Fast Algorithms for Solving SDPs on Graphs
SPUR Final Paper, Summer 2014

Yibo Gao and Yuzhou Gu
Mentor: Adrian Vladu
Project Suggested By Richard Peng

July 30, 2014

Abstract

Semidefinite programming (SDP) is a basic optimization primitive, which has been successfully applied to many combinatorial graph algorithms. Unfortunately, in the era of big data, standard methods for solving SDP's (Interior Point, Ellipsoid) are prohibitively slow. Instead of focusing on these, we shift our attention to first order methods, which appear to be very fast for practical purposes. It turns out that simply by applying mirror descent, one can solve SDP's within ϵ accuracy using $O(\epsilon^{-2})$ iterations, each of them consisting of simple matrix operations. However, the standard setup for mirror descent has the fundamental flaw that iterations are computationally expensive. In this work, we consider the SDP relaxation for MAX-CUT, and attempt to provide an efficient algorithm for solving it, which takes into account the sparsity of the input. We provide a different (and simpler) setup for mirror descent in hope of achieving cheaper iterations.

1 Introduction

Semidefinite programming (SDP) is a fundamental primitive employed in modern approximation algorithms. Starting with the work of Goemans and Williamson [GW95], algorithmists started focusing on using the power of this method. One prohibitive factor though, remains the efficiency of solving semidefinite programs. While we have a pretty good understanding of linear programs, and standard approaches such as the Interior Point or Ellipsoid Method work very well in practice, SDP's remain more elusive. One insight that came to the rescue was that in many cases, when solving SDP's we only need need to find solutions that satisfy the set of constraints *approximately*. Therefore, one can use first order methods to find a good solution in a small number of iterations. Arora and Kale [AK07] formalized this intuition and provided a new approach to solving SDP's, where the linear constraints are satisfied within a $1 + \epsilon$ multiplicative factor.

While this looks pretty promising, the ϵ dependency in the running time is quite bad (ϵ^{-5}), and it would be nicer to produce efficient algorithms that solve SDP's crafted for specific problems. More specifically, we consider two SDP relaxations arising in combinatorial graph problems: MAX-CUT and the Lovász Theta Function. Past work shows that at least in the case of LP's, underlying graph structure enables computation of much more efficient iterations, using primitives such as Laplacian solvers [DS08]. Therefore, our goal is two-fold: we want to find a more structured way of approximately solving SDP's, and we want to exploit the underlying graph structure in order to efficiently solve each iteration of our algorithm.

Our work so far consists of providing a proper setup for first order methods, in our case mirror descent [NY83], in order to replace the multiplicative weights update. We first describe the standard setup which achieves a converge rate of ϵ^{-2}, but it has the fundamental flaw that each iteration is expensive because it requires computing matrix exponentials. Then we give a simpler setup, that is more similar in spirit to projected gradient descent: after taking gradient steps, we project back onto our domain simply by zeroing out the negative eigenvalues of our current iterate.

We also discovered that one specific condition from the setup is unneeded. Namely, after we add a penalty term to the objective function [Lee14], we no longer need the trace constraints. Although deleting it will theoretically make our algorithm faster, it results in slow convergence experimentally. We are now trying to fix that problem and also working on different setups for mirror descent and custom iterative methods.

2 Preliminaries

2.1 Semidefinite Programming

A semidefinite program (SDP) is a problem in the form

$$\begin{aligned} \min\ & \mathbf{C} \cdot \mathbf{X} \\ \text{s.t.}\quad & \mathbf{A}_k \cdot \mathbf{X} \leq b_k, \quad k = 1, \ldots, m \\ & \mathbf{X} \succeq \mathbf{0} \end{aligned}$$

where $\mathbf{C}, \mathbf{A}_k, \mathbf{X} \in \mathbb{S}^n$, the space of $n \times n$ symmetric matrices equipped with inner product $\cdot$ defined as $\mathbf{A} \cdot \mathbf{B} = \operatorname{tr}(\mathbf{A}^T\mathbf{B})$.

A special subclass of SDP is linear programming (LP), which is SDP where $\mathbf{C}$ and $\mathbf{A}_k$ are all diagonal.

2.2 Examples

We look at some examples of SDPs on graphs.

2.2.1 MAX-CUT Problem

In the MAX-CUT problem, we are given a positively weighted undirected graph $G = (V, E, W)$. The objective is to find a partition $\{S_1, S_2\}$ of V such that $\sum_{e \in (S_1 \times S_2) \cap E} w(e)$ is maximized. The decision version of the MAX-CUT problem is known to be NP-Hard [Kar72]. Goemans and Williamson [GW95] gave a randomized polynomial-time γ-approximation algorithm for MAX-CUT, where $\gamma = \frac{2}{\pi} \min_{\theta \in [0,\pi]} \frac{\theta}{1-\cos\theta} \approx .87856$. The Goemans-Williamson algorithm involves solving the SDP

$$\begin{aligned} \max\ & \mathbf{L}_G \cdot \mathbf{X} \\ \text{s.t.}\quad & \textsc{diag}(\mathbf{X}) = \mathbf{1} \\ & \mathbf{X} \succeq \mathbf{0} \end{aligned}$$

where $\mathbf{L}_G$ is the Laplacian of G.

[WFLM13] and [BC06] are two applications in machine learning of variations of the MAX-CUT problem.

2.2.2 Lovász Theta Function

The Lovász theta function is another SDP problem on graphs. Given an unweighted undirected graph $G = (V, E)$, the Lovász theta function $\vartheta(G)$ is defined as

$$\begin{aligned} \max \ & \mathbf{1} \cdot \mathbf{X} \\ \text{s.t.} \quad & \operatorname{tr} \mathbf{X} = 1 \\ & \mathbf{X}_{ij} = 0, \quad \forall (i,j) \in E, i \neq j \\ & \mathbf{X} \succeq \mathbf{0} \end{aligned}$$

2.3 Algorithms

We summarize two methods used to solve semidefinite programs.

2.3.1 Multiplicative Weights Update Method

The multiplicative weights update (MWU) method is a widely used primitive. [AHK12] It appeared independently in a number of different places (Machine Learning, approximation algorithms). Plotkin et al. [PST95] used this method to solve fractional packing and covering problems. More specifically, given an LP from this class, they use MWU in order to efficiently obtain a solution that satisfies all the linear constraints within a multiplicative $(1 + \epsilon)$ factor. Later, Arora and Kale [AK07] generalized this method and applied it to solving SDP's.

2.3.2 Mirror Descent

While MWU is a tempting approach, it is formulated using an oracle used for solving each iteration, and is essentially applied in a black box manner. Instead of this, we would like to resort to a more structured method, where we have a more solid handle on our algorithms. Therefore, we shift our attention to the textbook numerical methods. Lee [Lee14] and Zhu & Orecchia [AZO14] showed that the MWU method is a special case of mirror descent. [NY83] Our goal is to push this intuition further, and obtain a customized algorithm that beats MWU. In the following section, we present the standard implementation of mirror descent.

2.4 Mirror Descent Method

This method was first introduced by Nemirovski and Yudin [NY83], in order to generalize the projected gradient descent method whenever the domain we want to optimize over is an arbitrary convex set. For the sake of our presentation, we follow the description from [Bub14].

Suppose we would like to minimize an $L-$Lipschitz (with respect to a norm $||\cdot||$) convex function $f(x)$ over a convex set $\mathcal{X} \subseteq \mathbb{R}^n$. Let $\mathcal{D} \in \mathbb{R}^n$ be an open convex set such that $\mathcal{X} \subseteq \overline{\mathcal{D}}$ and $\mathcal{D} \cap \mathcal{X} \neq 0$. Let $\Phi : \mathcal{D} \to \mathbb{R}$ be a map such that

(1) Φ is α-strongly convex with respect to $||\cdot||$ and differentiable;

(2) For any $x \in \mathcal{X} \cap \mathcal{D}, \eta > 0$, there exists $y \in \mathcal{D}$ such that $\nabla\Phi(y) = \nabla\Phi(x) - \eta\nabla f(x)$;

(3) $\lim_{x\to\partial\mathcal{D}} ||\nabla\Phi(x)|| = +\infty$.

in which α-strongly convexity of Φ is defined by

$$\forall x, y \in \mathcal{X} \cap \mathcal{D}, \Phi(x) - \Phi(y) \leq \nabla\Phi(x)^T(x-y) - \frac{\alpha}{2}||x-y||^2$$

We call Φ a mirror map.

Define $\Pi_{\mathcal{X}}^{\Phi} : \mathcal{D} \to \mathcal{X} \cap \mathcal{D}$ as

$$\Pi_{\mathcal{X}}^{\Phi}(y) = \arg\min_{x\in\mathcal{X}\cap\mathcal{D}} D_{\Phi}(x, y)$$

where $D_{\Phi}(x, y)$ is a distance function, called the Bregman divergence induced by Φ. It is defined as

$$D_{\Phi}(x, y) = \Phi(x) - \Phi(y) - \nabla\Phi(y)^T(x-y)$$

Let $x^* = \arg\min_{x\in\mathcal{X}} f(x)$. Choose an R such that $R \geq D_{\Phi}(x^*, x_1)$. The mirror descent algorithm is given below.

Algorithm 1 Mirror Descent

Input: $f, \mathcal{X}, \mathcal{D}, \Phi, t$

Output: $x \in \mathcal{X}$ such that $f(x) - \min_{y\in\mathcal{X}} f(y) \leq \sqrt{\frac{2RL^2}{\alpha t}}$

1: $\eta \leftarrow \sqrt{\frac{2\alpha R}{L^2 t}}$.
2: Choose initial value $x_1 \in \mathcal{X} \cap \mathcal{D}$.
3: **for** $s = 1 \to t$ **do**
4: Choose $y_{s+1} \in \mathcal{D}$ such that $\nabla\Phi(y_{s+1}) = \nabla\Phi(x_s) - \eta\nabla f(x_s)$.
5: $x_{s+1} \leftarrow \Pi_{\mathcal{X}}^{\Phi}(y_{s+1})$.
6: **end for**
7: **return** $\frac{1}{t}\sum_{s=1}^{t} x_s$.

We have the following theorem.

Theorem 2.1. *In t iterations, the mirror descent method produces a solution x such that $f(x) - f(x^*) \leq \sqrt{\frac{2RL^2}{\alpha t}}$.*

2.5 Approximation

Suppose we are to maximize a function $f(x)$ over some constraint set $\mathcal{X}$. Let $x^* = \arg\min_{x\in\mathcal{X}} f(x)$. We assume that $f(x) \geq 0$. We call a solution $x \in \mathcal{X}$ an ϵ-approximate solution if $f(x) \geq (1-\epsilon)f(x^*)$.

3 Setup for Example Problems

In this section we show how to set up the Goemans-Williamson relaxation for MAX-CUT in order to solve it using mirror descent. In the near future, we will also consider the Lovász Theta function, which seems to be slightly harder.

3.1 MAX-CUT SDP

We make an interpretation of [Lee14].

Let $\mathbf{C}$ be a given graph Laplacian. Recall that the MAX-CUT problem is

$$\text{Minimize } -\mathbf{C}\cdot\mathbf{X} \text{ subject to } \textsc{diag}(\mathbf{X}) = \mathbf{1}, \mathbf{X} \succeq \mathbf{0}.$$

We remove the constraint $\textsc{diag}\mathbf{X} = \mathbf{1}$ by adding an L^1 penalty and transform the problem to

$$\text{Minimize } -\mathbf{C}\cdot\mathbf{X} + \sum \rho_i(\mathbf{X}_{i,i} - 1)^+ \text{ subject to } \mathbf{X} \succeq \mathbf{0}.$$

Here x^+ is defined as $\max\{0, x\}$.

Theorem 3.1. *The above two problems have the same optimal values if* $\rho_i \geq 2\sum_{j=1}^n |C_{ij}|$.

Proof. For simplicity of proof, we rewrite the two problems as follows:

- MAX-CUT-SDP-1: Minimize $g(\mathbf{X}) = -\mathbf{C}\cdot\mathbf{X}$ over $\mathcal{X}_g = \{\mathbf{X} : \mathbf{X} \succeq 0, \textsc{diag}\mathbf{X} = \mathbf{1}\}$.
- MAX-CUT-SDP-2: Minimize $h(\mathbf{X}) = -\mathbf{C}\cdot\mathbf{X} + \sum \rho_i(\mathbf{X}_{ii} - 1)^+$ over $\mathcal{X}_h = \mathbb{S}^n$.

Let the optima of MAX-CUT-SDP-1 and MAX-CUT-SDP-2 be $\mathbf{X}_g^*$ and $\mathbf{X}_h^*$ respectively. Let $g^* = g(\mathbf{X}_g^*), h^* = h(\mathbf{X}_h^*)$. $\mathcal{X}_g \subseteq \mathcal{X}_h$, so $g^* \geq h^*$. So we only need to prove $h^* \geq g^*$.

Because $\mathbf{X}_h^* \succeq \mathbf{0}$, there exist $\mathbf{v}_1, \ldots, \mathbf{v}_n$ such that $\mathbf{X}_{h,i,j}^* = \mathbf{v}_i \cdot \mathbf{v}_j, \forall i, j$. Then

$$h^* = -\sum_{i,j} \mathbf{C}_{ij}\mathbf{v}_i\cdot\mathbf{v}_j + \sum_{i=1}^n \rho_i(\mathbf{v}_i\cdot\mathbf{v}_i - 1)^+$$

We use L^2-norm $||\cdot||_2$ for vectors in the analysis below.

For a vector $\mathbf{v}_k$, the terms in h^* relevant to it is

$$-2\left(\sum_{j\neq k}\mathbf{C}_{kj}\mathbf{v}_j\right)\cdot\mathbf{v}_k-\mathbf{C}_{kk}\mathbf{v}_k\cdot\mathbf{v}_k+\rho_k(\mathbf{v}_k\cdot\mathbf{v}_k-1)^+$$

If $\left(\sum_{j\neq k}\mathbf{C}_{kj}\mathbf{v}_j\right)\cdot\mathbf{v}_k<0$, we simply change the sign of $\mathbf{v}_k$ and get a smaller answer. If $\left(\sum_{j\neq k}\mathbf{C}_{kj}\mathbf{v}_j\right)\cdot\mathbf{v}_k=0$, we replace $\mathbf{v}_k$ by $\mathbf{v}'_k$ so that $|\mathbf{v}_k|=|\mathbf{v}'_k|$ and $\left(\sum_{j\neq k}\mathbf{C}_{kj}\mathbf{v}_j\right)\cdot\mathbf{v}'_k\neq 0$. Then either the $\mathbf{v}'_k$ or $-\mathbf{v}'_k$ will yield a smaller answer. So we have $\left(\sum_{j\neq k}\mathbf{C}_{kj}\mathbf{v}_j\right)\cdot\mathbf{v}_k>0$ for all k. If for some k, $|\mathbf{v}_k|<1$, we replace it by $\mathbf{v}'_k=\frac{\mathbf{v}_k}{|v_k|}$. The answer will become smaller. So $|\mathbf{v}_k|\geq 1$ for all k.

Let $\mathbf{v}_k$ be the vector that has the greatest norm. We replace it by $\mathbf{v}'_k=\frac{\mathbf{v}_k}{|\mathbf{v}_k|}$. The change of h is

$$\begin{aligned}
&\left(-\sum_{j\neq k}2\mathbf{C}_{kj}\mathbf{v}'_k\cdot\mathbf{v}'_j-\mathbf{C}_{kk}\mathbf{v}'_k\cdot\mathbf{v}'_k+\rho_k(\mathbf{v}'_k\cdot\mathbf{v}'_k-1)^+\right)\\
&-\left(-\sum_{j\neq k}2\mathbf{C}_{kj}\mathbf{v}_k\cdot\mathbf{v}_j-\mathbf{C}_{kk}\mathbf{v}_k\cdot\mathbf{v}_k+\rho_k(\mathbf{v}_k\cdot\mathbf{v}_k-1)^+\right)\\
&=\left(1-\frac{1}{|\mathbf{v}_k|}\right)\sum_{j\neq k}2\mathbf{C}_{kj}\mathbf{v}_k\cdot\mathbf{v}_j+\left(1-\frac{1}{|\mathbf{v}_k|^2}\right)\mathbf{C}_{kk}\mathbf{v}_k\cdot\mathbf{v}_k-\rho_k(|\mathbf{v}_k|^2-1)\\
&=\frac{|\mathbf{v}_k|-1}{|\mathbf{v}_k|}\left(2\sum_{j\neq k}\mathbf{C}_{kj}\mathbf{v}_k\cdot\mathbf{v}_j+\frac{|\mathbf{v}_k|+1}{|\mathbf{v}_k|}\mathbf{C}_{kk}\mathbf{v}_k\cdot\mathbf{v}_k-(|\mathbf{v}_k|^2+|\mathbf{v}_k|)\rho_k\right)\\
&\leq\frac{|\mathbf{v}_k|-1}{|\mathbf{v}_k|}\left(2\sum_{j\neq k}|\mathbf{C}_{kj}||\mathbf{v}_k||\mathbf{v}_j|+2|\mathbf{C}_{kk}||\mathbf{v}_k|^2-\rho_k|\mathbf{v}_k|^2\right)\\
&\leq\frac{|\mathbf{v}_k|-1}{|\mathbf{v}_k|}\left(2\sum_{j\neq k}|\mathbf{C}_{kj}||\mathbf{v}_k|^2-2\sum_{j}|\mathbf{C}_{kj}||\mathbf{v}_k|^2\right)\\
&\leq 0
\end{aligned}$$

Repeat the process and we will get a optimum $\mathbf{X}^*_h\in\mathcal{X}_g$. So $h^*\geq g^*$ and we are done. □

We can further rewrite the problem by rescaling the variable as

$$\text{Minimize } -\sqrt{\text{DIAG}(\rho^{-1})}\mathbf{C}\sqrt{\text{DIAG}(\rho^{-1})}\cdot\mathbf{X}+\textstyle\sum_{i=1}^n(\mathbf{X}_{ii}-\rho_i)^+ \text{ subject to } \mathbf{X}\succeq\mathbf{0}.$$

Aside from conceptual convenience (note that $-\sqrt{\text{DIAG}(\rho^{-1})}\mathbf{C}\sqrt{\text{DIAG}(\rho^{-1})}$ can be set to a multiple of the normalized Laplacian of the graph), this is going to be a useful trick

in order to make the function we are trying to optimize 1-Lipschitz. This renormalization is going to be reflected in the convergence rate of the algorithm.

For the standard setup of mirror descent to work, we need to add a trace constraint. That is, we solve

$$\begin{aligned}\text{Minimize } &-\sqrt{\operatorname{DIAG}(\rho^{-1})}\mathbf{C}\sqrt{\operatorname{DIAG}(\rho^{-1})}\cdot\mathbf{X}+\textstyle\sum_{i=1}^{n}(\mathbf{X}_{ii}-\rho_i)^+\\ \text{subject to } &\mathbf{X}\succeq\mathbf{0},\ \operatorname{tr}\mathbf{X}=\textstyle\sum_{1\le i\le n}\rho_i.\end{aligned}$$

It is easily proved that this SDP has exactly the same solution as the previous one.

4 Analysis of the Standard Setup for Mirror Descent on Example Problems

In this section we analyze the number of iterations needed for mirror descent with standard setup to get an ϵ-approximation on example problems.

4.1 MAX-CUT SDP

Define $f(\mathbf{X})=-\sqrt{\operatorname{DIAG}(\rho^{-1})}\mathbf{C}\sqrt{\operatorname{DIAG}(\rho^{-1})}\cdot\mathbf{X}+\sum_{i=1}^{n}(\mathbf{X}_{ii}-\rho_i)^+$. Let $r=\sum_{1\le i\le n}\rho_i$. Let $\mathcal{X}=\{\mathbf{X}:\mathbf{X}\succeq\mathbf{0},\ \operatorname{tr}\mathbf{X}=r\}$.

Then the problem we want to solve is

$$\text{Minimize } f(\mathbf{X}) \text{ subject to } \mathbf{X}\in\mathcal{X}.$$

Let $\mathbf{X}^*=\arg\min_{\mathbf{X}\in\mathcal{X}}f(\mathbf{X})$. Let R,L,α be defined as in section 2.4. By theorem 2.1, we need $t=\frac{2RL^2}{\alpha f(\mathbf{X}^*)^2}\epsilon^{-2}$ iterations to find an ϵ-approximate solution.

We first bound $f(\mathbf{X}^*)$ because it is independent of the setup we use. Let $m=\sum_{1\le i<j\le n}|\mathbf{C}_{i,j}|$ be sum of weight of edges in the graph. We have

Theorem 4.1. $f(\mathbf{X}^*)\ge\frac{m}{2}$.

Proof. An $\mathbf{X}$ that $f(\mathbf{X})\ge\frac{m}{2}$ can be achieved by a simple greedy algorithm which was introduced by Sahni and Gonzalez [SG76]. □

We actually have $r=8m$ because

$$\begin{aligned}r&=\sum_{1\le i\le n}\rho_i\\&=2\sum_{1\le i\le n}\sum_{1\le j\le n}|\mathbf{C}_{i,j}|\\&=2(\sum_{1\le i\le n}\mathbf{C}_{i,i})+4(\sum_{1\le i<j\le n}|\mathbf{C}_{i,j}|)\\&=8\sum_{1\le i<j\le n}|\mathbf{C}_{i,j}|=8m\end{aligned}$$

Then we bound L, R, α. One thing noteworthy is that we need to modify the standard setup in appendix B a little bit because here the trace is r which is not 1. However it is easily shown that the gradient step does not change and the projection step becomes $\Pi^{\Phi}_{\mathcal{X}}(\mathbf{Y}) = \frac{r\mathbf{Y}}{\operatorname{tr}\mathbf{Y}}$.

Theorem 4.2. *f is $\frac{3}{2}$-Lipschitz with respect to the trace norm.*

Proof. Let $\mathbf{Q} = \sqrt{\operatorname{DIAG}(\rho^{-1})}\mathbf{C}\sqrt{\operatorname{DIAG}(\rho^{-1})}$. Let $f_1(\mathbf{X}) = \mathbf{Q} \cdot \mathbf{X}$, $f_2(\mathbf{X}) = \sum_{1 \le i \le n}(\mathbf{X}_{i,i} - \rho_i)^+$. Then $f(\mathbf{X}) = -f_1(\mathbf{X}) + f_2(\mathbf{X})$. Let L_1 be the Lipschitz constant of f_1, L_2 be the Lipschitz constant of f_2. Then $L \le L_1 + L_2$.

We first prove that $L_1 \le \frac{1}{2}$. Recall that the trace norm $||\mathbf{X}|| = \sum_{1 \le i \le n} |\lambda_i|$ where λ_i are eigenvalues of $\mathbf{X}$. We have

$$\begin{aligned}
L_1 &= \sup_{\mathbf{X},\mathbf{Y} \in \mathcal{X}, \mathbf{X} \neq \mathbf{Y}} \frac{f_1(\mathbf{X}) - f_1(\mathbf{Y})}{||\mathbf{X} - \mathbf{Y}||} \\
&= \sup_{\mathbf{X},\mathbf{Y} \in \mathcal{X}, \mathbf{X} \neq \mathbf{Y}} \frac{f_1(\mathbf{X} - \mathbf{Y})}{||\mathbf{X} - \mathbf{Y}||} \\
&= \sup_{\mathbf{X} \in \operatorname{Sym}_n, \operatorname{tr} X = 0, \mathbf{X} \neq \mathbf{0}} \frac{f_1(\mathbf{X})}{||\mathbf{X}||} \\
&\le \sup_{\mathbf{X} \in \operatorname{Sym}_n, \mathbf{X} \neq \mathbf{0}} \frac{f_1(\mathbf{X})}{||\mathbf{X}||} \\
&= \sup_{\mathbf{X} \in \operatorname{Sym}_n p, \mathbf{X} \neq \mathbf{0}} \frac{\operatorname{tr}(\mathbf{Q}\mathbf{X})}{||\mathbf{X}||}
\end{aligned}$$

We write $\mathbf{X} = \mathbf{V}\mathbf{D}\mathbf{V}^{-1}$ where $\mathbf{V}$ is orthogonal and $\mathbf{D}$ is diagonal. Then

$$\frac{\operatorname{tr}(\mathbf{Q}\mathbf{X})}{||\mathbf{X}||} = \frac{\operatorname{tr}(\mathbf{Q}\mathbf{V}^{-1}\mathbf{D}\mathbf{V})}{||\mathbf{V}^{-1}\mathbf{D}\mathbf{V}||} = \frac{\operatorname{tr}(\mathbf{V}\mathbf{Q}\mathbf{V}^{-1}\mathbf{D})}{||\mathbf{D}||} = \operatorname{tr}(\mathbf{V}\mathbf{Q}\mathbf{V}^{-1}\frac{\mathbf{D}}{||\mathbf{D}||}).$$

$||\frac{\mathbf{D}}{||\mathbf{D}||}|| = 1$. So

$$L_1 \le \sup_{\mathbf{D} \text{ diagonal}, \mathbf{V} \text{ orthogonal}, ||\mathbf{D}||=1} \operatorname{tr}(\mathbf{V}\mathbf{Q}\mathbf{V}^{-1}\mathbf{D}).$$

$\mathbf{D}$ is diagonal, so $||\mathbf{D}|| = \sum_{1 \le i \le n} |\mathbf{D}_{i,i}|$. Let $\mathbf{Q}' = \mathbf{V}\mathbf{Q}\mathbf{V}^{-1}$. Let $M = \max_{1 \le i \le n} \mathbf{Q}'_{i,i}$. $\mathbf{Q} \succeq 0$, so $\mathbf{Q}' \succeq 0$. So $\mathbf{Q}'_{i,i} \ge 0$ for all i. Then

$$\operatorname{tr}(\mathbf{Q}'\mathbf{D}) = \sum_{1 \le i \le n} \mathbf{Q}'_{i,i}\mathbf{D}_{i,i} \le \sum_{1 \le i \le n} M|\mathbf{D}_{i,i}| = M.$$

So $L_1 \le M$.

Let $\lambda_{\max}$ be the largest eigenvalue of $\mathbf{Q}$. Then it is also the largest eigenvalue of $\mathbf{Q}'$ because basis change preserves eigenvalues. Note that $\mathbf{Q}$ is $\frac{1}{4}$ the normalized Laplacian

matrix of the original graph. It is a standard result [Chu97] that all eigenvalues of the normalized Laplacian matrix are no greater than 2. So $\lambda_{\max} \leq \frac{1}{2}$.

By the Courant-Fischer theorem, $M \leq \lambda_{\max}$. Combine the inequalities above and we get $L_1 \leq \frac{1}{2}$.

Then we prove that $L_2 \leq 1$.

Let $\mathbf{M} \in \mathrm{Sym}_n$. Let $\mathbf{v}_1, \ldots, \mathbf{v}_n$ be its eigenvectors. Let $\lambda_1, \ldots, \lambda_n$ be the corresponding eigenvalues. Let $\mathbf{e}_1, \ldots, \mathbf{e}_n$ be standard basis of $\mathbb{R}^n$. Let $\mathbf{e}_i = \sum_{1 \leq j \leq n} a_{i,j} \mathbf{v}_j$. Then

$$\begin{aligned}
&\sum_{1 \leq i \leq n} |\mathbf{M}_{i,i}| = \sum_{1 \leq i \leq n} |\mathbf{e}_i^T M \mathbf{e}_i| = \sum_{1 \leq i \leq n} |\sum_{1 \leq j \leq n} a_{i,j}^2 \lambda_j| \\
&\leq \sum_{1 \leq i \leq n} \sum_{1 \leq j \leq n} a_{i,j}^2 |\lambda_j| = \sum_{1 \leq j \leq n} (|\lambda_j| \sum_{1 \leq i \leq n} a_{i,j}^2) = \sum_{1 \leq j \leq n} |\lambda_j| = ||\mathbf{M}||
\end{aligned}$$

Let $\mathbf{X}, \mathbf{Y} \in \mathcal{X}, \mathbf{X} \neq \mathbf{Y}$. Then

$$\begin{aligned}
&\frac{|f_2(\mathbf{X}) - f_2(\mathbf{Y})|}{||\mathbf{X} - \mathbf{Y}||} \\
&= \frac{|\sum_{1 \leq i \leq n} ((\mathbf{X}_{i,i} - \rho_i)^+ - (\mathbf{Y}_{i,i} - \rho_i)^+)|}{||\mathbf{X} - \mathbf{Y}||} \\
&\leq \frac{\sum_{1 \leq i \leq n} |\mathbf{X}_{i,i} - \mathbf{Y}_{i,i}|}{||\mathbf{X} - \mathbf{Y}||} \\
&\leq 1
\end{aligned}$$

So $L_2 \leq 1$. □

Theorem 4.3. *$R \leq r \log n$.*

Proof. Choose $\mathbf{X}_1 = \frac{r}{n}\mathbf{I}$. Let $\mathbf{X}' = \frac{\mathbf{X}^*}{r}$. Then

$$\begin{aligned}
&D_\Phi(\mathbf{X}^*, \mathbf{X}_1) \\
&= \mathrm{tr}(\mathbf{X}^* \log \mathbf{X}^*) - \mathrm{tr}(\mathbf{X}^* \log \mathbf{X}_1) \\
&= \mathrm{tr}(r\mathbf{X}'(\log \mathbf{X}' + \log r\mathbf{I})) - r \log \frac{r}{n} \\
&= \mathrm{tr}(\mathbf{X}' \log \mathbf{X}') + r \log n \\
&\leq r \log n
\end{aligned}$$

□

Theorem 4.4. *Φ is $\frac{1}{2r}$-strongly convex over $\mathcal{X}$ with respect to the trace norm.*

Proof. Suppose $\mathbf{X}, \mathbf{Y} \in \mathcal{X}$. Let $\mathbf{X}' = \frac{\mathbf{X}}{r}$, $\mathbf{Y}' = \frac{\mathbf{Y}}{r}$. By theorem B.1 we know that

$$-\frac{\Phi(\mathbf{X}') - \Phi(\mathbf{Y}') - \nabla\Phi(\mathbf{X}') \cdot (\mathbf{X}' - \mathbf{Y}')}{\frac{1}{2}||\mathbf{X}' - \mathbf{Y}'||^2} \geq \frac{1}{2}.$$

Then

$$
\begin{aligned}
&- \frac{\Phi(\mathbf{X}) - \Phi(\mathbf{Y}) - \nabla\Phi(\mathbf{X}) \cdot (\mathbf{X} - \mathbf{Y})}{\frac{1}{2}||\mathbf{X} - \mathbf{Y}||^2} \\
&= -\operatorname{tr}((r\mathbf{X}')(\log \mathbf{X}' + \log r\mathbf{I}) - (r\mathbf{Y}')(\log \mathbf{Y}' + \log r\mathbf{I}) \\
&- (\mathbf{I} + \log \mathbf{X}' + \log r\mathbf{I})(r\mathbf{X}' - r\mathbf{Y}'))/(\frac{1}{2}r^2||\mathbf{X}' - \mathbf{Y}'||^2) \\
&= -\frac{\operatorname{tr}\left(r(\mathbf{X}' \log \mathbf{X}' - \mathbf{Y}' \log \mathbf{Y}' - (\mathbf{X}' - \mathbf{Y}') \log \mathbf{X}')\right)}{\frac{1}{2}r^2||\mathbf{X}' - \mathbf{Y}'||^2} \\
&= -\frac{1}{r}\frac{\Phi(\mathbf{X}') - \Phi(\mathbf{Y}') - \nabla\Phi(\mathbf{X}') \cdot (\mathbf{X}' - \mathbf{Y}')}{\frac{1}{2}||\mathbf{X}' - \mathbf{Y}'||^2} \\
&\geq \frac{1}{2r}
\end{aligned}
$$

□

Theorem 4.5. *In $\frac{2304 \log n}{\epsilon^2}$ iterations, the standard setup for mirror descent produces an ϵ-approximate solution.*

Proof. By theorem 2.1, theorem 4.1, theorem 4.2, theorem 4.3, theorem 4.4 and the fact that $r = 8m$. □

5 A Simpler Setup for SDP

Here we let the norm $||\cdot||$ be the Frobenius norm $||\cdot||_F$ defined as $||\mathbf{X}||_F = \sqrt{\operatorname{tr}(\mathbf{X}^2)}$. Let $\mathcal{D} = \operatorname{Sym}_n$, the space of $n \times n$ symmetric matrices. Let $\Phi(\mathbf{X}) = \frac{1}{2}||\mathbf{X}||^2$. For this choice of mirror map, the Bregman divergence is given by

$$
\begin{aligned}
D_\Phi(\mathbf{X}, \mathbf{Y}) &= \Phi(\mathbf{X}) - \Phi(\mathbf{Y}) - \nabla\Phi(\mathbf{Y}) \cdot (\mathbf{X} - \mathbf{Y}) \\
&= \frac{1}{2}\operatorname{tr}\mathbf{X}^2 - \frac{1}{2}\operatorname{tr}\mathbf{Y}^2 - \mathbf{Y} \cdot (\mathbf{X} - \mathbf{Y}) \\
&= \frac{1}{2}\operatorname{tr}(\mathbf{X} - \mathbf{Y})^2 \\
&= \Phi(\mathbf{X} - \mathbf{Y})
\end{aligned}
$$

Theorem 5.1. *Φ satisfies the required properties for a mirror map.*

Proof. Let $\mathbf{X}, \mathbf{Y} \in \mathcal{D}$.

(1): $\Phi(\mathbf{X}) - \Phi(\mathbf{Y}) - \nabla\Phi(\mathbf{X})^T(\mathbf{X} - \mathbf{Y}) = \frac{1}{2}\operatorname{tr}(\mathbf{X}^2) - \frac{1}{2}\operatorname{tr}(\mathbf{Y}^2) - \mathbf{X} \cdot (\mathbf{X} - \mathbf{Y}) = -\frac{1}{2}||\mathbf{X} - \mathbf{Y}||^2$. So Φ is 1-strongly convex. $\nabla\Phi(\mathbf{Y}) = \mathbf{Y}$. So Φ is differentiable.

(2): We can let $\mathbf{Y} = \mathbf{X} - \eta\nabla f(\mathbf{X})$.

(3): $\mathcal{D}$ is unbounded. As $\mathbf{X}$ approaches to infinity, $||\nabla\Phi(\mathbf{X})|| = ||\mathbf{X}||$ also approaches to infinity. □

Because $\nabla\Phi(\mathbf{M}) = \mathbf{M}$, our iterations are simply given by $\mathbf{Y}_{s+1} = \mathbf{X}_s - \eta\nabla f(\mathbf{X}_s)$.

We still need to understand how to project back on to the domain, after taking a gradient step. We present the projection algorithm below. Depending on whether there is the trace constraint $\operatorname{tr}\mathbf{X} = r$, we have two different projections.

5.1 Projection without Trace Constraint

In this case, $\mathcal{X} = \mathbb{S}^n$.

Algorithm 2 Calculate $\Pi^{\Phi}_{\mathcal{X}}(\mathbf{Y})$ where $\mathcal{X} = \mathbb{S}^n$

Input: $\mathbf{Y} \in \mathrm{Sym}_n$
Output: $\mathbf{X} = \Pi^{\Phi}_{\mathcal{X}}(\mathbf{Y})$
1: Write $\mathbf{Y}$ as $\mathbf{Y} = \mathbf{V}\mathbf{D}\mathbf{V}^{-1}$ where $\mathbf{V}$ is orthogonal and $\mathbf{D}$ is diagonal.
2: Let $\mathbf{C}$ be defined as $\mathbf{C}_{i,j} = \max\{\mathbf{D}_{i,j}, 0\}$ for $1 \le i, j \le n$.
3: **return** $\mathbf{V}\mathbf{C}\mathbf{V}^{-1}$

Theorem 5.2. *The above algorithm produces the correct projection.*

Proof. First, note that

$$||\mathbf{V}\mathbf{M}\mathbf{V}^{-1}||^2 = \operatorname{tr}(\mathbf{V}\mathbf{M}\mathbf{V}^{-1}\mathbf{V}\mathbf{M}^T\mathbf{V}^{-1}) = \operatorname{tr}(\mathbf{V}\mathbf{M}\mathbf{M}^T\mathbf{V}^{-1}) = \operatorname{tr}(\mathbf{M}^2) = ||\mathbf{M}||^2$$

which means that change of basis preserves Frobenius norm. Also, it preserves the domain $\mathcal{X}$. At this point we only need to calculate $\Pi^{\Phi}_{\mathcal{X}}(\mathbf{D})$.

Let $\mathbf{C} = \Pi^{\Phi}_{\mathcal{X}}(\mathbf{D})$. If $\mathbf{C}$ is not diagonal, let $\mathbf{C}'$ be diagonal of $\mathbf{C}$. Because $\mathbf{C} \in \mathcal{X} \subseteq \mathbb{S}^n$, entries of $\mathbf{C}'$ are all non-negative. Also, $\operatorname{tr}\mathbf{C}' = \operatorname{tr}\mathbf{C}$. So $\mathbf{C}' \in \mathcal{X}$. We have that

$$\Phi(\mathbf{C} - \mathbf{D}) = \frac{1}{2}\sum_{1\le i,j\le n} |\mathbf{C}_{i,j} - \mathbf{D}_{i,j}|^2 \ge \frac{1}{2}\sum_{1\le i\le n} |\mathbf{C}_{i,i} - \mathbf{D}_{i,i}|^2 = \Phi(\mathbf{C}' - \mathbf{D})$$

So $\mathbf{C}$ is diagonal.

For any diagonal positive semidefinite matrix $\mathbf{C}'$, we have $||\mathbf{C}' - \mathbf{D}||^2 = \sum_{1\le i\le n}(\mathbf{C}'_{i,i} - \mathbf{D}_{i,i})^2 \ge \sum_{1\le i\le n}(\max\{\mathbf{D}_{i,i}, 0\} - \mathbf{D}_{i,i})^2 = ||\mathbf{C} - \mathbf{D}||^2$. □

5.2 Projection with Trace Constraint

In this case, $\mathcal{X} = \{\mathbf{X} : \operatorname{tr}\mathbf{X} = r, \mathbf{X} \succeq \mathbf{0}\}$.

Theorem 5.3. *The above algorithm produces the correct projection.*

Proof. As in the proof of theorem 5.2, we only need to find projection of $\mathbf{D}$, and $\mathbf{C} = \Pi^{\Phi}_{\mathcal{X}}(\mathbf{D})$ must be diagonal.

Algorithm 3 Calculate $\Pi_{\mathcal{X}}^{\Phi}(\mathbf{Y})$ where $\mathcal{X} = \{\mathbf{X} : \operatorname{tr}\mathbf{X} = r, \mathbf{X} \succeq \mathbf{0}\}$

Input: $\mathbf{Y} \in \mathrm{Sym}_n$

Output: $\mathbf{X} = \Pi_{\mathcal{X}}^{\Phi}(\mathbf{Y})$

1: Write $\mathbf{Y}$ as $\mathbf{Y} = \mathbf{V}\mathbf{D}\mathbf{V}^{-1}$ where $\mathbf{V}$ is orthogonal and $\mathbf{D}$ is diagonal. W.l.o.g, suppose $\forall i, \mathbf{D}_{i,i} \leq \mathbf{D}_{i+1,i+1}$.

2: Choose the smallest k so that $r \geq \sum_{i=k}^{n}(\mathbf{D}_{i,i} - \mathbf{D}_{k,k})$.

3: Let $\mathbf{C}$ be defined as: $\mathbf{C}_{i,j} = 0$ for $i \neq j$, $\mathbf{C}_{i,i} = 0$ for $i < k$, $\mathbf{C}_{i,i} = \mathbf{D}_{i,i} + \frac{r - \sum_{i=k}^{n}\mathbf{D}_{i,i}}{n-k+1}$ for $i \geq k$.

4: **return** $\mathbf{V}\mathbf{C}\mathbf{V}^{-1}$

Suppose there exist $i < j$ such that $\mathbf{C}_{i,i} \neq 0$ and $\mathbf{D}_{i,i} - \mathbf{C}_{i,i} \neq \mathbf{D}_{j,j} - \mathbf{C}_{j,j}$. Let U be some neighbourhood of 0 in $\mathbb{R}$. Define $\mathbf{C}' : U \to \mathcal{X}$ as $\mathbf{C}'(x)$ differs with $\mathbf{C}$ only at $\mathbf{C}'(x)_{i,i} = \mathbf{C}_{i,i} + x$ and $\mathbf{C}'(x)_{j,j} = \mathbf{C}_{j,j} - x$. Then

$$\begin{aligned}
\frac{\partial\Phi(\mathbf{C}'(x) - \mathbf{D})}{\partial x}\Big|_{x=0} &= \frac{\partial\frac{1}{2}((\mathbf{C}_{i,i} + x - \mathbf{D}_{i,i})^2 + (\mathbf{C}_{j,j} - x - \mathbf{D}_{j,j})^2)}{\partial x}\Big|_{x=0} \\
&= \frac{\partial x(\mathbf{C}_{i,i} - \mathbf{D}_{i,i} - \mathbf{C}_{j,j} + \mathbf{D}_{j,j})}{\partial x}\Big|_{x=0} \\
&= \mathbf{C}_{i,i} - \mathbf{D}_{i,i} - \mathbf{C}_{j,j} + \mathbf{D}_{j,j} \neq 0
\end{aligned}$$

So $\mathbf{C}$ is not the best answer. So for all $i < j$, either $\mathbf{C}_{i,i} = 0$ or $\mathbf{D}_{i,i} - \mathbf{C}_{i,i} = \mathbf{D}_{j,j} - \mathbf{C}_{j,j}$.

Hence there exist k and C such that for $i < k$, $\mathbf{C}_{i,i} = 0$ and for $i \geq k$, $\mathbf{C}_{i,i} = \mathbf{D}_{i,i} + C$. Also, by the trace constraint, $C = \frac{r - \sum_{i=k}^{n}\mathbf{D}_{i,i}}{n-k+1}$. It is obvious that a choice of k is valid if and only if $\mathbf{C}_{k,k} \geq 0$, which is equivalent to $\mathbf{D}_{k,k} + \frac{r - \sum_{i=k}^{n}\mathbf{D}_{i,i}}{n-k+1} \geq 0$, which is equivalent to $r \geq \sum_{i=k}^{n}(\mathbf{D}_{i,i} - \mathbf{D}_{k,k})$. $\sum_{i=k}^{n}(\mathbf{D}_{i,i} - \mathbf{D}_{k,k})$ is non-increasing with respect to k. So if k satisfies the property, so does $k+1$.

Let k satisfy the property, and $k' = k + 1$. Let $\mathbf{C}$ be the matrix corresponding to k and $\mathbf{C}'$ be the matrix corresponding to k'. For the simplicity of proof, let $S = \sum_{i=k}^{n}\mathbf{D}_{i,i}$.

Then

$$
\begin{aligned}
&2\left(\Phi\left(\mathbf{C}'-\mathbf{D}\right)-\Phi\left(\mathbf{C}-\mathbf{D}\right)\right)\\
&=\sum_{i=1}^{k'-1}\mathbf{D}_{i,i}^2+\frac{(r-\sum_{i=k'}^{n}\mathbf{D}_{i,i})^2}{n-k'+1}-\sum_{i=1}^{k-1}\mathbf{D}_{i,i}^2-\frac{(r-\sum_{i=k}^{n}\mathbf{D}_{i,i})^2}{n-k+1}\\
&=\mathbf{D}_{k,k}^2+\frac{\left(r-\sum_{i=k'}^{n}\mathbf{D}_{i,i}\right)^2}{n-k'+1}-\frac{\left(r-\sum_{i=k}^{n}\mathbf{D}_{i,i}\right)^2}{n-k+1}\\
&=\mathbf{D}_{k,k}^2+\frac{\left(r-S+\mathbf{D}_{k,k}\right)^2}{n-k'+1}-\frac{\left(r-S\right)^2}{n-k+1}\\
&=\mathbf{D}_{k,k}^2+\frac{(r-S)^2}{n-k}+\frac{2\mathbf{D}_{k,k}(r-S)}{n-k}+\frac{\mathbf{D}_{k,k}^2}{n-k}-\frac{(r-S)^2}{n-k+1}\\
&=\frac{n-k+1}{n-k}\mathbf{D}_{k,k}^2+\frac{\left(r-S\right)^2}{(n-k)\left(n-k+1\right)}+\frac{2\mathbf{D}_{k,k}\left(r-S\right)}{n-k}\\
&\geq 0
\end{aligned}
$$

So the smallest k must be the best choice. □

6 Analysis of the Simple Setup for Mirror Descent on Example Problems

In this section we analyze the number of iterations needed for mirror descent with the simple setup to get an ϵ-approximation on example problems.

6.1 MAX-CUT SDP

Define $f(\mathbf{X}) = -\sqrt{\text{DIAG}(\rho^{-1})}\mathbf{C}\sqrt{\text{DIAG}(\rho^{-1})}\cdot\mathbf{X}+\sum_{i=1}^{n}(\mathbf{X}_{ii}-\rho_i)^+$. Let $r=\sum_{1\leq i\leq n}\rho_i$. Let $\mathcal{X}=\mathbb{S}^n$. Note that there is not trace constraint here.

We would like to solve

$$\text{Minimize } f(\mathbf{X}) \text{ subject to } \mathbf{X}\in\mathcal{X}.$$

Let $\mathbf{X}^*=\arg\min_{\mathbf{X}\in\mathcal{X}}f(\mathbf{X})$. Let R, L, α be defined as in section 2.4. By theorem 2.1, we need $t=\frac{2RL^2}{\alpha f(\mathbf{X}^*)^2}\epsilon^{-2}$ iterations to find an ϵ-approximate solution.

We have bounded $f(\mathbf{X}^*)$ in section 4.1. Then we bound L, R and α.

Theorem 6.1. *f is $\frac{3}{2}\sqrt{n}$-Lipschitz with respect to the Frobenius norm.*

Proof. We first prove that for $\mathbf{X}\in\mathbb{S}^n$, we have $||\mathbf{X}||_1\leq\sqrt{n}||\mathbf{X}||_F$, where $||\cdot||_1$ is the trace norm and $||\cdot||_F$ is the Frobenius norm.

Let $\mathbf{X}=\mathbf{V}\mathbf{Y}\mathbf{V}^{-1}$, where $\mathbf{Y}$ is diagonal and $\mathbf{V}$ is orthogonal. The diagonal entries of $\mathbf{Y}$ are the eigenvalues of $\mathbf{X}$.

$$||\mathbf{X}||_F = \sqrt{\sum_{i=1}^{n} |\mathbf{Y}_{i,i}^2|} \geq \sqrt{\frac{1}{n}(\sum |\mathbf{Y}_{i,i}|)^2} = \sqrt{\frac{1}{n}} \sum_{i=1}^{n} |\mathbf{Y}_{i,i}| = \sqrt{\frac{1}{n}} ||\mathbf{X}||_1$$

According to theorem 4.2,

$$\forall \mathbf{X}, \mathbf{Y} \in \mathcal{D},\ f(\mathbf{X}) - f(\mathbf{Y}) \leq \frac{3}{2}||\mathbf{X} - \mathbf{Y}||_1 \leq \frac{3}{2}\sqrt{n}||\mathbf{X} - \mathbf{Y}||_F.$$

So f is $\frac{3}{2}\sqrt{n}$-Lipschitz with respect to the Frobenius norm. □

Theorem 6.2. $R \leq \frac{1}{2}(1 - \frac{1}{n})r^2$.

Proof. We first prove that for all $1 \leq i, j \leq n$, $|\mathbf{X}^*_{i,j}| \leq \sqrt{\mathbf{X}^*_{i,i}\mathbf{X}^*_{j,j}}$. Suppose for some i, j, $|\mathbf{X}^*_{i,j}| > \sqrt{\mathbf{X}^*_{i,i}\mathbf{X}^*_{j,j}}$. Obviously $i \neq j$.

Suppose $\mathbf{X}^*_{i,j} < 0$. We define a vector $\mathbf{v}$ as $\mathbf{v}_k = 0$ for $k \neq i, j$, $\mathbf{v}_i = \sqrt{\mathbf{X}^*_{j,j}}$ and $\mathbf{v}_j = \sqrt{\mathbf{X}^*_{i,i}}$. Then

$$\begin{aligned}
&\mathbf{v}^T\mathbf{X}^*\mathbf{v} \\
&= \mathbf{v}_i^2\mathbf{X}^*_{i,i} + \mathbf{v}_j^2\mathbf{X}^*_{j,j} + 2\mathbf{v}_i\mathbf{v}_j\mathbf{X}^*_{i,j} \\
&< \mathbf{v}_i^2\mathbf{X}^*_{i,i} + \mathbf{v}_j^2\mathbf{X}^*_{j,j} - 2\mathbf{v}_i\mathbf{v}_j\sqrt{\mathbf{X}^*_{i,i}\mathbf{X}^*_{j,j}} \\
&= (\mathbf{v}_i\sqrt{\mathbf{X}^*_{i,i}} - \mathbf{v}_j\sqrt{\mathbf{X}^*_{j,j}})^2 = 0
\end{aligned}$$

Suppose $\mathbf{X}^*_{i,j} > 0$. We define a vector $\mathbf{v}$ as $\mathbf{v}_k = 0$ for $k \neq i, j$, $\mathbf{v}_i = \sqrt{\mathbf{X}^*_{j,j}}$ and $\mathbf{v}_j = -\sqrt{\mathbf{X}^*_{i,i}}$. Then

$$\begin{aligned}
&\mathbf{v}^T\mathbf{X}^*\mathbf{v} \\
&= \mathbf{v}_i^2\mathbf{X}^*_{i,i} + \mathbf{v}_j^2\mathbf{X}^*_{j,j} + 2\mathbf{v}_i\mathbf{v}_j\mathbf{X}^*_{i,j} \\
&< \mathbf{v}_i^2\mathbf{X}^*_{i,i} + \mathbf{v}_j^2\mathbf{X}^*_{j,j} + 2\mathbf{v}_i\mathbf{v}_j\sqrt{\mathbf{X}^*_{i,i}\mathbf{X}^*_{j,j}} \\
&= (\mathbf{v}_i\sqrt{\mathbf{X}^*_{i,i}} + \mathbf{v}_j\sqrt{\mathbf{X}^*_{j,j}})^2 = 0
\end{aligned}$$

Both cases contradict the positive-semidefiniteness of $\mathbf{X}^*$. So we must have $|\mathbf{X}^*_{i,j}| \leq \sqrt{\mathbf{X}^*_{i,i}\mathbf{X}^*_{j,j}}$ for all $1 \leq i, j \leq n$.

By theorem 3.1, $\textsc{diag}(\mathbf{X}^*) = \rho$. We choose the initial point $\mathbf{X}_1$ to be the diagonal

matrix with diagonal equal to ρ. Then

$$\begin{aligned}
D_\Phi(\mathbf{X}^*, \mathbf{X}_1) &= \frac{1}{2}||\mathbf{X}^* - \mathbf{X}_1||^2 \\
&= \frac{1}{2} \sum_{1\le i,j\le n, i\neq j} |\mathbf{X}^*_{i,j}|^2 \\
&\le \frac{1}{2} \sum_{1\le i,j\le n, i\neq j} \mathbf{X}^*_{i,i}\mathbf{X}^*_{j,j} \\
&= \frac{1}{2} \sum_{1\le i,j\le n, i\neq j} \rho_i\rho_j \\
&= \frac{1}{2}((\sum_{1\le i\le n} \rho_i)^2 - \sum_{1\le i\le n} \rho_i^2) \\
&\le \frac{1}{2}(r^2 - \frac{r^2}{n}) \\
&= \frac{1}{2}(1 - \frac{1}{n})r^2
\end{aligned}$$

□

Theorem 6.3. *Φ is 1-strongly convex over $\mathcal{X}$ with respect to the Frobenius norm.*

We have proved this in theorem 5.1.

Theorem 6.4. *In $\frac{576n}{\epsilon^2}$ iterations, the simple setup for mirror descent produces an ϵ-approximate solution.*

Proof. By theorem 2.1, theorem 4.1, theorem 6.1, theorem 6.2, theorem 6.3 and the fact that $r = 8m$. □

7 Discussion

For now, we have two essentially different setups for the max-cut SDP. For the standard setup, we need $O(\log n/\epsilon^2)$ iterations to get an $\epsilon-$approximation. In each iteration, the most expensive step is calculating a matrix exponential. For the simpler setup, we need $o(n/\epsilon^2)$ iterations to get an $\epsilon-$approximation. In each iteration, we need to diagonalize a matrix which is even more time consuming then computing matrix exponentials. In experiment, both algorithms produce satisfying results while the standard setup indeed is practically faster. However, it seems that the simpler setup can be improved in some ways. Generally speaking, for simpler setup in each iteration, the gradient step is very easy to do (simply by adding a matrix) but the projection step is harder. In order to project the resulting matrix back to the space of positive semidefinite matrices, we use diagonalization to eliminate negative eigenvalues. One thought is finding a polynomial

$p(x)$ that approximates $|x|$ really well. If we get such polynomial p, $\frac{1}{2}(p(\mathbf{X}) + \mathbf{X})$ will be a reasonably good projection of $\mathbf{X}$. Calculating a polynomial is significantly faster than calculating exponentials. Another thought is adjusting step lengths. The general idea of Nesterov's method may apply here.

We are also trying new mirror maps to do mirror descent. For example, we intend to use a polynomial to replace the standard mirror map $\operatorname{tr}(\mathbf{X} \log \mathbf{X})$.

Since what we have now is already good in practical purposes, we are trying to apply the method to other SDP problems, such as Lovász Theta Function.

Acknowledgements

This project was done in the Summer Program in Undergraduate Research (SPUR) of the Massachusetts Institute of Technology Mathematics Department. We thank Adrian Vladu for being our mentor and helping us throughout the project. We thank Richard Peng for suggesting this project and giving valuable suggestions. We thank Pavel Etingof and David Jerison for giving valuable suggestions. We thank Slava Gerovitch for organizing the SPUR program.

References

[AHK12] Sanjeev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(6):121–164, 2012. 4

[AK07] Sanjeev Arora and Satyen Kale. A combinatorial, primal-dual approach to semidefinite programs. In *Proceedings of the Thirty-ninth Annual ACM Symposium on Theory of Computing*, STOC '07, pages 227–236, New York, NY, USA, 2007. ACM. 2, 4

[AZO14] Zeyuan Allen-Zhu and Lorenzo Orecchia. A novel, simple interpretation of nesterov's accelerated method as a combination of gradient and mirror descent. 2014. 4

[BC06] Tijl De Bie and Nello Cristianini. Fast sdp relaxations of graph cut clustering, transduction, and other combinatorial problems. *J. Mach. Learn. Res.*, 7:1409–1436, December 2006. 3

[Bub14] Sébastien Bubeck. Theory of Convex Optimization for Machine Learning. *ArXiv e-prints*, May 2014. 4

[Chu97] Fan RK Chung. *Spectral graph theory*, volume 92. American Mathematical Soc., 1997. 10

[DS08] Samuel I. Daitch and Daniel A. Spielman. Faster approximate lossy generalized flow via interior point algorithms. *CoRR*, abs/0803.0988, 2008. 2

[GW95] Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42(6):1115–1145, 1995. 2, 3

[Kar72] Richard M. Karp. Reducibility among combinatorial problems. In Raymond E. Miller and James W. Thatcher, editors, *Complexity of Computer Computations*, The IBM Research Symposia Series, pages 85–103. Plenum Press, New York, 1972. 3

[Kle31] Otto Klein. Zur quantenmechanischen begründung des zweiten hauptsatzes der wärmelehre. *Zeitschrift für Physik*, 72(11-12):767–775, 1931. 21

[KSST09] Sham M. Kakade, Shai Shalev-Shwartz, and Ambuj Tewari. On the duality of strong convexity and strong smoothness: Learning applications and matrix regularization, 2009. 20

[Lee14] Yin Tat Lee. Experiment on mmwu for max cut, 2014. 2, 4, 6

[NY83] Arkadi Semenovich Nemirovsky and David Borisovich Yudin. *Problem complexity and method efficiency in optimization*. Wiley (Chichester and New York), 1983. 2, 4

[PST95] Serge A. Plotkin, David B. Shmoys, and Eva Tardos. Fast approximation algorithms for fractional packing and covering problems. *Mathematics of Operations Research*, 20:257–301, 1995. 4

[SG76] Sartaj Sahni and Teofilo Gonzalez. P-complete approximation problems. *Journal of the ACM (JACM)*, 23(3):555–565, 1976. 8

[WFLM13] Sida Wang, Roy Frostig, Percy Liang, and Christopher D. Manning. Relaxations for inference in restricted boltzmann machines. 2013. 3

A Analysis of Mirror Descent

First we have a theorem regarding the Bregman divergence.

Theorem A.1. $D_\Phi(x, y) \geq D_\Phi(x, \Pi^\Phi_{\mathcal{X}}(y)) + D_\Phi(\Pi^\Phi_{\mathcal{X}}(y), y)$.

Proof. Define $g_y(x) = \Phi(x) - \Phi(y) - \nabla\Phi(y)^T(x - y)$. Then g is a convex function. By the first order optimality condition and the definition of $\Pi^\Phi_{\mathcal{X}}(y)$, we have

$$\nabla g(\Pi^\Phi_{\mathcal{X}}(y))^T(\Pi^\Phi_{\mathcal{X}}(y) - x) \leq 0, \forall x \in \mathcal{X}.$$

Substituting in the gradient of g, $\nabla g_y(x) = \nabla\Phi(x) - \nabla\Phi(y)$, gives:

$$(\nabla\Phi(\Pi^\Phi_{\mathcal{X}}(y)) - \nabla\Phi(y))^T(\Pi^\Phi_{\mathcal{X}}(y) - x) \leq 0, \forall x \in \mathcal{X}$$

Equivalently, by moving terms around, this can be rewritten as

$$\begin{aligned}-\nabla\Phi(y)^T(x-y) \geq &-\Phi(\Pi^\Phi_{\mathcal{X}}(y)) - \nabla\Phi(\Pi^\Phi_{\mathcal{X}}(y))^T(x - \Pi^\Phi_{\mathcal{X}}(y)) + \\ &\Phi(\Pi^\Phi_{\mathcal{X}}(y)) - \nabla\Phi(y)^T(\Pi^\Phi_{\mathcal{X}}(y) - y)\end{aligned}$$

Adding $\Phi(x) - \Phi(y)$ to both sides yields

$$\begin{aligned}&\Phi(x) - \Phi(y) - \nabla\Phi(y)^T(x-y) \geq \\ &\quad \Phi(x) - \Phi(\Pi^\Phi_{\mathcal{X}}(y)) - \nabla\Phi(\Pi^\Phi_{\mathcal{X}}(y))^T(x - \Pi^\Phi_{\mathcal{X}}(y)) + \\ &\quad \Phi(\Pi^\Phi_{\mathcal{X}}(y)) - \Phi(y) - \nabla\Phi(y)^T(\Pi^\Phi_{\mathcal{X}}(y) - y)\end{aligned}$$

which gives the desired inequality. □

Let $x^* = \arg\min_{x\in\mathcal{X}\cap\mathcal{D}} f(x)$. Then

$$\begin{aligned}&f(x_s) - f(x^*) \\ &\leq \nabla f(x_s)^T(x_s - x^*) \\ &= \frac{1}{\eta}(\nabla\Phi(x_s) - \nabla\Phi(y_{s+1}))^T(x_s - x^*) \\ &= \frac{1}{\eta}(D_\Phi(x^*, x_s) + D_\Phi(x_s, y_{s+1}) - D_\Phi(x^*, y_{s+1})) \\ &\leq \frac{1}{\eta}(D_\Phi(x^*, x_s) + D_\Phi(x_s, y_{s+1}) - D_\Phi(x^*, x_{s+1}) - D_\Phi(x_{s+1}, y_{s+1}))\end{aligned}$$

We notice that the terms $D_\Phi(x^*, x_s) - D_\Phi(x^*, x_{s+1})$ will telescope when we sum over all these inequalities from $s = 1$ to $s = t$.

Then we bound the other terms.

$$\begin{aligned}&D_\Phi(x_s, y_{s+1}) - D_\Phi(x_{s+1}, y_{s+1}) \\ &= \Phi(x_s) - \Phi(x_{s+1}) - \nabla\Phi(y_{s+1})^T(x_s - x_{s+1}) \\ &\leq (\nabla\Phi(x_s) - \nabla\Phi(y_{s+1}))^T(x_s - x_{s+1}) - \frac{\alpha}{2}||x_s - x_{s+1}||^2 \\ &= \eta\nabla f(x_s)^T(x_s - x_{s+1}) - \frac{\alpha}{2}||x_s - x_{s+1}||^2 \\ &\leq \eta L||x_{s+1} - x_s|| - \frac{\alpha}{2}||x_{s+1} - x_s||^2 \\ &\leq \frac{\eta^2 L^2}{2\alpha}\end{aligned}$$

in which the first inequality comes from the α-strong-convexity of Φ and the second inequality above comes from the assumption that f is $L-$Lipschitz.

Summing up all inequalities from $s = 1$ to $s = t$, we get

$$\begin{aligned}
&f\left(\frac{1}{t}\sum_{s=1}^{t} x_s\right) - f\left(x^*\right) \\
&\leq \frac{1}{t}\sum_{s=1}^{t} f\left(x_s\right) - f\left(x^*\right) \\
&\leq \frac{D_\Phi(x^*, x_1) - D_\Phi(x^*, x_{t+1})}{\eta t} + \frac{\eta L^2}{2\alpha} \\
&= \frac{R}{\eta t} + \frac{\eta L^2}{2\alpha}
\end{aligned}$$

Let $\eta = \sqrt{\frac{2\alpha R}{L^2 t}}$ and we get $f(\frac{1}{t}\sum_{s=1}^{t} x_s) - f(x^*) \leq \sqrt{\frac{2RL^2}{\alpha t}}$.

B Standard Setup for Mirror Descent on SDP

Suppose we are optimizing a convex function f over $\mathcal{X} = \{\mathbf{X} | \mathbf{X} \succeq \mathbf{0}, \mathrm{tr}(\mathbf{X}) = 1\}$. We let the norm $||\cdot||$ be the trace norm $||\cdot||_1$. Namely, $||\mathbf{A}||_1 = \mathrm{tr}(\sqrt{\mathbf{A}^T\mathbf{A}})$. Let $\Phi(x) = \mathrm{tr}(\mathbf{X}\log\mathbf{X})$, $\mathcal{D} = \mathbb{S}^n$.

Theorem B.1. *Φ satisfies the required properties.*

Proof. Let $\mathbf{X}, \mathbf{Y} \in \mathcal{D}$.

(1): According to [KSST09], Φ is $\frac{1}{2}-$strongly convex and differentiable.

(2): $\nabla\Phi(\mathbf{X}) = \mathbf{I} + \log\mathbf{X}$. For any $\mathbf{X} \in \mathcal{D}, \eta > 0$, $\mathbf{Y} = \exp(\log\mathbf{X} - \eta\nabla f(\mathbf{X}))$ satisfies $\nabla\Phi(\mathbf{Y}) = \nabla\Phi(\mathbf{X}) - \eta\nabla f(\mathbf{X})$.

(3): The boundary of $\mathcal{D}$ is matrices with eigenvalue 0. Thus when $\mathbf{X}$ approaches the boundary, $||\nabla\Phi(\mathbf{X})|| = ||\mathbf{I} + \log\mathbf{X}||$ approaches infinity. □

Theorem B.2. $\Pi_{\mathcal{X}}^{\Phi}(\mathbf{Y}) = \dfrac{\mathbf{Y}}{\mathrm{tr}(\mathbf{Y})}$.

Proof. Let $\mathbf{Z} = \frac{\mathbf{Y}}{\mathrm{tr}(\mathbf{Y})}$, which is a positive semidefinite symmetric matrix with trace 1.

$$\begin{aligned}
\Pi_{\mathcal{X}}^{\Phi}(\mathbf{Y}) &= \arg\min_{\mathbf{X}\in\mathcal{X}\cap\mathcal{D}} D_{\Phi}(\mathbf{X},\mathbf{Y}) \\
&= \arg\min_{\mathbf{X}\in\mathcal{X}\cap\mathcal{D}} (\Phi(\mathbf{X}) - \Phi(\mathbf{Y}) - \nabla\Phi(\mathbf{Y})^T(\mathbf{X}-\mathbf{Y})) \\
&= \arg\min_{\mathbf{X}\in\mathcal{X}\cap\mathcal{D}} (\Phi(\mathbf{X}) - \nabla\Phi(\mathbf{Y})^T\mathbf{X}) \\
&= \arg\min_{\mathbf{X}\in\mathcal{X}\cap\mathcal{D}} (\mathrm{tr}(\mathbf{X}\log\mathbf{X}) - \mathrm{tr}(\mathbf{X}) - \mathrm{tr}(\mathbf{X}\log\mathbf{Y})) \\
&= \arg\min_{\mathbf{X}\in\mathcal{X}\cap\mathcal{D}} \mathrm{tr}(\mathbf{X}\log\mathbf{X} - \mathbf{X}\log\mathbf{Y}) \\
&= \arg\min_{\mathbf{X}\in\mathcal{X}\cap\mathcal{D}} \mathrm{tr}(\mathbf{X}\log\mathbf{X} - \mathbf{X}(\log\mathbf{Z} + \log\mathrm{tr}(\mathbf{Y}))) \\
&= \arg\min_{\mathbf{X}\in\mathcal{X}\cap\mathcal{D}} (\mathrm{tr}(\mathbf{X}\log\mathbf{X} - \mathbf{X}\log\mathbf{Z}) - \log\mathrm{tr}(\mathbf{Y})) \\
&= \arg\min_{\mathbf{X}\in\mathcal{X}\cap\mathcal{D}} \mathrm{tr}(\mathbf{X}\log\mathbf{X} - \mathbf{X}\log\mathbf{Z}) \\
&= \mathbf{Z}
\end{aligned}$$

where the last step comes from the Klein's inequality [Kle31].

□