arXiv:1802.04332v4 [math.OC] 1 Aug 2019

Entropy-Penalized Semidefinite Programming

Mikhail Krechetov[1*], **Jakub Mareček**[2], **Yury Maximov**[1,3] and **Martin Takáč**[4]
[1] Skolkovo Institute of Science and Technology, Nobel Street 3, Moscow, Russia
[2] IBM Research – Ireland, Technology Campus Damastown, Dublin D15, Ireland
[3] Los Alamos National Laboratory, MS-B284, Los Alamos, NM 87545, USA
[4] Lehigh University, 200 West Packer Avenue, Bethlehem, PA 18015, USA
mikhail.krechetov@skoltech.ru, jakub.marecek@ie.ibm.com, yury@lanl.gov, mat614@lehigh.edu

Abstract

Low-rank methods for semi-definite programming (SDP) have gained a lot of interest recently, especially in machine learning applications. Their analysis often involves determinant-based or Schatten-norm penalties, which are difficult to implement in practice due to high computational efforts. In this paper, we propose Entropy-Penalized Semi-Definite Programming (EP-SDP)[1], which provides a unified framework for a broad class of penalty functions used in practice to promote a low-rank solution. We show that EP-SDP problems admit an efficient numerical algorithm, having (almost) linear time complexity of the gradient computation; this makes it useful for many machine learning and optimization problems. We illustrate the practical efficiency of our approach on several combinatorial optimization and machine learning problems.

1 Introduction

Semidefinite programming (SDP) has become a key tool in solving numerous problems across operations research, machine learning, and artificial intelligence. While there are too many applications of SDP to present even a representative sample, inference in graphical models [Wainwright and Jordan, 2008; Erdogdu *et al.*, 2017], multi-camera computer vision [Torr, 2003], and applications of polynomial optimization [Parrilo, 2003; Lasserre, 2015] in power systems [Ghaddar *et al.*, 2016] stand out. Under some assumptions [Madani *et al.*, 2014], the rank at the optimum of the SDP relaxation is bounded from above by the tree-width of a certain hypergraph, plus one. When a rank-one solution is not available, it is often not needed [Mareček and Takáč, 2017], as one should like to construct a stronger SDP relaxation.

Penalization of the objective is a popular approach for obtaining low-rank solutions, at least in theory [Recht *et al.*, 2010; Lemon *et al.*, 2016; Zhou, 2019; Fawzi *et al.*, 2019]. Notice that without a further penalization, an interior-point method for SDP provides a solution on the boundary of the feasible set, where SDP corresponds to the optimum of the highest rank, whenever there are optima of multiple ranks available. The use of a penalization provides a counter-balance in this respect. In practice, however, the penalties are often ignored, as it is believed that their computation is too demanding for large-scale problems and does not guarantee low-rank solutions in general.

An alternative approach develops numerical optimization methods that seek *a priori* low-rank solutions. This approach, widely attributed to [Burer and Monteiro, 2003], considers a factorization of a semidefinite matricial variable $X = V \cdot V^\top$ with $V \in \mathbb{R}^{n \times k}$ for increasing $1 \leq k \ll n$. In general, the resulting problems are non-convex. Early analyses required determinant-based penalty terms [Burer and Monteiro, 2005], although no efficient implementations were known. Under mild assumptions, for large-enough k, there is a unique optimum over such a factorization even without a penalization and it recovers the optimum of the initial SDP problem [Boumal *et al.*, 2016]. For smaller values of k, it is known that the low-rank relaxation achieves $\mathcal{O}(1/k)$ relative error [Mei *et al.*, 2017]. Much more elaborate analyses [Erdogdu *et al.*, 2018] are now available. Especially when combined with efficient gradient computation, e.g. within low-rank coordinate descent [Mareček and Takáč, 2017, e.g.], this approach can tackle sufficiently large instances and is increasingly popular.

In this paper, we aim to develop a method combining both approaches, i.e., utilize an efficient low-rank-promoting penalty in the Burer-Monteiro approach. We present efficient first-order numerical algorithms for solving the resulting penalized problem, with (almost) linear-time per-iteration complexity. This makes the combined approach applicable to a wide range of practical problems.

In a case study, we focus on certain combinatorial optimization problems and inference in graphical models. We show that despite the non-convexity of the penalized problem, our approach successfully recovers rank-one solutions in practice. We compare our solutions against non-penalized SDP, belief propagation, and state-of-the-art branch-and-bound solvers of [Krislock *et al.*, 2014].

Contribution. Our contributions can be summarized as follows. We show

*Contact Author

[1] See https://github.com/mkrechetov/epsdp for the implementation details.

1. convergence properties of optimization methods employing a wide class of penalty functions that promote low-rank solutions;
2. linear-time algorithms for computing the gradient of these penalty functions;
3. computational results on the penalized SDP relaxation of maximum a posteriori (MAP) estimates in Markov random fields (MRF), which considerably improve upon the results obtained by interior-point methods and randomized rounding.

This allows for both well-performing and easy-to-analyze low-rank methods for SDPs coming from graphical models, combinatorial optimization, and machine learning.

Paper structure. This paper is organized as follows. First, we define the conic optimization problem together with a penalized form with a list of suitable penalization functions. Next, we present theoretical guarantees for solution recovery. These extend known results for solution recovery to the penalty case. Then, we consider the MAP problem in Markov Random Fields (MRF) and introduce an iterative procedure for it, together with a first-order method for solving a subproblem at each step; we also show how to compute the gradients efficiently. Finally, we provide computational experiments for different inference problems in MRF.

2 Background

SDP is the following conic optimization problem:

$$\min_{X\in\mathbb{S}^n_+} \sum_{i\in I} f_i(\operatorname{tr} XS_i) \qquad \text{(SDP)}$$
$$\text{s.t.: } g_j(\operatorname{tr} XC_j) \le 0, \qquad j \in J,$$

where $X \in \mathbb{S}^n_+$ denotes that the $n \times n$ matrix variable W is symmetric positive semidefinite, I and J are finite index sets, each f_i and g_j are convex functions $\mathbb{R}^{n\times n} \to \mathbb{R}$, and $C_j \in R^{n\times n}$ and $S_i \in \mathbb{R}^{n\times n}$ are constant matrices.

In the context of combinatorial optimization, one may also consider even more powerful methods such as Sum-of-Squares hierarchies of [Parrilo, 2003]. However, even an SDP relaxation, which is, in fact, the first step of this hierarchy, may be too computationally challenging. It is usually solved by interior-point methods in the dimension that is quadratic in the number of variables and thus becomes intractable even for medium-scale problems (with a few thousand variables). The problem becomes even less scalable for higher orders of the hierarchy since it requires one to solve SDP with $n^{\Theta(d)}$ variables, where d is the level of the hierarchy.

2.1 Low-rank Relaxations and Penalized Problem

First, let us formally define our notion of a penalty function and explain related work on first-order methods for SDP.

Assumption 1. Eq. (SDP) has an optimum solution with rank r.

Let us consider the following proxy problem:

$$P_{q,\lambda} \doteq \min_{V\in R^{n\times q}} \sum_{i\in I} f_i(\operatorname{tr}(V^\top S_i V)) + R_{q,\lambda}(V) \qquad \text{(P-SDP)}$$
$$\text{s.t.: } g_j(\operatorname{tr}(V^\top C_j V)) \le 0,\ j \in J.$$

Where $q > r$ and $R_{q,\lambda}(V)$ satisfies the following:

Definition 1 (Strict penalty function)**.** A function $\mathcal{R}_{q,\lambda}(V) : \mathbb{R}^{n\times n} \to \mathbb{R}$ is a *penalty function* that promotes low-rank solutions if for some integers $q' \le q$ and a multiplier $\lambda \in \mathbb{R}^+$:

$$\lim_{\lambda\to\infty} \mathcal{R}_{q,\lambda}(V) = \begin{cases} 0 & \text{if } \operatorname{rank}(V) < q', \\ \infty & \text{if } \operatorname{rank}(V) = q. \end{cases}$$

Moreover, if $q' = q$ and the $\operatorname{rank}(X) < q$, then $\mathcal{R}_{q,\lambda}(V) = 0,\ \forall\lambda$, $R(V)$ is a *strict penalty function*.

We use the word *penalty* instead of penalty function that promotes low-rank solutions, where there is no risk of confusion. This notion of a penalty is rather wide. When multiplied by λ, a determinant is a prime example. One may also consider functions of the following quasi-norms.

1. The nuclear norm:
$$\|X\|_* = \sum_i \sigma_i, \qquad (1)$$
where σ_i is the i-th singular value, cf. [Lemon *et al.*, 2016]. The norm is also known as a trace norm, Schatten 1-norm, and Ky Fan norm. As shown by [Srebro *et al.*, 2004], in the method of [Burer and Monteiro, 2003], one can benefit from a bi-Frobenius reformulation:
$$\|X\|_* = \min_{\substack{U\in\mathbb{R}^{n\times d} \\ V\in\mathbb{R}^{n\times d}: X=UV^T}} \|U\|_F\|V\|_F = \min_{U,V:X=UV^T} \frac{\|U\|_F^2 + \|V\|_F^2}{2}.$$
There are also truncated [Hu *et al.*, 2013] and capped variants [Sun *et al.*, 2013].
2. Schatten-p quasi-norm for $p > 0$:
$$\|X\|_{S_p} = \left(\sum_{i=1}^n \sigma_i^p(X)\right)^{1/p}, \qquad (2)$$
where $\sigma_i(X)$ denotes the i-th singular value of X.
3. A smoothed variant of Schatten-p quasi-norm by [Pogodin *et al.*, 2017] for $p, \epsilon > 0$:
$$\|X\|^p_{S_{p,\varepsilon}} = \sum_{i=1}^n \left(\sigma_i^2 + \varepsilon\right)^{\frac{p}{2}} = \operatorname{tr}\left(X^\top X + \varepsilon I\right)^{\frac{p}{2}}. \qquad (3)$$
4. Tri-trace quasi-norm of [Shang *et al.*, 2018]:
$$\|X\|_{\text{Tri-tr}} = \min_{X=UV\Upsilon^\top} \|U\|_*\|V\|_*\|\Upsilon\|_*, \qquad (4)$$
which is also the Schatten-1/3 quasi-norm.
5. Bi-nuclear (BiN) quasi-norm of [Shang *et al.*, 2018]:
$$\|X\|_{\text{BiN}} = \min_{X=UV^T} \|U\|_*\|V\|_*, \qquad (5)$$
which is also the Schatten-$1/2$ quasi-norm.
6. Frobenius/nuclear quasi-norm of [Shang *et al.*, 2018]:
$$\|X\|_{\text{F/N}} = \min_{X=UV^T} \|U\|_*\|V\|_F, \qquad (6)$$
which is also the Schatten-$2/3$ quasi-norm.

We also note there has been considerable interest in the analysis of low-rank approaches without penalization, especially in matrix-completion applications. Much of the analysis goes back to the work of [Keshavan *et al.*, 2010]. For further important contributions, see [Arora *et al.*, 2012].

2.2 Entropy Viewpoint

One could see the penalty functions introduced above from the entropy-penalization perspective. This is useful not only from a methodological standpoint, but also from a computational one. To this end, we consider the Tsallis entropy:

$$\mathcal{S}_\alpha^T(X) = \frac{1}{1-\alpha}\left(\frac{\operatorname{tr} X^\alpha}{(\operatorname{tr} X)^\alpha} - 1\right).$$

The Tsallis entropy is crucial in our study because it generalizes many popular penalties considered earlier. The Schatten p-norm coincides with the Tsallis entropy $\mathcal{S}_p^T$ over a set of matrices with a fixed trace norm, so that the tri-quasi norm and bi-nuclear norm (2–6) are covered as well. The Log-Det function, $-\log\det X$, which is also used in low-rank SDP, is up to an additive constant factor relative (Shannon) entropy taken concerning a unit matrix, while Renyi ($\mathcal{S}^R$) and von Neumann ($\mathcal{S}^N$) entropies,

$$\mathcal{S}_\alpha^R(X) = \frac{\log \operatorname{tr}(X/\operatorname{tr} X)^\alpha}{1-\alpha} \text{ and}$$
$$S_\alpha^N(X) = -\operatorname{tr}(\log(X/\operatorname{tr} X)\cdot X/\operatorname{tr}(X)),$$

respectively, can also be used as penalties to promote a low-rank solution. To the best of our knowledge, neither Renyi, von Neumann, nor Tsallis entropies have been studied in the context of low-rank SDP.

3 Exact Recovery

Let us now present a unified view of the penalties and their properties:

Lemma 1. *Any of:*

1. $\lambda \det(X)$*;*
2. $\lambda\sigma_q(X)$*, where* $\sigma_i(X)$ *denotes the* i*-th singular value of* X*;*
3. *Tsallis, Renyi, and von Neumann entropies defined on the last* $n-q+1$ *singular values;*
4. $\lambda \max\left\{0, \frac{\|X\|_*}{\max\{\sigma_{\min}(X),\sigma_q(X)\}} - q\right\}$,

is a penalty function that promotes low-rank solutions. Moreover, penalties 1–3 are strict.

Proof. Sketch. (1.) The proof is by simple algebra. (2.) If $\sigma_q(X)$ is 0, we know the rank is $q-1$ or less. Otherwise, for large values of λ, the value of the penalties goes to infinity, and hence $q' = q$. (3.) The definition of entropy assumes that $S(0,...,0) = 0$, thus all entropies are strict penalty functions by definition. (4.) First, consider the case where all non-zero singular values are equal. In that case, $\|X\|_*/\sigma_{\min}(X) = \operatorname{rank}(X)$, and subtracting q results either in a non-positive number when the rank is less than q or a positive number otherwise. If the singular values are non-equal, $\|X\|_*/\sigma_{\min}(X)$ provides an upper bound on the rank of X, which can be improved as suggested. The use of the upper bound results in the value of the penalty tending to infinity for values between q' and q in the large limit of λ. □

Crucially, under mild assumptions, any penalty allows for the recovery of the optimum of a feasible instance of (SDP) from the iterations of an algorithm on the non-convex problem in variable $V \in \mathbb{R}^{n\times r}$, such as in the methods of [Frostig *et al.*, 2014] or [Burer and Monteiro, 2003]. In contrast to the traditional results of [Burer and Monteiro, 2005], who consider the $\det$ penalty, we allow for the use of any strict penalty function.

Theorem 2. *Assume that we solve the proxy problem* (P-SDP) *iteratively and* $\mathcal{R}_{q,\lambda}(V)$ *is a strict penalty function that promotes low-rank solutions. In each iteration, if* $\mathcal{R}_{q,\lambda}(V) \neq 0$*, we increase* λ *(e.g., set* $\lambda_{t+1} = \gamma\lambda_t$*, with* $u > 1$ *as some fixed parameter). Furthermore, let us assume that the solution we found is denoted by* $\tilde{V}_q$ *with* $\operatorname{rank}(\tilde{V}_q) = q' < q$*. Let us also denote* $\tilde{V}_{q'} \in \mathbb{R}^{n\times q'}$ *some factorization of* $\tilde{V}_q\tilde{V}_q^\top$ *(such factorization exists because* $\operatorname{rank}(\tilde{V}_q) = q'$*). Also assume that we have an optimal solution of* (SDP)*,* X^* *with a* $\operatorname{rank}(X^*) = r$*.*

If

$$V_{q'+r+1} \triangleq [\tilde{V}_{q'}, \mathbf{0}_{n\times r}, \mathbf{0}_{n\times 1}] \tag{7}$$

is a local minimum of $P_{q'+r+1,\lambda}$*, then* $(\tilde{V}_{q'})\tilde{V}_{q'}^\top$ *is a global solution of SDP.*

Proof. Let us define a family of matrices for $\tau \in [0,1]$ as follows:

$$V(\tau) \triangleq [\sqrt{\tau}\tilde{V}_{q'}, \sqrt{(1-\tau)}V_*, \mathbf{0}_{n\times 1}],$$

where $(V_*)^\top(V_*)$ is some factorization of X^* with $V_* \in \mathbb{R}^{n\times r}$.

Note that $\forall\tau$, we have $\operatorname{rank}(X(\tau)) < r+q'+1$, and hence $\forall\lambda,\tau : R_{q'+r+1,\lambda}(V(\tau)) = 0$. Now, assume the contradiction, that is, $V_{q'+r+1}$ is a local optimum solution but $\tilde{V}_{q'}$ is not a global solution.

We show that $\forall\tau \in [0,1]$, $V(\tau)$ is a feasible solution. Indeed, for any $j \in J$ we have

$$\begin{aligned}
&g_j(\operatorname{tr}(V(\tau)^T C_j \operatorname{tr}(V(\tau)) \leq \\
&\quad \tau g_j(\operatorname{tr}([\tilde{V}_{q'}, \mathbf{0}_{n\times r+1}]^\top C_j[\tilde{V}_{q'}, \mathbf{0}_{n\times r+1}]))+ \\
&\quad (1-\tau)\operatorname{tr}([\mathbf{0}_{n\times q'}, V_*, \mathbf{0}_{n\times 1}]^\top C_j[\mathbf{0}_{n\times q'}, V_*, \mathbf{0}_{n\times 1}])) = \\
&\qquad\qquad \tau g_j(\operatorname{tr}(\tilde{V}_{q'}^\top C_j\tilde{V}_{q'})) + (1-\tau)\operatorname{tr}(X^*C_j)) \leq 0.
\end{aligned}$$

We just showed that for each $V(\tau), \tau \in [0,1]$ is a feasible point. Now, let us compute the objective value at this point. For all $\tau \in [0,1]$, we have

$$\begin{aligned}
&\sum_{i\in I}\operatorname{tr}(V(\tau)^\top S_i V(\tau)) \leq \tau\sum_{i\in I}\operatorname{tr}(\tilde{V}_{q'}^\top S_i\tilde{V}_{q'})+ \\
&\quad +(1-\tau)\sum_{i\in I}\operatorname{tr}(X^*S_i) < \sum_{i\in I}\operatorname{tr}(\tilde{V}_{q'}^\top S_i\tilde{V}_{q'}),
\end{aligned}$$

which is a contradiction under the assumption that $\tilde{V}_q$ is a local optimum. □

4 Efficient Implementation for MAP Inference

A pairwise Markov Random Field (MRF) is defined for an arbitrary graph $G = (V, E)$ with n vertices. We associate a binary variable $x_i \in \{-1, +1\}$ with each vertex $i \in V$. Let $\theta_i : \{\pm 1\} \to \mathbb{R}$ and $\theta_{ij} : \{\pm 1\}^2 \to \mathbb{R}$ defined for each vertex and edge of the graph be vertex and pairwise potential, respectively. Thus, *a posteriori* distribution of x follows the Gibbs distribution:

$$p(x|\theta) = \frac{1}{Z(\theta)} e^{U(x|\theta)},$$

with $U(x;\theta) = \sum_{i\in V} \theta_i(x_i) + \sum_{(i,j)\in E} \theta_{ij}(x_i, x_j)$. The maximum *a posteriori* (MAP) estimate is then

$$\hat{x} = \operatorname*{argmax}_{x\in\{-1,1\}^n} p(x|\theta) = \operatorname*{argmax}_{x\in\{-1,1\}^n} U(x;\theta), \qquad \text{(MAP)}$$

which is its turn an NP-hard binary quadratic optimization problem,

$$\hat{x} = \operatorname*{argmax}_{x\in\{-1,1\}^n} x^\top S x,$$

with indefinite matrix S. The SDP relaxation for this problem is given by [Goemans and Williamson, 1995; Nesterov, 1998]:

$$\min_{X\in\mathbb{S}_n^+} \operatorname{tr} SX, \quad \text{s.t.: } X_{ii} = 1, \qquad (8)$$

which also covers the Ising model in statistical physics and a number of combinatorial optimization problems. We believe that the approach can be extended to a general setup given by Eq. (SDP).

An entropy-penalized SDP relaxation of (8) has the form

$$\min_{X\in\mathbb{S}_n^+} \operatorname{tr} V^\top S V + R_\lambda(V), \quad \text{s.t.: } \|V^i\|_2^2 = 1, \quad \text{(EP-SDP)}$$

where V^i is the i-th column of matrix $V \in \mathbb{R}^{n\times k}$, $X = VV^\top$.

4.1 Numerical Method.

To solve Problem (EP-SDP), we use the Augmented Lagrangian method starting from a sufficiently small value of the penalty parameter $\lambda > 0$ and increasing it in geometric progression, $\lambda_{k+1} = \lambda_k \gamma$, with $\gamma > 1$, as summarized in Algorithm **??**. The efficiency of the method is due to the efficient computability of gradients of Tsallis, Renyi, and von Neumann entropies:

Lemma 3. *For any matrix $V \in \mathbb{R}^{n\times k}$ with $k = \mathcal{O}(1)$, let $X(V) = V^\top V$. Then, gradients of $\mathcal{S}_\alpha^T(X)$, $\mathcal{S}^R(X)$, and $\mathcal{S}^N(X)$ can be computed in $\mathcal{O}(n)$ time. Moreover, if the number of non-zero elements in matrix A is $\mathcal{O}(n)$, then the iteration complexity of Algorithm* **??** *is $\mathcal{O}(n)$.*

Proof. We start our analysis with Tsallis entropy. First, compute the gradient of $\mathcal{S}_\alpha^T$ in V:

$$\frac{\partial \mathcal{S}_\alpha^T(X)}{\partial V} = \frac{\alpha}{1-\alpha}\left(\frac{X^{\alpha-1}}{(\operatorname{tr} X)^\alpha} - \frac{\operatorname{tr} X^\alpha}{(\operatorname{tr} X)^{\alpha+1}} I\right) V.$$

Algorithm 1: Entropy-Penalized SDP.

Data: Quadratic matrix S of the MAP inference problem, staring point λ_0, $\gamma > 1$, step size policy $\{\eta_k\}_{k\geq 1}$ accuracy parameters ε, ϵ
Result: Solution V_* as a local minimum of (EP-SDP) of unit rank

```
begin
    V_0 ← random initialization in R^{n×k};
    while tr(V_t^⊤ V_t) − λ_max V_t > ε do
        Find local minimum of EP-SDP(S, λ_t) starting
          from V_{t−1}, assign it to V_t;
        while ∇(tr V^⊤ S V + R_λ(V)) ≤ ε do
            V = V − η_k ∇(tr V^⊤ S V +
              R_λ(V))/‖∇(tr V^⊤ S V + R_λ(V))‖_2;
            V_i ← V_i/‖V_i‖_2 for each row V_i;
        end
        λ_{t+1} = λ_t · γ;
    end
end
```

Return: first singular vector of V_t.;

Similarly for Renyi, $\mathcal{S}^R(X)$, and von Neumann, $\mathcal{S}^N(X)$, entropies we have

$$\frac{\partial \mathcal{S}_\alpha^R(X)}{\partial V} = \frac{\alpha}{1-\alpha}\frac{X^{\alpha-1}}{\operatorname{tr} X^\alpha}\left(I - \frac{X}{\operatorname{tr} X}\right) V$$

and

$$\frac{\partial \mathcal{S}_\alpha^N(X)}{\partial V} = \frac{\operatorname{tr} X I - X}{(\operatorname{tr} X)^2}\left(I + \log\frac{X}{\operatorname{tr} X}\right) V.$$

Following [Holmes *et al.*, 2007], the singular-value decomposition of matrix $V = U_1 D U_2$ with $U_1 \in \mathbb{R}^{n\times n}$, $D \in \mathbb{R}^{n\times k}$, and $U_2 \in \mathbb{R}^{k\times k}$ can be performed in $\mathcal{O}(\min(nk^2, n^2k)) = \mathcal{O}(nk^2)$ time.

For any $\alpha > 1$, the product $X^{\alpha-1}\cdot V = U_1 D^{2\alpha-1} U_2$ can be computed in time $\mathcal{O}(n)$ together with $\operatorname{tr} X^\alpha = \operatorname{tr} D^{2\alpha}$ and $\operatorname{tr} X = \operatorname{tr} D^2$. Thus, for a fixed k, the gradient $\frac{\partial \mathcal{S}^T(X)}{\partial V}$ computation time is linear in its dimension. (Here, for any $\alpha \in (0,1)$, we use the identity $\partial\lambda_i = \mathbf{v_i}^\top \partial X \mathbf{v_i}$.) To finish the proof of the statement, it remains to note that matrix-vector multiplication takes $\mathcal{O}(n)$ time for any matrix with $\mathcal{O}(n)$ non-zero entries. □

5 Case Study

In this section, we compare our penalized algorithm with other conventional approaches to MAP problems. We fix the width of factorization to $k = 10$, since there is no significant gain in practice for larger values of k, cf. [Mei *et al.*, 2017]. We choose $2\eta_k\beta = 1$, where β is the Lipschitz constant of the gradient in ℓ_2 norm and $\gamma = 3/2$. Parameters λ_0 and γ of Algorithm **??** are usually chosen by a few iterations of random search. It is usually enough to have about 35 iterations for penalty updates and a few hundred iterations to find a local minimum using Algorithm **??**. We emphasize that matrices we obtain by solving EP-SDP are rank-one solutions on all

Instance	gka1f	gka2f	gka3f	gka4f	gka5f
SDP					
objective	59426	97809	1347603	168616	185090
upper bound	66783	109826	152758	out of	out of
time [s]	669	673	592	memory	memory
EP-SDP					
objective	60840	**99268**	**136567**	**170669**	**189762**
upper bound	n/a	n/a	n/a	n/a	n/a
time [s]	3.3	5.0	5.3	5.2	5.7
Gurobi					
objective	**64678**	97594	131898	162875	189324
upper bound	73267	112223	153726	190073	218428
time [s].	70	70	71	70	70

Table 1: Results for the BiqMac collection.

MAP instances presented. Thus, we do not need any further rounding procedure.

First, in Table 1, we show the performance of our algorithm on selected hard MAP inference problems from the BiqMac collection[2]. We selected a few of the hardest instances ("gkaif" among them)—dense quadratic binary problems of 500 variables.

We compared our algorithm (EP-SDP with Tsallis entropy and $\alpha = 2$) with the plain-vanilla semidefinite programming instance solved by the interior-point method, possibly with rounding using the best of one thousand roundings of [Goemans and Williamson, 1995] and also with Gurobi solver for Mixed-Integer Problems. To avoid any confusion, we solve the corresponding maximization problems; by the objective value, we mean the value at a feasible solution produced by the method (e.g., rounded solution of SDP relaxation), which is a lower bound for the corresponding problem. Because these problems are of the same size (but varying density), the running time of each method is almost constant. It took around 10 minutes for CVXPY to solve the SDP relaxation, and it runs out of memory for the two problems with higher density. Within five seconds, EP-SDP obtains results that are better than what Gurobi can produce in 70 seconds.

In our study, parameter α of entropies $\mathcal{S}_\alpha^T$, $\mathcal{S}_\alpha^N$, and $\mathcal{S}_\alpha^R$ is chosen on an exponential grid from 1 to 10 with a step 1.1. After experimentation, we note that $\alpha = 1.1$ and $\alpha = 5.0$ seem to improve the results the best for the low-rank SDP with Tsallis and Renyi entropies, respectively, although the difference between different $\alpha \in (1, 10)$ is not very significant for either of the (Tsallis and Renyi) entropies.

Table 2 summarizes the results of solving the Max-Cut problem over a GSET collection of sparse graphs[3]. As we see from the experiments, the results of applying suitable entropy often outperform both the plain-vanilla SDP with the classical Goermans-Williamson rounding, the mean-field approximation, as well as the results of UGM solver[4] for loopy belief propagation and mean-field inference. It is worth noting that for several instances of the GSET graph collection, loopy belief propagation provides rather weak results. Usually, strong

[2]http://biqmac.uni-klu.ac.at/biqmaclib.html

[3]https://sparse.tamu.edu/Gset

[4]https://www.cs.ubc.ca/ schmidtm/Software/UGM.html

	GSET Instance			
	1	2	3	4
EP-SDP				
(T, $\alpha = 2.0$)	11485	11469	11429	11442
(T, $\alpha = 1.1$)	11454	11463	11444	11508
(R, $\alpha = 5$)	11508	**11519**	11496	**11531**
(R, $\alpha = 10$)	**11520**	11420	11523	11523
SDP	11372	11363	11279	11355
Loopy BP	10210	10687	10415	10389
Mean-Field	11493	11515	**11525**	11512

	GSET Instance			
	5	6	7	8
EP-SDP				
(T, $\alpha = 2$)	11427	2059	1888	1866
(T, $\alpha = 1.1$)	11506	2075	1858	1895
(R, $\alpha = 5$)	11527	**2127**	**1942**	1954
(R, $\alpha = 10$)	**11538**	2112	1940	**1958**
SDP	11313	1945	1728	1727
Loopy BP	10143	1076	964	731
Mean-Field	11528	2096	1906	1912

	GSET Instance				
	9	10	11	12	13
EP-SDP					
(T, $\alpha = 2$)	1933	1882	532	530	560
(T, $\alpha = 1.1$)	1969	1861	544	536	**568**
(R, $\alpha = 5$)	1992	1960	**550**	**548**	**568**
(R, $\alpha = 10$)	**2006**	**1982**	544	546	564
SDP	1767	1784	524	514	540
Loopy BP	1021	820	424	412	482
Mean-Field	1940	1902	542	538	564

Table 2: Results for the GSET collection.

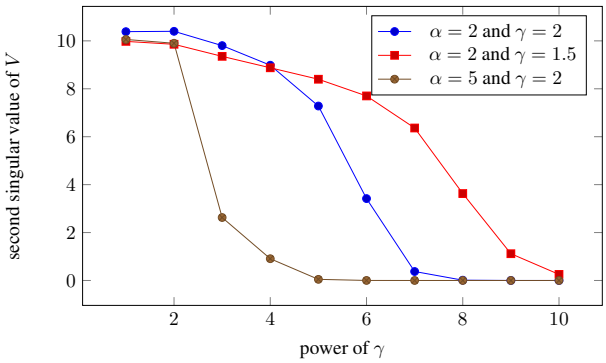


Figure 1: Rank decrement.

results of the loopy belief propagation are complementary to those of the mean-field approximation, which is supported by our empirical results. Results of both loopy belief propagation and mean-field approximation can be substantially improved using the linear-programming belief-propagation approach (LP-BP).

We also want to point out that our iterative algorithm successfully decreases the rank of the solution. The higher the penalization parameter, the lower the rank. We illustrate this

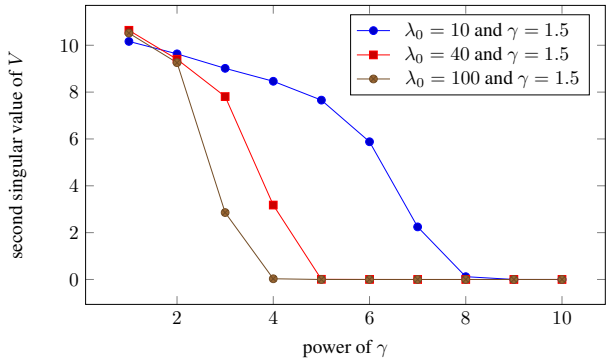


Figure 2: Rank decrement with multistart.

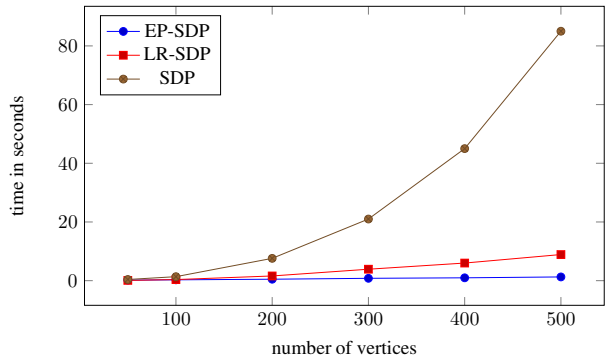


Figure 3: Time complexity

in Figure 1, where for Tsallis entropies with $\alpha = 2$ and $\alpha = 5$, we plot the second singular value of matrix V. For this plot, we considered the Max-Cut problem for the first graph from the GSET collection and the Tsallis entropy as the penalization function. In Figure 2, we illustrate the same concept for fixed penalization (Tsallis entropy with $\alpha = 2$) and different initial values of the multiplier λ. We observe that for different penalization functions and update schemes, the rank of the solution decreases gradually with each step. In practice, our iterative algorithm could be seen as a universal rounding procedure for SDP relaxations. Indeed, if we choose a large-enough penalization update (e.g., $\gamma = 2$ as in Figure 1), we easily obtain a rank-one solution that is not worse and often is substantially better than solutions obtained by randomized rounding.

Overall, we would like to stress that Algorithm **??** is very fast. This is shown in Figure 3 and Table 3, where we compare run times of EP-SDP, low-rank Burer-Monteiro approach (LR-SDP), and interior-point method solvers (SDP) for various Erdos-Renyi random graphs. From the data, we see that (assuming the fixed width of factorization $k = 10$) EP-SDP run time increases linearly with the number of vertices. Indeed, throughout the benchmark instances tested, the run time does not exceed a few seconds per each of the test cases. At the same time, the bound is often almost as good as that of the Branch and Bound Biq-Mac Solver of [Krislock *et al.*, 2014], which requires a significant amount of time.

Instance	**EP-SDP**	**LR-SDP**	**SDP**
E-R(50, 0.2)	0.2s	0.1s	0.4s
G(100, 0.2)	0.3s	0.4s	1.4s
G(200, 0.2)	0.5s	1.6s	7.6s
G(300, 0.2)	0.8s	3.9s	21.0s
G(400, 0.2)	1.0s	6.0s	45.0s
G(500, 0.2)	1.3s	8.9s	85.0s

Table 3: Run time for random graphs.

6 Conclusions

This paper presented a unified view of the penalty functions used in low-rank semidefinite programming using entropy as a penalty. This makes it possible to find a low-rank optimum, where there are optima of multiple ranks. Semidefinite programs with an entropy penalty can be solved efficiently using first-order optimization methods with linear-time per-iteration complexity, which makes them applicable to large-scale problems that appear in machine learning and polynomial optimization. Our case study illustrated the practical efficiency on binary MAP inference problems. The next step in this direction is to consider the structure of the SDP, which seems to be crucial for further scalability.

References

[Arora *et al.*, 2012] Sanjeev Arora, Rong Ge, Ravindran Kannan, and Ankur Moitra. Computing a nonnegative matrix factorization–provably. In *Proceedings of the forty-fourth annual ACM Symposium on Theory of Computing*, pages 145–162. ACM, 2012.

[Boumal *et al.*, 2016] Nicolas Boumal, Vlad Voroninski, and Afonso Bandeira. The non-convex Burer-Monteiro approach works on smooth semidefinite programs. In *Advances in Neural Information Processing Systems*, pages 2757–2765, 2016.

[Burer and Monteiro, 2003] Samuel Burer and Renato DC Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming*, 95:329–357, 2003.

[Burer and Monteiro, 2005] Samuel Burer and Renato DC Monteiro. Local minima and convergence in low-rank semidefinite programming. *Mathematical Programming*, 103(3):427–444, 2005.

[Erdogdu *et al.*, 2017] Murat Erdogdu, Yash Deshpande, and Andrea Montanari. Inference in graphical models via semidefinite programming hierarchies. In *Advances in Neural Information Processing Systems*, pages 416–424, 2017.

[Erdogdu *et al.*, 2018] Murat A. Erdogdu, Asuman Ozdaglar, Pablo A. Parrilo, and Nuri Denizcan Vanli. Convergence rate of block-coordinate maximization Burer-Monteiro method for solving large sdps. *arXiv preprint arXiv:1807.04428*, 2018.

[Fawzi *et al.*, 2019] Hamza Fawzi, James Saunderson, and Pablo A. Parrilo. Semidefinite approximations of the matrix logarithm. *Foundations of Computational Mathematics*, 19(2):259–296, Apr 2019.

[Frostig *et al.*, 2014] Roy Frostig, Sida Wang, Percy S. Liang, and Christopher D. Manning. Simple MAP inference via low-rank relaxations. In *Advances in Neural Information Processing Systems*, pages 3077–3085, 2014.

[Ghaddar *et al.*, 2016] Bissan Ghaddar, Jakub Marecek, and Martin Mevissen. Optimal power flow as a polynomial optimization problem. *IEEE Transactions on Power Systems*, 31(1):539–546, Jan 2016.

[Goemans and Williamson, 1995] Michel Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42(6):1115–1145, 1995.

[Holmes *et al.*, 2007] Michael Holmes, Alexander Gray, and Charles Isbell. Fast SVD for large-scale matrices. In *Workshop on Efficient Machine Learning at NIPS*, volume 58, pages 249–252. 2007.

[Hu *et al.*, 2013] Yao Hu, Debing Zhang, Jieping Ye, Xuelong Li, and Xiaofei He. Fast and accurate matrix completion via truncated nuclear norm regularization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 2117–2130, 2013.

[Keshavan *et al.*, 2010] Raghunandan Keshavan, Andrea Montanari, and Sewoong Oh. Matrix completion from a few entries. *IEEE Transactions on Information Theory*, 56(6):2980–2998, 2010.

[Krislock *et al.*, 2014] Nathan Krislock, Jérôme Malick, and Frédéric Roupin. Improved semidefinite bounding procedure for solving max-cut problems to optimality. *Mathematical Programming*, 143(1-2):61–86, 2014.

[Lasserre, 2015] Jean-Bernard Lasserre. *An introduction to polynomial and semi-algebraic optimization*, volume 52. Cambridge University Press, 2015.

[Lemon *et al.*, 2016] Alex Lemon, Anthony Man-Cho So, and Yinyu Ye. *Low-rank semidefinite programming: Theory and applications.* Foundations and Trends® in Optimization, 2016.

[Madani *et al.*, 2014] Ramtin Madani, Ghazal Fazelnia, Somayeh Sojoudi, and Javad Lavaei. Low-rank solutions of matrix inequalities with applications to polynomial optimization and matrix completion problems. In *53rd IEEE Conference on Decision and Control*, pages 4328–4335, Dec 2014.

[Mareček and Takáč, 2017] Jakub Mareček and Martin Takáč. A low-rank coordinate-descent algorithm for semidefinite programming relaxations of optimal power flow. *Optimization Methods and Software*, 32(4):849–871, 2017.

[Mei *et al.*, 2017] Song Mei, Theodor Misiakiewicz, Andrea Montanari, and Roberto Imbuzeiro Oliveira. Solving SDPs for synchronization and MaxCut problems via the Grothendieck inequality. In *2017 Conference on Learning Theory*, volume 65 of *Proceedings of Machine Learning Research*, pages 1476–1515, 07–10 Jul 2017.

[Nesterov, 1998] Yury Nesterov. Semidefinite relaxation and nonconvex quadratic optimization. *Optimization Methods and Software*, 9(1-3):141–160, 1998.

[Parrilo, 2003] Pablo Parrilo. Semidefinite programming relaxations for semialgebraic problems. *Mathematical programming*, page 293–320, 2003.

[Pogodin *et al.*, 2017] Roman Pogodin, Mikhail Krechetov, and Yury Maximov. Efficient rank minimization to tighten semidefinite programming for unconstrained binary quadratic optimization. *55th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1153–1159, 2017.

[Recht *et al.*, 2010] Benjamin Recht, Maryam Fazel, and Pablo A Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Review*, 52(3):471–501, 2010.

[Shang *et al.*, 2018] Fanhua Shang, James Cheng, Yuanyuan Liu, Zhi-Quan Luo, and Zhouchen Lin. Bilinear factor matrix norm minimization for robust PCA: Algorithms and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(9):2066–2080, 2018.

[Srebro *et al.*, 2004] Nathan Srebro, Jason D. M. Rennie, and Tommi S. Jaakkola. Maximum-margin matrix factorization. In *Advances in Neural Information Processing Systems*, NIPS'04, pages 1329–1336, 2004.

[Sun *et al.*, 2013] Qian Sun, Shuo Xiang, and Jieping. Ye. Robust principal component analysis via capped norms. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 311–319, 2013.

[Torr, 2003] Philip Torr. Solving Markov random fields using semidefinite programming. In *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics, AISTATS*, pages 1–8, 2003.

[Wainwright and Jordan, 2008] Martin J. Wainwright and Michael Jordan. *Graphical models, exponential families, and variational inference.* Foundations and Trends® in Machine Learning, 2008.

[Zhou, 2019] Cunlu Zhou. *Entropy, optimization and coding theory*. PhD thesis, University of Notre Dame, Notre Dame, Indiana, 2019.