CS395T: Continuous Algorithms, Part VII
Matrix multiplicative weights

Kevin Tian

1 Quantum entropy

In this lecture, we develop the *matrix multiplicative weights* (MMW) algorithm, a matrix generalization of the multiplicative weights algorithm in Section 4, Part III, proposed by [TRW05, KW07, AK07]. Just as multiplicative weights is a framework for regret minimization when the action set is the probability simplex $\Delta^d := \{x \in \mathbb{R}^d_{>0} \mid \|x\|_1 = 1\}$, MMW bounds regret over the *spectraplex*,

$$\Delta^{d\times d} := \left\{\mathbf{X} \in \mathbb{S}^{d\times d}_{\succ \mathbf{0}} \mid \mathrm{Tr}\mathbf{X} = 1\right\}. \tag{1}$$

In other words, $\Delta^{d\times d}$ is the set of all matrices expressible as $\mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$, where $\mathbf{U} \in \mathbb{R}^{d\times d}$ is unitary and diagonal $\mathbf{\Lambda}$ has elements in Δ^d. By equivalently writing such an $\mathbf{X} \in \Delta^{d\times d}$ as

$$\mathbf{X} = \sum_{i\in[d]} \lambda_i u_i u_i^\top, \text{ where } \lambda \in \Delta^d \text{ and } \{u_i\}_{i\in[d]} \text{ are columns of } \mathbf{U},$$

we can instead interpret $\mathbf{X}$ as given by a probability distribution over the rank-one matrices $u_i u_i^\top$. For the rest of the lecture, we use the notation $\mathcal{X} := \Delta^{d\times d}$ to denote this action set, unless specified otherwise. To apply the regret minimization techniques of Part III, we first must provide a strongly convex regularizer in an appropriate norm over $\mathcal{X}$. We choose $\|\cdot\| = \|\cdot\|_1$, the Schatten-1 norm, which is just the trace when the argument is in $\mathbb{S}^{d\times d}_{\succeq \mathbf{0}}$. We also use the regularizer:

$$\varphi(\mathbf{X}) := \langle \mathbf{X}, \log(\mathbf{X})\rangle = \sum_{i\in[d]} \lambda_i(\mathbf{X}) \log \lambda_i(\mathbf{X}), \tag{2}$$

where $\log(\mathbf{X})$ is defined as in Section 2.1, Part V. This is the spectral analog of the entropy regularizer used in Section 4, Part III, and is hence amenable to our analysis techniques from Part V. The function (2) is often called the *quantum entropy* or *von Neumann entropy* in the literature, due to its fundamental role in the study of quantum information theory [NC10].

We begin by establishing regularity properties, e.g. strong convexity, of φ. It is helpful to first compute the dual of φ. Because entropy is permutation-invariant, Lewis's theorem (Theorem 8, Part V) tells us that φ^* agrees with its vector definition applied to the spectrum. We computed the dual of vector entropy in Lemma 8, Part III, such that

$$\varphi^*(\mathbf{Y}) := \log\left(\sum_{i\in[d]} \exp\left(\lambda_i(\mathbf{Y})\right)\right) = \log\left(\mathrm{Tr}\exp(\mathbf{Y})\right), \text{ for all } \mathbf{Y} \in \mathbb{S}^{d\times d}. \tag{3}$$

By applying Lewis's theorem again, we can also check that (paralleling Lemma 8, Part III):

$$\nabla\varphi^*(\mathbf{Y}) = \frac{\exp(\mathbf{Y})}{\mathrm{Tr}\exp(\mathbf{Y})} \in \mathcal{X} = \Delta^{d\times d}. \tag{4}$$

It is convenient to first prove smoothness of φ^*, because along the way, we will establish a tighter characterization of the Hessian of φ^*, which is useful in several applications.

Lemma 1. *Define φ^* as in (3). Then for all $\mathbf{Y}, \mathbf{M} \in \mathbb{S}^{d\times d}$,*

$$\nabla^2\varphi^*(\mathbf{Y})[\mathbf{M}, \mathbf{M}] \le \left\langle \nabla\varphi^*(\mathbf{Y}), \mathbf{M}^2\right\rangle \le \|\mathbf{M}\|^2_{\mathrm{op}}.$$

Proof. We give a proof using the Lewis-Sendov formula (Theorem 9, Part V).[1] Throughout, let $\mathbf{X} := \nabla\varphi^*(\mathbf{Y}) \in \mathcal{X}$ (see (4)), let $\lambda \in \mathbb{R}^d$ be the vector of eigenvalues of $\mathbf{Y}$, and let the eigendecomposition of $\mathbf{Y}$ be $\mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$ where $\mathbf{\Lambda} = \mathbf{diag}\,(\lambda)$. For $y \in \mathbb{R}^d$, let $\varphi^*_{\mathrm{vec}}(y) := \log(\sum_{i\in[d]} \exp(y_i))$ be the vector dual of entropy, as in Lemma 8, Part III. The Lewis-Sendov formula states that

$$\begin{aligned}
&\nabla^2 r^*(\mathbf{Y})[\mathbf{M},\mathbf{M}] = \nabla^2\varphi^*_{\mathrm{vec}}(\lambda)\left[\mathrm{diagvec}\left(\widetilde{\mathbf{M}}\right), \mathrm{diagvec}\left(\widetilde{\mathbf{M}}\right)\right] + \langle \mathbf{A}, \mathbf{M}\circ\mathbf{M}\rangle\,, \\
&\text{where } \mathbf{A}_{ij} := \begin{cases} 0 & i = j \\ \nabla^2_{ii}\varphi^*_{\mathrm{vec}}(\lambda) - \nabla^2_{ij}\varphi^*_{\mathrm{vec}}(\lambda) & i \neq j,\ \lambda_i = \lambda_j \ \text{ for } (i,j) \in [d]\times[d], \\ \frac{\nabla_i\varphi^*_{\mathrm{vec}}(\lambda) - \nabla_j\varphi^*_{\mathrm{vec}}(\lambda)}{\lambda_i - \lambda_j} & i \neq j,\ \lambda_i \neq \lambda_j \end{cases}
\end{aligned} \tag{5}$$

and $\widetilde{\mathbf{M}} := \mathbf{U}^\top\mathbf{M}\mathbf{U}$. Here, diagvec is the function which takes $\mathbf{M} \in \mathbb{S}^{d\times d}$ and returns the diagonal elements of $\mathbf{M}$ as a vector in $\mathbb{R}^d$, and $\circ$ denotes the Hadmard (entrywise) product. Now, let $x := \nabla\varphi^*_{\mathrm{vec}}(\lambda)$ and $\mathbf{H} := \nabla^2\varphi^*_{\mathrm{vec}}(\lambda)$. A straightforward computation yields

$$x = \frac{\exp(\lambda)}{N(\lambda)},\ \mathbf{H} = \mathbf{diag}\,(x) - xx^\top \preceq \mathbf{diag}\,(x)\,,\ \text{for } N(\lambda) := \sum_{i\in[d]}\exp(\lambda_i),$$

where $\exp(\lambda)$ is entrywise. Therefore, we can bound the first term in (5):

$$\begin{aligned}
\nabla^2\varphi^*_{\mathrm{vec}}(\lambda)\left[\mathrm{diagvec}\left(\widetilde{\mathbf{M}}\right), \mathrm{diagvec}\left(\widetilde{\mathbf{M}}\right)\right] &\leq \mathbf{diag}\,(x)\left[\mathrm{diagvec}\left(\widetilde{\mathbf{M}}\right), \mathrm{diagvec}\left(\widetilde{\mathbf{M}}\right)\right] \\
&= \frac{1}{N(\lambda)}\sum_{i\in[d]}\exp(\lambda_i)\widetilde{\mathbf{M}}^2_{ii}.
\end{aligned} \tag{6}$$

We proceed to the second term in (5). By applying the inequality $\frac{\exp(a)-\exp(b)}{a-b} \leq \frac{\exp(a)+\exp(b)}{2}$, which holds for all $a, b \in \mathbb{R}$, we have for all $i \neq j \in [d]$,

$$\begin{aligned}
\mathbf{A}_{ij}\widetilde{\mathbf{M}}^2_{ij} &= (\mathbf{H}_{ii} - \mathbf{H}_{ij})\,\widetilde{\mathbf{M}}^2_{ij} = (x_i - 2x_i^2)\widetilde{\mathbf{M}}^2_{ij} \\
&\leq x_i\widetilde{\mathbf{M}}^2_{ij} = \frac{\exp(\lambda_i)+\exp(\lambda_j)}{2N(\lambda)}\widetilde{\mathbf{M}}^2_{ij},\ \text{if } \lambda_i = \lambda_j, \\
\mathbf{A}_{ij}\widetilde{\mathbf{M}}^2_{ij} &= \frac{x_i - x_j}{\lambda_i - \lambda_j}\widetilde{\mathbf{M}}^2_{ij} = \frac{1}{N(\lambda)}\left(\frac{\exp(\lambda_i)-\exp(\lambda_j)}{\lambda_i-\lambda_j}\right)\widetilde{\mathbf{M}}^2_{ij} \\
&\leq \frac{\exp(\lambda_i)+\exp(\lambda_j)}{2N(\lambda)}\widetilde{\mathbf{M}}^2_{ij},\ \text{if } \lambda_i \neq \lambda_j.
\end{aligned} \tag{7}$$

Combining (6) and (7), we have the conclusion:

$$\begin{aligned}
\nabla^2 r^*(\mathbf{Y})[\mathbf{M},\mathbf{M}] &\leq \frac{1}{N(\lambda)}\sum_{(i,j)\in[d]\times[d]}\frac{\exp(\lambda_i)+\exp(\lambda_j)}{2}\widetilde{\mathbf{M}}^2_{ij} \\
&= \frac{1}{N(\lambda)}\sum_{(i,j)\in[d]\times[d]}\exp(\lambda_i)\widetilde{\mathbf{M}}^2_{ij} \\
&= \frac{1}{N(\lambda)}\sum_{i\in[d]}\exp(\lambda_i)\sum_{j\in[d]}\widetilde{\mathbf{M}}^2_{ij} = \frac{1}{N(\lambda)}\sum_{i\in[d]}\exp(\lambda_i)[\widetilde{\mathbf{M}}^2]_{ii} = \left\langle \mathbf{diag}\,(x)\,, \widetilde{\mathbf{M}}^2\right\rangle.
\end{aligned}$$

In the second line, we used that $\widetilde{\mathbf{M}}$ is symmetric. The first conclusion follows from

$$\left\langle \mathbf{diag}\,(x)\,, \widetilde{\mathbf{M}}^2\right\rangle = \left\langle \mathbf{U}\mathbf{diag}\,(x)\,\mathbf{U}^\top, \mathbf{M}^2\right\rangle = \left\langle \nabla\varphi^*(\mathbf{Y}), \mathbf{M}^2\right\rangle,$$

where we applied Lewis's theorem (Theorem 8, Part V) to show that $\mathbf{U}\mathbf{diag}\,(x)\,\mathbf{U}^\top = \nabla\varphi^*(\mathbf{Y})$. The second conclusion then follows from the matrix Hölder inequality:

$$\left\langle \nabla\varphi^*(\mathbf{Y}), \mathbf{M}^2\right\rangle \leq \|\nabla\varphi^*(\mathbf{Y})\|_{\mathrm{tr}}\left\|\mathbf{M}^2\right\|_{\mathrm{op}} = \left\|\mathbf{M}^2\right\|_{\mathrm{op}} = \|\mathbf{M}\|^2_{\mathrm{op}}.$$

□

[1] This is in part to give an example of how to apply this formula, to make it seem slightly less scary.

Remark 1. *In the special case of quantum entropy, there is an alternative proof of Lemma 1 that does not go through the Lewis-Sendov formula, due to [Nes07], which we reproduce. Using (4),*

$$\langle \nabla \varphi^*(\mathbf{Y}), \mathbf{M} \rangle = \frac{\langle \mathbf{M}, \exp(\mathbf{Y}) \rangle}{\operatorname{Tr} \exp(\mathbf{Y})},$$

so that

$$\begin{aligned} \nabla^2 \varphi^*(\mathbf{Y})[\mathbf{M}, \mathbf{M}] &= \left\langle \mathbf{M}, \nabla \left(\frac{\langle \mathbf{M}, \exp(\mathbf{Y}) \rangle}{\operatorname{Tr} \exp(\mathbf{Y})} \right) \right\rangle \\ &= \left\langle \mathbf{M}, \frac{\nabla \left(\langle \mathbf{M}, \exp(\mathbf{Y}) \rangle \right)}{\operatorname{Tr} \exp(\mathbf{Y})} \right\rangle - \left(\frac{\langle \mathbf{M}, \exp(\mathbf{Y}) \rangle}{\operatorname{Tr} \exp(\mathbf{Y})} \right)^2 \leq \left\langle \mathbf{M}, \frac{\nabla \left(\langle \mathbf{M}, \exp(\mathbf{Y}) \rangle \right)}{\operatorname{Tr} \exp(\mathbf{Y})} \right\rangle, \end{aligned} \tag{8}$$

where the second line used $\nabla \operatorname{Tr} \exp(\mathbf{Y}) = \exp(\mathbf{Y})$ *by Lewis's theorem (Theorem 8, Part V). Next, by using a Taylor expansion and the fact*[2] *that* $\nabla \left\langle \mathbf{M}, \mathbf{Y}^k \right\rangle = \sum_{i=0}^{k-1} \mathbf{Y}^i \mathbf{M} \mathbf{Y}^{k-1-i}$*, we have*

$$\nabla \left(\langle \mathbf{M}, \exp(\mathbf{Y}) \rangle \right) = \sum_{k \geq 0} \frac{1}{k!} \nabla \left(\left\langle \mathbf{M}, \mathbf{Y}^k \right\rangle \right) = \sum_{k \geq 1} \sum_{i=0}^{k-1} \frac{1}{k!} \mathbf{Y}^i \mathbf{M} \mathbf{Y}^{k-1-i},$$

so

$$\begin{aligned} \left\langle \mathbf{M}, \nabla \left(\langle \mathbf{M}, \exp(\mathbf{Y}) \rangle \right) \right\rangle &= \sum_{k=1}^{\infty} \sum_{i=0}^{k-1} \frac{1}{k!} \left\langle \mathbf{M}, \mathbf{Y}^i \mathbf{M} \mathbf{Y}^{k-1-i} \right\rangle \\ &\leq \sum_{k \geq 1} \frac{1}{(k-1)!} \left\langle \mathbf{M}^2, \mathbf{Y}^{k-1} \right\rangle = \left\langle \mathbf{M}^2, \exp(\mathbf{Y}) \right\rangle, \end{aligned} \tag{9}$$

where we used Lemma 7, Part V in the only inequality. Lemma 1 follows from (8) and (9).

As an immediate consequence of Lemma 1 and the smoothness-strong convexity duality characterization in Lemma 4, Part III, we therefore have also shown the following.

Corollary 1. *The quantum entropy* φ *in (2) is* 1*-strongly convex in* $\|\cdot\|_{\mathrm{tr}}$ *over* $\Delta^{d \times d}$*.*

2 Matrix multiplicative weights

We now present the MMW algorithm. Given a sequence of loss matrices $\{\mathbf{G}_t\}_{0 \leq t < T} \in \mathbb{S}^{d \times d}$, MMW minimizes regret over $\mathcal{X} := \Delta^{d \times d}$ via mirror descent (Theorem 2 and Remark 4, Part III) with the quantum entropy regularizer φ in (2). We prove some basic facts about an initialization strategy, following a suggestion in Remark 6, Part III to use the minimizer of φ.

Lemma 2. *Let* $\mathbf{X}_0 := \frac{1}{d} \mathbf{I}_d \in \mathcal{X}$.[3] *Then* $\mathbf{X}_0$ *minimizes* φ *over* $\mathcal{X}$*, and* $D_\varphi(\mathbf{X} \| \mathbf{X}_0) \leq \varphi(\mathbf{X}) - \varphi(\mathbf{X}_0) \leq \log d$ *for all* $\mathbf{X} \in \mathcal{X}$*.*

Proof. Because φ is a spectral function, the minimizing argument has eigenvalues agreeing with the minimizer of vector entropy over Δ^d, i.e. it has eigenvalues $\frac{1}{d} \mathbb{1}_d$. The only such matrix in $\mathcal{X}$ is $\frac{1}{d} \mathbf{I}_d$. The last claim follows from Remark 6, Part III, as $\varphi(\mathbf{X}) \leq 0$ and $\varphi(\mathbf{X}_0) = \log(d)$. □

By directly applying Lemma 2 and Corollary 1 in the context of mirror descent (i.e. Theorem 2, Part III), we have the following generic regret minimization guarantee over $\mathcal{X}$.

Theorem 1 (Matrix multiplicative weights)**.** *Let* $\{\mathbf{G}_t\}_{0 \leq t < T} \in \mathbb{S}^{d \times d}$ *satisfy* $\|\mathbf{G}_t\|_{\mathrm{op}} \leq L$ *for all* $0 \leq t < T$*. Let* $\mathbf{X}_0 \leftarrow \frac{1}{d} \mathbf{I}_d \in \mathcal{X}$*, and iteratively define*

$$\mathbf{X}_{t+1} \leftarrow \nabla \varphi^*(\nabla \varphi(\mathbf{X}_t) - \eta \mathbf{G}_t), \text{ for all } 0 \leq t < T, \tag{10}$$

where φ *is quantum entropy (2). We have*

$$\frac{1}{T} \sum_{0 \leq t < T} \langle \mathbf{G}_t, \mathbf{X}_t - \mathbf{X}^\star \rangle \leq \frac{\log d}{\eta T} + \frac{\eta L^2}{2}, \text{ for all } \mathbf{X}^\star \in \mathcal{X}.$$

[2] The easiest way to see this fact is to expand $(\mathbf{Y} + \mathrm{d}\mathbf{Y})^k$ to first order.

[3] This matrix is also sometimes called the *maximally mixed state* in quantum information theory, and this lemma is analogous to the fact that the uniform distribution minimizes (vector) entropy over the simplex.

In particular, choosing $\eta = \frac{1}{L}(\frac{2\log d}{T})^{1/2}$*, the right-hand side is* $L(\frac{2\log d}{T})^{1/2}$.

We pause to interpret the updates (10). Because φ is a Legendre function over $\mathcal{X}$, we may apply Lemma 3, Part III to conclude that $\nabla\varphi$ and $\nabla\varphi^*$ are inverse functions, sending $\mathcal{X}$ to $\mathbb{S}^{d\times d}$ and back respectively. By defining $\mathbf{S}_t := \nabla\varphi(\mathbf{X}_t)$ in each iteration, the recursion (10) is equivalent to

$$\mathbf{S}_0 \leftarrow \mathbf{0}_d,\ \mathbf{S}_{t+1} \leftarrow \mathbf{S}_t - \eta\mathbf{G}_t \text{ for all } 0 \le t < T. \tag{11}$$

In other words, a dual view of MMW is that it maintains a scaled running sum $\mathbf{S}_t := -\eta\sum_{0\le s<t}\mathbf{G}_s$, and updates its actions by using the Bregman projection induced by quantum entropy, $\mathbf{X}_t \leftarrow \nabla\varphi^*(\mathbf{S}_t)$. Recalling our earlier computation of $\nabla\varphi^*$ in (4), we have the more explicit description

$$\mathbf{X}_t \leftarrow \frac{\exp(\mathbf{S}_t)}{\operatorname{Tr}\exp(\mathbf{S}_t)} = \frac{\exp(-\eta\sum_{0\le s<t}\mathbf{G}_s)}{\operatorname{Tr}\exp(-\eta\sum_{0\le s<t}\mathbf{G}_s)}. \tag{12}$$

This dual view turns out to give an alternative proof of the MMW regret bound in Theorem 1, which is able to use the tighter characterization of $\nabla^2\varphi^*$ given in Lemma 1, as long as the matrices $\{\mathbf{G}_t\}_{0\le t<T}$ are all negative semidefinite.[4] We begin with a helper lemma.

Lemma 3. *For* $\mathbf{G} \in \mathbb{S}^{d\times d}_{\preceq\mathbf{0}}$ *satisfying* $\|\mathbf{G}\|_{\mathrm{op}} \le \frac{1}{2}$*, and all* $\mathbf{Y} \in \mathbb{S}^{d\times d}$*,*

$$D_{\varphi^*}(\mathbf{Y}+\mathbf{G}\|\mathbf{Y}) \le \|\mathbf{G}\|_{\mathrm{op}} \langle \mathbf{G}, \nabla\varphi^*(\mathbf{Y})\rangle .$$

Proof. Note that φ^* is twice-differentiable because its vector counterpart is, by Theorem 9, Part V. Therefore, by a Taylor expansion, we compute

$$\begin{aligned} D_{\varphi^*}(\mathbf{Y}+\mathbf{G}\|\mathbf{Y}) &= \int_0^1\int_0^t \left(\frac{\mathrm{d}^2}{\mathrm{d}s^2} D_{\varphi^*}(\mathbf{Y}+s\mathbf{G}\|\mathbf{Y})\right)\mathrm{d}s\mathrm{d}t \\ &= \int_0^1\int_0^t \nabla^2\varphi^*(\mathbf{Y}+s\mathbf{G})[\mathbf{G},\mathbf{G}]\mathrm{d}s\mathrm{d}t. \end{aligned}$$

By applying Lemma 1 and the fact that $\mathbf{G}^2 \preceq \|\mathbf{G}\|_{\mathrm{op}}\mathbf{G}$ for $\mathbf{G} \in \mathbb{S}^{d\times d}_{\preceq\mathbf{0}}$,

$$\begin{aligned} D_{\varphi^*}(\mathbf{Y}+\mathbf{G}\|\mathbf{Y}) &\le \|\mathbf{G}\|_{\mathrm{op}} \int_0^1\int_0^t \langle \nabla\varphi^*(\mathbf{Y}+s\mathbf{G}), \mathbf{G}\rangle \,\mathrm{d}s\mathrm{d}t \\ &= \|\mathbf{G}\|_{\mathrm{op}} \int_0^1 (\varphi^*(\mathbf{Y}+t\mathbf{G}) - \varphi^*(\mathbf{Y}))\,\mathrm{d}t \\ &= \|\mathbf{G}\|_{\mathrm{op}} \int_0^1 (D_{\varphi^*}(\mathbf{Y}+t\mathbf{G}\|\mathbf{Y}) + \langle t\mathbf{G}, \nabla\varphi^*(\mathbf{Y})\rangle)\,\mathrm{d}t. \end{aligned}$$

Next, observe that for $t \in [0,1]$,

$$\begin{aligned} t\cdot\frac{\mathrm{d}}{\mathrm{d}t} D_{\varphi^*}(\mathbf{Y}+t\mathbf{G}\|\mathbf{Y}) &= t\cdot\frac{\mathrm{d}}{\mathrm{d}t}\left(\varphi^*(\mathbf{Y}+t\mathbf{G}) - \langle t\mathbf{G}, \nabla\varphi^*(\mathbf{Y})\rangle\right) \\ &= \langle t\mathbf{G}, \nabla\varphi^*(\mathbf{Y}+t\mathbf{G}) - \nabla\varphi^*(\mathbf{Y})\rangle \ge 0, \end{aligned}$$

where we used that convexity of φ^* implies monotonicity of its gradient (Eq. (2), Part III). Therefore, $D_{\varphi^*}(\mathbf{Y}+t\mathbf{G}\|\mathbf{Y})$ is increasing in t, so we can bound the above display further:

$$\begin{aligned} D_{\varphi^*}(\mathbf{Y}+\mathbf{G}\|\mathbf{Y}) &\le \|\mathbf{G}\|_{\mathrm{op}} \int_0^1 (D_{\varphi^*}(\mathbf{Y}+\mathbf{G}\|\mathbf{Y}) + \langle t\mathbf{G}, \nabla\varphi^*(\mathbf{Y})\rangle)\,\mathrm{d}t \\ &\le \frac{1}{2} D_{\varphi^*}(\mathbf{Y}+\mathbf{G}\|\mathbf{Y}) + \frac{1}{2}\|\mathbf{G}\|_{\mathrm{op}} \langle \mathbf{G}, \nabla\varphi^*(\mathbf{Y})\rangle . \end{aligned}$$

Here, we used our assumption $\|\mathbf{G}\|_{\mathrm{op}} \le \frac{1}{2}$. The conclusion follows by rearranging. □

By using Lemma 3 in a dual variant of the mirror descent proof, we obtain the following.

[4] This restriction can be somewhat loosened; see Theorem 3.1 of [ZLO15] for a generalization of Theorem 2, which tolerates a larger set of matrices. In the applications we consider later in the course, Theorem 2 suffices.

Theorem 2 (Local norm MMW)**.** *In the setting of Theorem 2, suppose* $\{-\mathbf{G}_t\}_{0\le t<T} \in \mathbb{S}^{d\times d}_{\succeq \mathbf{0}}$ *and* $\eta L \le \frac{1}{2}$*. Then, following the notation in* (11)*,*

$$\frac{1}{T}\sum_{0\le t<T}\langle \mathbf{G}_t, \mathbf{X}_t - \mathbf{X}^\star\rangle \le \frac{\log d}{\eta T} - \frac{\eta}{T}\sum_{0\le t<T}\|\mathbf{G}_t\|_{\mathrm{op}}\langle \mathbf{G}_t, \mathbf{X}_t\rangle.$$

Proof. Let $\mathbf{S}^\star := \nabla\varphi(\mathbf{X}^\star)$ throughout. Following (10), (11), recall that we have $\mathbf{X}_t = \nabla\varphi^*(\mathbf{S}_t)$ in each iteration, so that the three-point equality of Bregman divergences (Eq. (7), Part III) shows

$$\begin{aligned}\eta\langle \mathbf{G}_t, \mathbf{X}_t - \mathbf{X}^\star\rangle &= \langle \mathbf{S}_t - \mathbf{S}_{t+1}, \nabla\varphi^*(\mathbf{S}_t) - \nabla\varphi^*(\mathbf{S}^\star)\rangle \\ &= D_{\varphi^*}(\mathbf{S}_t\|\mathbf{S}^\star) - D_{\varphi^*}(\mathbf{S}_{t+1}\|\mathbf{S}^\star) + D_{\varphi^*}(\mathbf{S}_{t+1}\|\mathbf{S}_t) \\ &\le D_{\varphi^*}(\mathbf{S}_t\|\mathbf{S}^\star) - D_{\varphi^*}(\mathbf{S}_{t+1}\|\mathbf{S}^\star) - \|\mathbf{G}\|_{\mathrm{op}}\langle \mathbf{G}_t, \mathbf{X}_t\rangle,\end{aligned}$$

where the last inequality applied Lemma 3. Telescoping as usual yields the claim, where we use that by combining Fact 4, Part III, and Lemma 2,

$$D_{\varphi^*}(\mathbf{S}_0\|\mathbf{S}^\star) = D_\varphi(\mathbf{X}^\star\|\mathbf{X}_0) \le \log d.$$

□

To understand the condition that $\{-\mathbf{G}_t\}_{0\le t<T} \in \mathbb{S}^{d\times d}_{\succeq \mathbf{0}}$, later in the course, we will encounter several applications where $\mathbf{M}_t := -\mathbf{G}_t$ is an empirical second moment matrix, i.e. for some vector $X_t \sim \mathcal{D}$ for a distribution $\mathcal{D}$ of interest, we let $\mathbf{M}_t = X_t X_t^\top$. In this case, our goal is to certify bounds on the empirical covariance $\frac{1}{T}\sum_{0\le t<T}\mathbf{M}_t$, whereby we have

$$\max_{\mathbf{X}^\star\in\mathcal{X}}\frac{1}{T}\sum_{0\le t<T}\langle -\mathbf{G}_t, \mathbf{X}^\star\rangle = \max_{\mathbf{X}^\star\in\mathcal{X}}\frac{1}{T}\sum_{0\le t<T}\langle \mathbf{M}_t, \mathbf{X}^\star\rangle = \left\|\frac{1}{T}\sum_{0\le t<T}\mathbf{M}_t\right\|_{\mathrm{op}}.$$

The left-hand side is exactly the type of quantity which Theorem 2 lets us control.

As we will see, the bound in Theorem 2 gives us significantly tighter bounds on such quantities than the looser Theorem 1. Bounds of the form in Theorem 2 are often called "local norms" bounds in the literature on multiplicative weights, because they are induced by a local reweighting by a fixed matrix $\mathbf{X}_t$, improving upon the "worst-case" bound $\|\mathbf{G}_t\|_{\mathrm{op}} \ge \langle \mathbf{G}_t, \mathbf{X}_t\rangle$. As a simple motivating example, if $\mathbf{G} = -XX^\top$ for $X \sim \mathcal{N}(\mathbb{0}_d, \mathbf{I}_d)$, then Gaussian moment bounds show

$$\mathbb{E}\|\mathbf{G}\|_{\mathrm{op}}^2 = \mathbb{E}\|X\|_2^4 = O(d^2), \text{ but } \mathbb{E}\left[\|\mathbf{G}\|_{\mathrm{op}}\langle -\mathbf{G}, \mathbf{X}\rangle\right] = O(d) \text{ for all } \mathbf{X}\in\mathcal{X},$$

since Gaussian samples have norm $\approx \sqrt{d}$ but their expected inner product with any fixed unit vector direction (or a convex combination thereof) scales as $O(1)$.[5] In particular, writing $\mathbf{X}\in\mathcal{X}$ as $\sum_{i\in[d]}\lambda_i u_i u_i^\top$ for unit vectors $\{u_i\}_{i\in[d]}$ and $\lambda\in\Delta^d$, we used

$$\mathbb{E}\left[\langle XX^\top, \mathbf{X}\rangle\right] = \sum_{i\in[d]}\lambda_i\mathbb{E}\langle X, u_i\rangle^2 = \sum_{i\in[d]}\lambda_i = 1.$$

3 Simplex-spectraplex games

In this section, we discuss computational aspects of the MMW algorithm. We are particularly motivated by an instantiation of MMW used to solve a minimax optimization problem between $\mathbf{X}\in\Delta^{d\times d}$ and $y\in\Delta^n$, i.e. a "simplex-spectraplex game," of the form

$$\min_{\mathbf{X}\in\Delta^{d\times d}}\max_{y\in\Delta^n}\sum_{i\in[n]}y_i\langle \mathbf{A}_i, \mathbf{X}\rangle. \tag{13}$$

[5] We will see a formal proof of the above claim, using that Gaussian random vectors have *hypercontractive moments*, i.e. $\mathbb{E}\langle X, v\rangle^4 = O(\mathbb{E}\langle X, v\rangle^2)^2$, later in the course.

Here, $\{\mathbf{A}_i\}_{i\in[n]} \in \mathbb{S}^{d\times d}$ are a set of specified matrices. This is not the only use case of MMW, but will be illustrative of relevant techniques, so we focus on it. We also introduce the notation

$$\mathcal{A}(y) := \sum_{i\in[n]} y_i \mathbf{A}_i \text{ for } y \in \mathbb{R}^n, \ \mathcal{A}^*(\mathbf{X}) := \{\langle \mathbf{A}_i, \mathbf{X}\rangle\}_{i\in[n]} \text{ for } \mathbf{X} \in \mathbb{S}^{d\times d}.$$

Under this notation, (13) is simply $\min_{\mathbf{X}\in\Delta^{d\times d}} \max_{y\in\Delta^n} \langle \mathcal{A}(y), \mathbf{X}\rangle$. Moreover,

$$g(\mathbf{X}, y) = (\mathcal{A}(y), -\mathcal{A}^*(\mathbf{X})) \tag{14}$$

is the monotone operator associated with (13), in the sense of Eq. (5), Part IV. Finally, in analogy to the simplex-simplex game $\min_{x\in\Delta^d} \max_{y\in\Delta^n} y^\top \mathbf{A} x$ explored in Section 2, Part IV, where the natural Lipschitz constant was

$$\|\mathbf{A}\|_{\max} = \max_{i\in[d]} \max_{j\in[n]} |\mathbf{A}_{ij}| = \max_{x\in\Delta^d} \max_{y\in\Delta^n} y^\top \mathbf{A} x,$$

and the norm in question was a joint ℓ_1-ℓ_1 norm, in this section we denote

$$\mathcal{Z} := \underbrace{\Delta^{d\times d}}_{:=\mathcal{X}} \times \underbrace{\Delta^n}_{:=\mathcal{Y}}, \ \|(\mathbf{X}, y)\| := \sqrt{\|\mathbf{X}\|_{\mathrm{tr}}^2 + \|y\|_1^2}, \ \varphi(\mathbf{X}, y) := \langle \mathbf{X}, \log(\mathbf{X})\rangle + \langle y, \log y\rangle, \tag{15}$$

where $\log y$ is entrywise in the last definition. Finally, we define the Lipschitz constant

$$L := \max_{i\in[n]} \|\mathbf{A}_i\|_{\mathrm{op}} = \max_{\mathbf{X}\in\mathcal{X}} \max_{y\in\mathcal{Y}} \langle \mathcal{A}(y), \mathbf{X}\rangle. \tag{16}$$

The following claim on the regularity of (13) is a straightforward extension of Lemma 5, Part IV.

Fact 1. *In the context of* (13)*, and following the notation* (14)*,* (15)*,* (16)*,*

$$\sup_{z\in\mathcal{Z}} \|g(z)\|_* \le \sqrt{2}L.$$

By applying Proposition 1, Part IV, to the game in (13) using the regularity bound in Fact 1, it is straightforward to show that we can obtain an ϵ-approximate saddle point to (13) in

$$T = O\left(\frac{L^2 \log(nd)}{\epsilon^2}\right)$$

iterations, in exactly the same way as was done in Corollary 1, Part IV. However, a naïve implementation of this strategy requires computing the updates $\mathbf{X}_t$ given by (12) exactly, in order to compute the operator (14). Because exponentiating a matrix is highly-expensive in general (requiring a full eigendecomposition), and runs into numerical stability issues in practice (see the famous documentation of these problems in [ML78]), we would like to avoid an exact implementation. After a brief motivation of the problem (13) in Section 3.1, we show how to run the algorithm in Proposition 1, Part IV in nearly-linear time in Section 3.2, via techniques we have developed.

3.1 Semidefinite programs

One application of (13) is solving *semidefinite programs* (SDPs), a matrix generalization of linear programs (LPs). In the standard form of SDP, we are given n matrices $\{\mathbf{A}_i\}_{i\in[n]} \in \mathbb{S}^{d\times d}$, and an additional vector and matrix $b \in \mathbb{R}^n_{\ge 0}$, $\mathbf{C} \in \mathbb{S}^{d\times d}$, and we wish to determine the value of

$$\max_{\mathbf{X}\in\mathbb{S}^{d\times d}_{\succeq \mathbf{0}}} \langle \mathbf{C}, \mathbf{X}\rangle, \text{ subject to } \langle \mathbf{A}_i, \mathbf{X}\rangle \le b_i \text{ for all } i \in [n]. \tag{17}$$

Note that (17) exactly recovers the standard form of an LP when all of $\{\mathbf{A}_i\}_{i\in[n]}, \mathbf{C}, \mathbf{X}$ are restricted to be diagonal matrices. We now sketch how (17) is reducible to (13).

Typically, we can find a value R such that the constraints in (17) imply that any feasible solution $\mathbf{X}$ must satisfy $\mathrm{Tr}(\mathbf{X}) \le R$, and we can introduce a dummy $(d+1)^{\text{th}}$ dimension to make the

trace inequality an equality, i.e. $\mathrm{Tr}(\mathbf{X}) = R$ exactly. Therefore, scaling the whole problem by $\frac{1}{R}$, overloading $d \leftarrow d+1$, and negating $\mathbf{C}$, (17) becomes

$$\min_{\mathbf{X} \in \Delta^{d \times d}} \langle \mathbf{C}', \mathbf{X} \rangle, \text{ subject to } \langle \mathbf{A}'_i, \mathbf{X} \rangle \leq b'_i \text{ for all } i \in [n], \tag{18}$$

for appropriate $\{\mathbf{A}'_i, b'_i\}_{i \in [n]}$, $\mathbf{C}'$ constructed from the original problem instance. Moreover, by binary searching on the optimal value $t = \langle \mathbf{C}', \mathbf{X} \rangle$, we can reduce (18) from an optimization problem into a feasibility problem, where we want to check if there is $\mathbf{X} \in \mathbb{S}^{d \times d}_{\succeq \mathbf{0}}$ satisfying the constraints $\langle \mathbf{A}'_i, \mathbf{X} \rangle \leq b'_i$ for all $i \in [n]$, as well as $\langle \mathbf{C}', \mathbf{X} \rangle \leq t$, i.e. we treat $\mathbf{C}'$ as another constraint matrix. Finally, subtracting $b'_i \mathbf{I}_d$ from each $\mathbf{A}'_i$, we have reduced (17) to determining the value of

$$\min_{\mathbf{X} \in \Delta^{d \times d}} \max_{i \in [n]} \left\langle \widetilde{\mathbf{A}}_i, \mathbf{X} \right\rangle,$$

in particular its sign, for some $\{\widetilde{\mathbf{A}}_i\}_{i \in [n]}$ constructed from the original problem instance. This new problem is precisely an instance of (13), up to reparameterizing $\mathbf{A}_i \leftarrow \widetilde{\mathbf{A}}_i$.

The upshot is that the MMW algorithm can be implemented to run highly-efficiently, albeit only to low accuracy, as discussed in the following Section 3.2 (see also Theorem 3). This is in contrast to standard high-accuracy solvers for semidefinite programming, where the state-of-the-art [JKL+20, HJS+22] runs in time $\approx \min(\sqrt{d}(nd^2 + n^\omega + d^\omega), \sqrt{d}(n^2 + d^4) + n^\omega + d^{2\omega})$, which can be significantly superlinear compared to the problem description size, bounded by $O(nd^2)$. What is somehow lost in this reduction is any sense of relative scale of the problem, since the worst-case normalization R can be very large. Nonetheless, in many interesting applications, we can use additional structure of the SDP problem to argue this overhead is small enough, so that we can still say interesting things about semidefinite program solutions using a nearly-linear time algorithm.

We take the liberty of listing a few examples here, for the reader's interest. First, MMW was used in [AK07, OSVV08] to solve combinatorial semidefinite programs which approximate graph partitioning problems such as sparsest cut and balanced separator. Moreover, it has been used for faster numerical linear algebra primitives such as spectral sparsification and preconditioning [ZLO15, LS17, JLM+23], which we discuss in more detail next lecture. It has also resulted in significantly faster algorithms for robust statistics in various settings [CDG19, DHL19, CMY20, DKK+21, DL22], discussed later in the course. Finally, MMW was the workhorse algorithm in the proof that QIP = PSPACE [JJUW11], a landmark result in quantum complexity theory.

3.2 Nearly-linear time implementation

Consider a run of the minimax variant of mirror descent (Proposition 1, Part IV) on the problem (13), under the setup described in (14), (15), (16). In each iteration t, parameterized by iterate blocks $z_t := (\mathbf{X}_t, y_t)$, the key computational problem is to compute the vector block of the minimax operator (14), i.e. $\mathcal{A}^*(\mathbf{X}_t)$. To see why, first note that on the vector side, given access to the vector block of $g(z_t)$, we can update the vector iterate y_t to y_{t+1} in linear time.

Moreover, we can implicitly maintain the matrix $\mathbf{X}_t$ as $\nabla\varphi^*(\mathbf{S}_t)$, as described by the updates (11), (12). In our particular case, where $\mathbf{G}_t = \mathcal{A}(y_t)$, is the matrix block of $g(z_t)$ in (14), we can instead maintain the vector $v_t := -\eta \sum_{0 \leq s < t} y_t$ in linear time, so that (11), (12) imply

$$\mathbf{X}_t = \nabla\varphi^*(\mathcal{A}(v_t)) = \frac{\exp(\mathcal{A}(v_t))}{\mathrm{Tr}\exp(\mathcal{A}(v_t))}. \tag{19}$$

Therefore, as long as this implicit representation of $\mathbf{X}_t$ is acceptable, the computational bottleneck in each iteration is to compute, for some explicit vector v,

$$\langle \nabla\varphi^*(\mathcal{A}(v)), \mathbf{A}_i \rangle = \frac{\langle \exp(\mathcal{A}(v)), \mathbf{A}_i \rangle}{\mathrm{Tr}\exp(\mathcal{A}(v))} \text{ for all } i \in [n]. \tag{20}$$

Indeed, this expression is precisely the vector block of (14), when $\mathbf{X} = \nabla\varphi^*(\mathcal{A}(v))$. We split our discussion by numerator and denominator, i.e. we will separately discuss computation of

$$\mathrm{Tr}\exp(\mathcal{A}(v)), \tag{21}$$

and

$$\langle \exp(\mathcal{A}(v)), \mathbf{A}_i \rangle \text{ for all } i \in [n]. \tag{22}$$

Finally, to simplify our runtime claims, we will only consider the setting where $\mathcal{T}_{\text{mv}}(\mathbf{A}_i) = \Omega(d)$ for all $i \in [n]$, which is typically true unless $\mathbf{A}_i$ is unusually sparse.

We begin by showing how to approximate the value of (21) to multiplicative accuracy.

Proposition 1. *Let $v \in \mathbb{R}^n$ satisfy $\|v\|_1 \le \rho$, and let $\epsilon, \delta \in (0,1)$. We can compute a value V such that with probability $\ge 1-\delta$, $V \in [(1-\epsilon)\operatorname{Tr}\exp(\mathcal{A}(v)), (1+\epsilon)\operatorname{Tr}\exp(\mathcal{A}(v))]$, in time*

$$O\left(\left(\sum_{i\in[n]} \mathcal{T}_{\text{mv}}(\mathbf{A}_i)\right) \cdot \frac{L\rho \log\left(\frac{1}{\epsilon}\right) \log\left(\frac{d}{\delta}\right)}{\epsilon^2}\right).$$

Proof. Throughout the proof, let $\mathbf{M} := \mathcal{A}(v)$, and note that following the notation (16),

$$\|\mathbf{M}\|_{\text{op}} \le \sum_{i\in[n]} |v_i| \, \|\mathbf{A}_i\|_{\text{op}} \le L \sum_{i\in[n]} |v_i| \le \underbrace{L\rho}_{:=R}.$$

Additionally, it is clear that $\mathcal{T}_{\text{mv}}(\mathbf{M}) = O(\sum_{i\in[n]} \mathcal{T}_{\text{mv}}(\mathbf{A}_i))$. We now describe our strategy.

First, in place of computing $\operatorname{Tr}\exp(\mathbf{M})$, we will compute $\operatorname{Tr}(p(\mathbf{M})^2)$, for a polynomial p such that p^2 is a multiplicative approximation to exp on the range $[-R, R]$. Indeed, letting q be the degree-$\Delta := O(R \log \frac{1}{\epsilon})$ polynomial on $[0,1]$ given by Lemma 2, Part VI, such that

$$\sup_{x\in[0,1]} |\exp(-Rx) - q(x)| \le \frac{\epsilon}{9}\exp(-R),$$

we can let $p(t) = \exp(\frac{R}{2}) \cdot q(\frac{1}{2R}(R-t))$, so that changing variables appropriately,

$$\begin{aligned}
\sup_{t\in[-R,R]} \left|\exp\left(\frac{t}{2}\right) - p(t)\right| &= \exp\left(\frac{R}{2}\right) \cdot \sup_{t\in[-R,R]} \left|\exp\left(-R\cdot\left(\frac{R-t}{2R}\right)\right) - q\left(\frac{R-t}{2R}\right)\right| \\
&\le \exp\left(\frac{R}{2}\right) \cdot \sup_{x\in[0,1]} |\exp(-Rx) - q(x)| \\
&\le \frac{\epsilon}{9}\exp\left(-\frac{R}{2}\right) \le \frac{\epsilon}{9}\exp\left(\frac{t}{2}\right), \text{ for all } t \in [-R, R].
\end{aligned}$$

Also, p is degree-Δ. Finally, since $p(t) \in [(1-\frac{\epsilon}{9})\exp(\frac{t}{2}), (1+\frac{\epsilon}{9})\exp(\frac{t}{2})]$ for all $t \in [-R, R]$,

$$p(t)^2 \in \left[\left(1-\frac{\epsilon}{3}\right)\exp(t), \left(1+\frac{\epsilon}{3}\right)\exp(t)\right], \text{ for all } t \in [-R, R]. \tag{23}$$

In other words, every eigenvalue of $p(\mathbf{M})^2$ is within a $\frac{\epsilon}{3}$ multiplicative factor of the corresponding eigenvalue of $\exp(\mathbf{M})$, so we lose only this factor when using $\operatorname{Tr}(p(\mathbf{M})^2)$ instead.

Second, let $\{r_i\}_{i\in[d]}$ be the rows of $p(\mathbf{M})$, so that

$$\operatorname{Tr}(p(\mathbf{M})^2) = \|p(\mathbf{M})\|_{\text{F}}^2 = \sum_{i\in[d]} \|r_i\|_2^2.$$

The next idea is to apply a random sketching matrix $\mathbf{G} \in \mathbb{R}^{k\times d}$, such that $\|\mathbf{G}r_i\|_2 \approx \|r_i\|_2$ for all $i \in [d]$. Fortunately, the Johnson-Lindenstrauss lemma (Corollary 1, Part V) shows that if we choose $k = O(\frac{1}{\epsilon^2}\log\frac{d}{\delta})$ for an appropriate constant, we have

$$\|\mathbf{G}r_i\|_2^2 \in \left[\left(1-\frac{\epsilon}{3}\right)\|r_i\|_2^2, \left(1+\frac{\epsilon}{3}\right)\|r_i\|_2^2\right] \text{ for all } i \in [d],$$

with probability $1-\delta$ over a random Gaussian matrix $\mathbf{G} \in \mathbb{R}^{k\times d}$ with each entry i.i.d. $\sim \mathcal{N}(0, \frac{1}{k})$, by a union bound over the rows. So, in other words we have shown that it suffices to compute

$$V := \sum_{i\in[d]} \|\mathbf{G}r_i\|_2^2 \in [(1-\epsilon)\operatorname{Tr}\exp(\mathbf{M}), (1+\epsilon)\operatorname{Tr}\exp(\mathbf{M})].$$

Finally, note that if we let the rows of $\mathbf{G}$ be denoted $\{g_j\}_{j\in[k]}$ for convenience,

$$\begin{aligned}\sum_{i\in[d]}\|\mathbf{G}r_i\|_2^2 = \|\mathbf{G}p(\mathbf{M})\|_{\mathrm{F}}^2 &= \mathrm{Tr}\left(p(\mathbf{M})\mathbf{G}^\top\mathbf{G}p(\mathbf{M})\right)\\ &= \mathrm{Tr}\left(\mathbf{G}p(\mathbf{M})^2\mathbf{G}^\top\right) = \left\|p(\mathbf{M})\mathbf{G}^\top\right\|_{\mathrm{F}}^2 = \sum_{j\in[k]}\|p(\mathbf{M})g_j\|_2^2.\end{aligned} \tag{24}$$

Now, given a row g_j, we can compute $p(\mathbf{M})g_j$ explicitly in Δ matrix-vector multiplications to $\mathbf{M}$. We need to do this k times (once per row), yielding an overall runtime as claimed:

$$O\left(\mathcal{T}_{\mathrm{mv}}(\mathbf{M})\cdot\Delta\cdot k\right) = O\left(\left(\sum_{i\in[n]}\mathcal{T}_{\mathrm{mv}}(\mathbf{A}_i)\right)\cdot\frac{L_{\mathcal{A}}\cdot\rho\log\left(\frac{1}{\epsilon}\right)\log\left(\frac{d}{\delta}\right)}{\epsilon^2}\right).$$

□

Similarly, we show how to approximate the value of (22) to additive accuracy. We begin with a simpler variant of the computation required for a single value in (22).

Proposition 2. *Let $\mathbf{N}\in\mathbb{S}^{d\times d}_{\succeq\mathbf{0}}$, $v\in\mathbb{R}^n$ have $\|\mathbf{N}\|_{\mathrm{op}}\le\nu$, $\|v\|_1\le\rho$ respectively, and let $\epsilon,\delta\in(0,1)$. We can compute a value V such that with probability $\ge 1-\delta$, $V\in[(1-\epsilon)\left\langle\exp(\mathcal{A}(v)),\mathbf{N}\right\rangle,(1+\epsilon)\left\langle\exp(\mathcal{A}(v)),\mathbf{N}\right\rangle]$, in time*

$$O\left(\left(\sum_{i\in[n]}\mathcal{T}_{\mathrm{mv}}(\mathbf{A}_i)\right)\cdot\frac{L\rho\log(\frac{1}{\epsilon})\log(\frac{d}{\delta})}{\epsilon^2}+\mathcal{T}_{\mathrm{mv}}(\mathbf{N})\cdot\frac{\log(\frac{d}{\delta})}{\epsilon^2}\right).$$

Proof. We follow the notation of Proposition 1, letting p be the degree-$\Delta = O(R\log\frac{1}{\epsilon})$ polynomial satisfying (23) for $R := L\rho$. This means that for $\mathbf{M} := \mathcal{A}(v)$,

$$\left\langle p(\mathbf{M})^2,\mathbf{N}\right\rangle\in\left[\left(1-\frac{\epsilon}{3}\right)\left\langle\exp(\mathbf{M}),\mathbf{N}\right\rangle,\left(1+\frac{\epsilon}{3}\right)\left\langle\exp(\mathbf{M}),\mathbf{N}\right\rangle\right].$$

Now, letting $\{r_i\}_{i\in[d]}$ be the rows of $\mathbf{N}^{\frac{1}{2}}p(\mathbf{M})$, we have following the derivation in (24) that for some $k = O(\frac{1}{\epsilon^2}\log\frac{d}{\delta})$, and $\mathbf{G}$ a random $k\times d$ matrix with i.i.d. entries $\sim\mathcal{N}(0,\frac{1}{k})$,

$$\mathrm{Tr}\left(\mathbf{G}p(\mathbf{M})\mathbf{N}p(\mathbf{M})\mathbf{G}^\top\right)\in\left[\left(1-\frac{\epsilon}{3}\right)\left\langle p(\mathbf{M})^2,\mathbf{N}\right\rangle,\left(1+\frac{\epsilon}{3}\right)\left\langle p(\mathbf{M})^2,\mathbf{N}\right\rangle\right].$$

Finally, observe that, letting $\{g_i\}_{i\in[k]}$ be the rows of $\mathbf{G}$ for readability,

$$\mathrm{Tr}\left(\mathbf{G}p(\mathbf{M})\mathbf{N}p(\mathbf{M})\mathbf{G}^\top\right) = \sum_{i\in[k]}g_i^\top p(\mathbf{M})\mathbf{N}p(\mathbf{M})g_i. \tag{25}$$

Each summand requires Δ matrix-vector multiplications through $\mathbf{N}$, and one matrix-vector multiplication through $\mathbf{N}$. The conclusion follows by doing these multiplications k times. □

Finally, we give the following summary of the results in this section.

Corollary 2. *Let $v\in\mathbb{R}^n$ have $\|v\|_1\le\rho$, and let $\epsilon,\delta\in(0,1)$. We can compute values $\{u_i\}_{i\in[n]}$ such that with probability $\ge 1-\delta$,*

$$\left|u_i-\frac{\left\langle\exp(\mathcal{A}(v)),\mathbf{A}_i\right\rangle}{\mathrm{Tr}\exp(\mathcal{A}(v))}\right|\le\epsilon\text{ for all }i\in[n],$$

in time

$$O\left(\left(\sum_{i\in[n]}\mathcal{T}_{\mathrm{mv}}(\mathbf{A}_i)\right)\cdot\frac{L^3\rho\log(\frac{L}{\epsilon})\log(\frac{nd}{\delta})}{\epsilon^2}\right).$$

Proof. First, let $\mathbf{N}_i \leftarrow \mathbf{A}_i + L\mathbf{I}_d$ for all $i \in [n]$, so that $\mathbf{0}_d \preceq \mathbf{N}_i \preceq 2L\mathbf{I}_d$ for all $i \in [n]$. Next, we use Proposition 1 to compute a value V such that

$$V \in \left[\left(1 - \frac{\epsilon}{6L}\right) \operatorname{Tr}\exp(\mathcal{A}(v)), \left(1 + \frac{\epsilon}{6L}\right) \operatorname{Tr}\exp(\mathcal{A}(v))\right],$$

within the allotted time, with failure probability $\frac{\delta}{2}$. We also use Proposition 2 n times, each with failure probability $\frac{\delta}{2n}$, to compute values $\{N_i\}_{i\in[n]}$ such that

$$N_i \in \left[\left(1 - \frac{\epsilon}{6L}\right) \langle \exp(\mathcal{A}(v)), \mathbf{N}_i\rangle, \left(1 + \frac{\epsilon}{6L}\right) \langle \exp(\mathcal{A}(v)), \mathbf{N}_i\rangle\right].$$

Note that all n calls fit within the allotted runtime. To see this, we can simply reuse the sketching matrix $\mathbf{G}$ in all n calls, since the matrix $p(\mathbf{M})$ is the same. Therefore, when computing the values (25) required, we can first precompute $v_j = p(\mathbf{M})g_j$ for all $j \in [k]$ a single time, and then compute all $\sum_{j\in[k]} v_j^\top \mathbf{N}_i v_j$, which only requires k matrix-vector multiplications. Hence, we only pay the first term in the runtime in Proposition 2 a single time, to perform the precomputation.

Finally, consider the use of the approximations $\{u_i := \frac{N_i}{V} - L\}_{i\in[n]}$. We have for all $i \in [n]$,

$$\begin{aligned}
\frac{N_i}{V} - L &\le \left(1 + \frac{\epsilon}{6L}\right)^2 \frac{\langle \exp(\mathcal{A}(v)), \mathbf{N}_i\rangle}{\operatorname{Tr}\exp(\mathcal{A}(v))} - L \\
&\le \left(1 + \frac{\epsilon}{2L}\right) \left(\frac{\langle \exp(\mathcal{A}(v)), \mathbf{A}_i\rangle}{\operatorname{Tr}\exp(\mathcal{A}(v))} + L\right) - L \\
&\le \left(1 + \frac{\epsilon}{2L}\right) \frac{\langle \exp(\mathcal{A}(v)), \mathbf{A}_i\rangle}{\operatorname{Tr}\exp(\mathcal{A}(v))} + \frac{\epsilon}{2} \le \frac{\langle \exp(\mathcal{A}(v)), \mathbf{A}_i\rangle}{\operatorname{Tr}\exp(\mathcal{A}(v))} + \epsilon.
\end{aligned}$$

In the last inequality, we used

$$\frac{\langle \exp(\mathcal{A}(v)), \mathbf{A}_i\rangle}{\operatorname{Tr}\exp(\mathcal{A}(v))} \le \left\|\frac{\exp(\mathcal{A}(v))}{\operatorname{Tr}\exp(\mathcal{A}(v))}\right\|_{\mathrm{tr}} \|\mathbf{A}_i\|_{\mathrm{op}} \le L.$$

We can establish analogous lower bounds on $\frac{N_i}{V} - L$, yielding the claim. □

In a culmination of our efforts, Corollary 2 indeed solves the computational problem of approximating (20) entrywise, as described in the beginning of the section.

3.3 Putting it all together

We are now ready to state a complete runtime bound on solving the problem (13).

Theorem 3 (Simplex-spectraplex games)**.** *Consider an instance of (13), and define L as in (16). For $\epsilon \in (0, L)$ and $\delta \in (0, 1)$, there is an algorithm which computes an ϵ-approximate saddle point to (13) with probability $\ge 1 - \delta$, in time*

$$O\left(\left(\sum_{i\in[n]} \mathcal{T}_{\mathrm{mv}}(\mathbf{A}_i)\right) \cdot \frac{L^5 \log^2(nd) \log(\frac{L}{\epsilon}) \log(\frac{ndL}{\delta\epsilon})}{\epsilon^5}\right).$$

Proof. We first describe the algorithm. We run an approximate variant of the minimax variant of mirror descent described in Proposition 1, Part IV, using the operator g in (14), and the regularizer φ in (15). We also initialize the algorithm from $z_0 = (\mathbf{X}_0, y_0) := (\frac{1}{d}\mathbf{I}_d, \frac{1}{n}\mathbb{1}_n) \in \mathcal{Z}$, which satisfies the bound $D_\varphi(z\|z_0) \le \log(nd)$ for all $z \in \mathcal{Z}$, by Lemma 2 and Lemma 7, Part III. In each step $0 \le t < T$, the $\mathcal{X}$ iterate $\mathbf{X}_t$ is updated to $\mathbf{X}_{t+1}$ using the exact matrix component of (14), which can be implicitly maintained via the representation (19), i.e. we let

$$\mathbf{X}_t := \frac{\exp(\mathcal{A}(v_t))}{\operatorname{Tr}\exp(\mathcal{A}(v_t))},$$

only maintaining the vector $v_t := -\eta \sum_{0\le s<t} y_s$, which takes $O(d)$ time to update in each iteration. Next, instead of updating the $\mathcal{Y}$ iterate y_t to y_{t+1} via exactly computing the vector component

of (14), i.e. $\mathcal{A}^*(\mathbf{X}_t)$, we instead use Corollary 2 to obtain a vector $u_t \in \mathbb{R}^n$ which entrywise approximates the vector component up to error $\frac{\epsilon}{4}$. It is straightforward to check that

$$\langle \eta g(z_t), z_t - z^\star \rangle \leq D_\varphi(z^\star \| z_t) - D_\varphi(z^\star \| z_{t+1}) + \frac{\eta^2 L^2}{2} + \frac{\eta\epsilon}{2} \text{ for all } z^\star = (\mathbf{X}^\star, y^\star) \in \mathcal{Z}$$

in each iteration instead, following the standard proof of mirror descent (Theorem 2, Part III), where we use the ℓ_1-ℓ_∞ Hölder's inequality to bound the correction term $\langle \mathcal{A}^*(\mathbf{X}_t) - u_t, y_t - y^\star \rangle \leq \frac{\epsilon}{2}$ due to our approximation error, since $\|y_t - y^\star\|_1 \leq 2$. Therefore, telescoping this guarantee shows the duality gap after T iterations using our approximate computations is bounded by

$$\frac{\log(nd)}{\eta T} + \frac{\eta L^2}{2} + \frac{\epsilon}{2}.$$

It hence suffices to take $\eta = \frac{\epsilon}{2L^2}$, $T = \frac{8L^2 \log(nd)}{\epsilon^2}$ for this quantity to be bounded by ϵ as desired. Finally, note that for all $0 \leq t < T$, we have

$$\|v_t\|_1 \leq \eta \sum_{0 \leq s < t} \|y_s\|_1 = \eta t \leq \eta T \leq \frac{4\log(nd)}{\epsilon}.$$

Hence, it suffices to set $\rho = \frac{4\log(nd)}{\epsilon}$ in Corollary 2, which we call T times to give the runtime. □

Remark 2. *The runtime in Theorem 3 is appealing because for small values of $\frac{L}{\epsilon}$, the runtime scales linearly in the total input size, represented by $\sum_{i \in [n]} \mathcal{T}_{\text{mv}}(\mathbf{A}_i)$, as opposed to exponentiating a matrix, which would take time at least d^ω and may run into additional computational issues [ML78]. However, the overhead of $\approx (\frac{L}{\epsilon})^5$ is a far cry from the iteration complexity of $\approx (\frac{L}{\epsilon})^2$ achieved by mirror descent, or the $\approx \frac{L}{\epsilon}$ iterations required by mirror prox (Theorem 1, Part IV). Note that this overhead comes from $\approx (\frac{L}{\epsilon})^2$ iterations, where each iteration applies a degree $\approx \frac{L}{\epsilon}$ polynomial to a rank-$\approx (\frac{L}{\epsilon})^2$ sketch via the Johnson-Lindenstrauss lemma.*

There have been significant improvements to this runtime for MMW, using either faster frameworks (e.g. mirror prox), lower-rank sketches than Johnson-Lindenstrauss, or using lower-degree polynomials [BBN13, GHM15, AL17, CDST19]. First, the degree $\approx \frac{L}{\epsilon}$ of the polynomial we used in Propositions 1 and 2 can be sharpened to $\approx (\frac{L}{\epsilon})^{1/2}$, by doing an initial preprocessing to estimate the largest eigenvalue of $\mathbf{M}$. This lets us avoid requiring accuracy $\approx \exp(-R)$ from the approximating polynomial, so we can obtain the square root savings from Lemma 2, Part VI.

The state-of-the-art runtime for obtaining ϵ duality gap in (13) has an overhead which scales as $\approx (\frac{L}{\epsilon})^{2.5}$. This was achieved in two different ways by [BBN13, CDST19]. The strategy of [BBN13] was to use mirror prox, and to show that substituting a rank-$\approx \frac{L}{\epsilon}$ sketch in place of Johnson-Lindenstrauss enjoys a small-enough variance such that the error of the resulting method does not accumulate poorly. In [CDST19], it was shown that the standard MMW implementation via mirror descent actually gives the same regret guarantees up to constant factors, even if a rank-1 sketch is used. However, the analysis of [CDST19] does not compose well with mirror prox; it is an interesting open direction to see if there is a method which benefits from both approaches.

Source material

This lecture is based on the author's own experience working in the field.

References

[AK07] Sanjeev Arora and Satyen Kale. A combinatorial, primal-dual approach to semidefinite programs. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing, 2007*, pages 227–236. ACM, 2007.

[AL17] Zeyuan Allen-Zhu and Yuanzhi Li. Follow the compressed leader: Faster online learning of eigenvectors and faster MMWU. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017*, pages 116–125, 2017.

[BBN13] Michel Baes, Michael Bürgisser, and Arkadi Nemirovski. A randomized mirror-prox method for solving structured large-scale matrix saddle-point problems. *SIAM Journal on Optimization*, 23(2):934–962, 2013.

[CDG19] Yu Cheng, Ilias Diakonikolas, and Rong Ge. High-dimensional robust mean estimation in nearly-linear time. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019*, pages 2755–2771. SIAM, 2019.

[CDST19] Yair Carmon, John C. Duchi, Aaron Sidford, and Kevin Tian. A rank-1 sketch for matrix multiplicative weights. In *Conference on Learning Theory, COLT 2019*, pages 589–623, 2019.

[CMY20] Yeshwanth Cherapanamjeri, Sidhanth Mohanty, and Morris Yau. List decodable mean estimation in nearly linear time. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020*, pages 141–148. IEEE, 2020.

[DHL19] Yihe Dong, Samuel B. Hopkins, and Jerry Li. Quantum entropy scoring for fast robust mean estimation and improved outlier detection. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019*, pages 6065–6075, 2019.

[DKK+21] Ilias Diakonikolas, Daniel Kane, Daniel Kongsgaard, Jerry Li, and Kevin Tian. List-decodable mean estimation in nearly-pca time. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021*, pages 10195–10208, 2021.

[DL22] Jules Depersin and Guillaume Lecué. Robust sub-gaussian estimation of a mean vector in nearly linear time. *Annals of Statistics*, 50(1):511–536, 2022.

[GHM15] Dan Garber, Elad Hazan, and Tengyu Ma. Online learning of eigenvectors. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015*, pages 560–568, 2015.

[HJS+22] Baihe Huang, Shunhua Jiang, Zhao Song, Runzhou Tao, and Ruizhe Zhang. Solving SDP faster: A robust IPM framework and efficient implementation. In *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022*, pages 233–244. IEEE, 2022.

[JJUW11] Rahul Jain, Zhengfeng Ji, Sarvagya Upadhyay, and John Watrous. QIP = PSPACE. *J. ACM*, 58(6):30:1–30:27, 2011.

[JKL+20] Haotian Jiang, Tarun Kathuria, Yin Tat Lee, Swati Padmanabhan, and Zhao Song. A faster interior point method for semidefinite programming. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020*, pages 910–918. IEEE, 2020.

[JLM+23] Arun Jambulapati, Jerry Li, Christopher Musco, Kirankumar Shiragur, Aaron Sidford, and Kevin Tian. Structured semidefinite programming for recovering structured preconditioners. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023*, 2023.

[KW07] Dima Kuzmin and Manfred K. Warmuth. Online kernel PCA with entropic matrix updates. In *Machine Learning, Proceedings of the Twenty-Fourth International Conference (ICML 2007*, volume 227 of *ACM International Conference Proceeding Series*, pages 465–472. ACM, 2007.

[LS17] Yin Tat Lee and He Sun. An sdp-based algorithm for linear-sized spectral sparsification. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017*, pages 678–687. ACM, 2017.

[ML78] Cleve Moler and Charles Van Loan. Nineteen dubious ways to compute the exponential of a matrix. *SIAM Review*, 20(4):831–836, 1978.

[NC10] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2010.

[Nes07] Y. Nesterov. Smoothing technique and its applications in semidefinite optimization. *Math. Program.*, 110:245–259, 2007.

[OSVV08] Lorenzo Orecchia, Leonard J. Schulman, Umesh V. Vazirani, and Nisheeth K. Vishnoi. On partitioning graphs via single commodity flows. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, 2008*, pages 461–470. ACM, 2008.

[TRW05] Koji Tsuda, Gunnar Rätsch, and Manfred K. Warmuth. Matrix exponentiated gradient updates for on-line learning and bregman projection. *J. Mach. Learn. Res.*, 6:995–1018, 2005.

[ZLO15] Zeyuan Allen Zhu, Zhenyu Liao, and Lorenzo Orecchia. Spectral sparsification and regret minimization beyond matrix multiplicative updates. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015*, pages 237–245. ACM, 2015.