_i e_i^\top, X \rangle = 1$. Thus the dominant term in the runtime is $\tilde{O}(\sqrt{n}kn^2 \log(1/\epsilon)) = \tilde{O}(n^{7/2} \log(1/\epsilon))$ as desired.

Running the solver with accuracy parameter $\frac{\epsilon}{10}$ yields a solution X such that

$$\text{sdp}_\infty(\mathbf{a}) \leq \left(1 + \frac{\epsilon}{10}\right) \langle M(\mathbf{a}), X \rangle. \tag{5}$$

Letting $\mathbf{a}^* = \text{argmin}_{\mathbf{a}} \text{sdp}_\infty(\mathbf{a})$ we have by the positive-semidefinite Grothendieck inequality that

$$\text{sdp}_\infty(\mathbf{a}^*) \leq \frac{\pi}{2} \text{opt}(P) \leq \pi. \tag{6}$$

Thus, if \mathbf{a} satisfies $\langle M(\mathbf{a}), X \rangle \leq \text{sdp}_\infty(\mathbf{a}^*) + \frac{\epsilon}{2}$, then by (5) and (6)

$$\text{sdp}_\infty(\mathbf{a}) \leq \langle M(\mathbf{a}), X \rangle + \frac{\epsilon}{10} \left(\pi + \frac{\epsilon}{2} \right) \leq \langle M(\mathbf{a}), X \rangle + \frac{\epsilon}{2}.$$

\square

B Analysis of the ℓ_2 -bounded case

The analysis of Algorithm 2 follows a similar outline to that of Algorithm 1, but is simpler in several regards. We begin with a lemma bounding the diameter of the set of feasible solutions to $\text{sdp}_2(\mathbf{a})$.

Lemma B.1. *Let P be a sample-target distribution. The optimum $\mathbf{a}^* = \text{argmin}_{\mathbf{a}} \text{sdp}_2(\mathbf{a})$ satisfies*

$$\frac{1}{m} \sum_{i=1}^m \|a_i^* - b_i\|^2 \leq \text{opt}(P)$$

Proof. Let $D = \{x \in \mathbb{R}^n \mid \|x\| \leq \sqrt{n}\}$ and $\mathbf{a} = \{a_1, \dots, a_m\}$. By the definition of $\text{sdp}_2(\mathbf{a})$ we have

$$\min_{\mathbf{a}} \text{sdp}_2(\mathbf{a}) = \min_{\mathbf{a}} \max_{x \in D} x^\top M(\mathbf{a})x = \text{opt}(P)$$

Letting $\mathbf{a}^* = \text{argmin}_{\mathbf{a}} \text{sdp}_2(\mathbf{a})$ we then have

$$\frac{1}{m} \sum_{i=1}^m \|a_i^* - b_i\|^2 = \text{Tr}(M(\mathbf{a}^*)) \leq n \|M(\mathbf{a}^*)\| = \min_{\mathbf{a}} \text{sdp}_2(\mathbf{a}) = \text{opt}(P)$$

as desired. \square

Next we bound the norm of the gradient of the cost function used in Algorithm 2.

Lemma B.2. *Let $\mathbf{a} = \{a_1, \dots, a_m\}$ with $a_i \in W_i$ for all i . Let $X = xx^\top$ with $\|x\| \leq \sqrt{n}$. Let $f(\mathbf{a}) = \langle M(\mathbf{a}), X \rangle$. For any \mathbf{a} satisfying*

$$\sum_{i=1}^m \|a_i - b_i\|^2 \leq r^2$$

we have

$$\|\nabla f(\mathbf{a})\| \leq \frac{2nr}{m}$$

Proof. Recalling the definition of $M(\mathbf{a})$ we have

$$f(\mathbf{a}) = \langle M(\mathbf{a}), X \rangle = \frac{1}{m} \sum_{i=1}^m (a_i - b_i)^\top X (a_i - b_i).$$

Therefore the component of $\nabla f(\mathbf{a})$ corresponding to a_i is given by

$$\nabla_{a_i} f(\mathbf{a}) = \frac{2}{m} \Pi_{W_i} X (a_i - b_i).$$

Thus we estimate the norm by

$$\begin{aligned} \|\nabla f(\mathbf{a})\|^2 &= \frac{4}{m^2} \sum_{i=1}^m \|\Pi_{W_i} X (a_i - b_i)\|^2 \\ &\leq \frac{4}{m^2} \sum_{i=1}^m \|X (a_i - b_i)\|^2 \\ &\leq \frac{4}{m^2} \sum_{i=1}^m n^2 \|a_i - b_i\|^2 \end{aligned}$$

where the last line follows from the fact that

$$\|X\| = \|xx^\top\| = \|x\|^2 \leq n.$$

Plugging in the assumed bound on $\sum_{i=1}^m \|a_i - b_i\|^2$ yields the desired result. \square

Next we show that an approximate eigenvector can be computed with the desired accuracy.

Lemma B.3. A vector x satisfying $\text{sdp}_2(\mathbf{a}) \leq (1 + \frac{\epsilon}{10}) \langle M(\mathbf{a}), xx^\top \rangle$ can be compute in time $\tilde{O}(\frac{mn}{\epsilon})$. Furthermore, if $\langle M(\mathbf{a}), xx^\top \rangle \leq \text{sdp}_2(\mathbf{a}^*) + \frac{\epsilon}{2}$ then the approximation is additive i.e. $\text{sdp}_2(\mathbf{a}) \leq \langle M(\mathbf{a}), xx^\top \rangle + \frac{\epsilon}{2}$.

Proof. Let L denote the $m \times n$ matrix whose rows are equal to $(a_i - b_i)$ and note that $L^\top L = M(\mathbf{a})$. By the classical power method of [15] we can compute a vector v such that $(1 + \frac{\epsilon}{10}) v^\top M(\mathbf{a}) v \geq \|M(\mathbf{a})\|$. The runtime is bounded by $\tilde{O}(\frac{1}{\epsilon})$ times the cost of multiplying a vector by the matrix $M(\mathbf{a})$. Since $M(\mathbf{a}) = L^\top L$, we can split each matrix-vector product into two steps, first multiply by L then by L^\top , for a total runtime of $O(mn)$. Thus, v can be computed in $\tilde{O}(\frac{mn}{\epsilon})$ time.

Letting $x = \sqrt{nv}$, and $X = xx^\top$ we have

$$\text{sdp}_2(\mathbf{a}) \leq \left(1 + \frac{\epsilon}{10}\right) \langle M(\mathbf{a}), X \rangle. \quad (7)$$

Letting $\mathbf{a}^* = \text{argmin}_{\mathbf{a}} \text{sdp}_\infty(\mathbf{a})$ we have that

$$\text{sdp}_2(\mathbf{a}^*) \leq \text{opt}(P) \leq 1. \quad (8)$$

Thus, if \mathbf{a} satisfies $\langle M(\mathbf{a}), X \rangle \leq \text{sdp}_2(\mathbf{a}^*) + \frac{\epsilon}{2}$, then by (7) and (8)

$$\text{sdp}_2(\mathbf{a}) \leq \langle M(\mathbf{a}), X \rangle + \frac{\epsilon}{10} \left(1 + \frac{\epsilon}{2}\right) \leq \langle M(\mathbf{a}), X \rangle + \frac{\epsilon}{2}.$$

□

We are now ready to give the full proof of Theorem 2.4.

Proof of Theorem 2.4. The main observation is that the algorithm is precisely online gradient descent where the convex cost function observed at each step is given by $f_t(\mathbf{a}) = \langle M(\mathbf{a}), X^{(t)} \rangle$. Indeed viewing \mathbf{a} as a vector in nm dimensions (one for each a_i) the component of the gradient $\nabla f_t(\mathbf{a})$ corresponding to a_i is $\frac{2}{m} \Pi_{W_i} X^{(t)} a_i$. By Lemma B.1 the optimal solution \mathbf{a}^* lies in $B_{r^*}(\mathbf{b})$, the ℓ_2 -ball of radius $r^* = \sqrt{\text{opt}(P)}$ centered at $\mathbf{b} = \{b_1, \dots, b_m\}$. Thus given an upper bound $p \geq \text{opt}(P)$ it is sufficient to limit the search to $B_r(\mathbf{b})$ for $r = \sqrt{mp}$. The last lines of the loop are just projection onto this ball (Lemma A.3). Finally, for \mathbf{a} in $B_r(\mathbf{b})$, Lemma B.2 gives $\|\nabla f_t(\mathbf{a})\| \leq \frac{2nr}{m}$.

Thus the algorithm is online gradient descent with feasible set diameter $D = 2r$ and gradients bounded by $G = \frac{2nr}{m}$. Thus by the textbook analysis of online gradient descent [11] we have

$$\frac{1}{T} \sum_{t=1}^T f_t(\mathbf{a}^{(t)}) - \min_{\mathbf{a}' \in B_r(\mathbf{b})} \frac{1}{T} \sum_{t=1}^T f_t(\mathbf{a}') \leq \frac{3GD}{2\sqrt{T}} = \frac{3np}{2\sqrt{T}}.$$

Letting $\mathbf{a}^* = \text{argmin}_{\mathbf{a}} \text{sdp}_2(\mathbf{a})$ (which we know is contained in $B_r(\mathbf{b})$) we have

$$\frac{1}{T} \sum_{t=1}^T f_t(\mathbf{a}^{(t)}) - \frac{1}{T} \sum_{t=1}^T f_t(\mathbf{a}^*) \leq \frac{3np}{2\sqrt{T}}.$$

Noting that $f_t(\mathbf{a}^*) = \langle M(\mathbf{a}^*), X^{(t)} \rangle \leq \text{sdp}_2(\mathbf{a}^*)$ yields

$$\frac{1}{T} \sum_{t=1}^T f_t(\mathbf{a}^{(t)}) \leq \text{sdp}_2(\mathbf{a}^*) + \frac{3np}{2\sqrt{T}} \leq \text{sdp}_2(\mathbf{a}^*) + \frac{\epsilon}{2}$$

where the last line follows from the choice of $T = \frac{36n^2 p^2}{\epsilon^2}$. For $t_* = \text{argmin}_t f_t(\mathbf{a}^{(t)})$ we therefore have

$$\langle M(\mathbf{a}^{(t_*)}), X^{(t_*)} \rangle = f_{t_*}(\mathbf{a}^{(t_*)}) \leq \text{sdp}_2(\mathbf{a}^*) + \frac{\epsilon}{2}.$$

Thus by Lemma B.3 $X^{(t_*)}$ is an $\frac{\epsilon}{2}$ -additive-approximate solution to $\text{sdp}_2(\mathbf{a}^{(t_*)})$, and so we conclude

$$\text{sdp}_2(\mathbf{a}^{(t_*)}) \leq \langle M(\mathbf{a}^{(t_*)}), X^{(t_*)} \rangle + \frac{\epsilon}{2} \leq \text{sdp}_2(\mathbf{a}^*) + \epsilon$$

i.e. $\mathbf{a}^{(t^*)}$ is an ϵ -additive-approximation of $\min_{\mathbf{a}} \text{sdp}_2(\mathbf{a})$.

To analyze the runtime note that each iteration requires approximately solving an instance of $\text{sdp}_2(\mathbf{a}^{(t)})$, multiplying the solution $X^{(t)}$ by each vector $a_i^{(t)} - b_i^{(t)}$, and then rescaling by $\lambda^{(t)}$. First, by Lemma B.3 the vector x_t can be computed in time $\tilde{O}(\frac{mn}{\epsilon})$. Further $X^{(t)}(a_i^{(t)} - b_i^{(t)}) = x_t \langle x_t, (a_i^{(t)} - b_i^{(t)}) \rangle$ can be computed in time $O(n)$ for each i , for a total runtime of $O(mn)$.

Finally, $\lambda^{(t)}$ can be computed in $O(mn)$ time. Putting it all together there are $O(\frac{P^2 n^2}{\epsilon^2})$ iterations each of which takes $\tilde{O}(\frac{mn}{\epsilon})$ time, given an upper bound $p \geq \text{opt}(P)$. We can find an upper bound that is at most $2 \text{opt}(P)$ by starting with $p = \frac{1}{n}$ and repeatedly doubling at most $\log(n \text{opt}(P)) = O(\log \rho)$ times. This yields the final iteration count of $(\frac{\rho^2 \log \rho}{\epsilon^2})$. \square

C Additional Details on Experiments

We based our code for the experiments (especially for the algorithm from [4]) on the publicly available code at <https://github.com/justc2/worst-case-randomly-collected>. The code is available under the MIT License. For running Algorithm 2 in practice we found that $T = 1000$ iterations was more the sufficient to compute a good solution. Though this is not a particularly scientific comparison, we found that in practice running the code on a laptop with an Intel 8th generation Core i5 and 16GB of RAM that Algorithm 2 was significantly faster than the public code for the algorithm from [4]. For example, in the snowball sampling experiment Algorithm 2 took about 3 seconds, while the algorithm from [4] took approximately 100 seconds to compile the program description into the correct form (including the automatic Schur complement reduction described in the previous section), and approximately 30 seconds to numerically solve the resulting SDP.