Sublinear Classical and Quantum Algorithms for General Matrix Games

Tongyang Li,[*1,2] **Chunhao Wang,**[*3,4] **Shouvanik Chakrabarti,**[1] **Xiaodi Wu**[1]

[1]Joint Center for Quantum Information and Computer Science, Department of Computer Science, and Institute for Advanced Computer Studies, University of Maryland
[2]Center for Theoretical Physics, Massachusetts Institute of Technology
[3]Department of Computer Science and Engineering, Pennsylvania State University
[4]Department of Computer Science, University of Texas at Austin
tongyang@mit.edu, cwang@psu.edu, {shouv,xwu}@cs.umd.edu

Abstract

We investigate sublinear classical and quantum algorithms for matrix games, a fundamental problem in optimization and machine learning, with provable guarantees. Given a matrix, sublinear algorithms for the matrix game were previously known only for two special cases: (1) the maximizing vectors live in the L1-norm unit ball, and (2) the minimizing vectors live in either the L1- or the L2-norm unit ball. We give a sublinear classical algorithm that can interpolate smoothly between these two cases: for any fixed q between 1 and 2, we solve, within some additive error, matrix games where the minimizing vectors are in an Lq-norm unit ball. We also provide a corresponding sublinear quantum algorithm that solves the same task with a quadratic improvement in dimensions of the maximizing and minimizing vectors. Both our classical and quantum algorithms are optimal in the dimension parameters up to poly-logarithmic factors. Finally, we propose sublinear classical and quantum algorithms for the approximate Carathéodory problem and the Lq-margin support vector machines as applications.

Introduction

Motivations. Minimax games between two parties, i.e., $\min_x \max_y f(x,y)$, is a basic model in game theory and has ubiquitous connections and applications to economics, optimization and machine learning, theoretical computer science, etc. Among minimax games, one of the most fundamental cases is the bilinear minimax game, also known as the *matrix game*, with the following form:

$$\min_{x\in\mathcal{X}} \max_{y\in\mathcal{Y}} y^\top Ax, \text{ where } A\in\mathbb{R}^{n\times d}, \mathcal{X}\subset\mathbb{R}^d, \mathcal{Y}\subset\mathbb{R}^n. \quad (1)$$

Matrix games are fundamental in algorithm design due to their equivalence to linear programs (Dantzig 1998), and also in machine learning because they contain classification (Novikoff 1963; Minsky and Papert 1988) as a special case, and many other important problems.

For many common domains $\mathcal{X}$ and $\mathcal{Y}$, matrix games can be solved efficiently within approximation error ϵ, i.e., to output $x'\in\mathcal{X}$ and $y'\in\mathcal{Y}$ such that $(y')^\top Ax'$ is ϵ-close to the optimum in (1). For some specific choices of $\mathcal{X}$ and $\mathcal{Y}$, the matrix game can even be solved in *sublinear time* in the size nd of A. When $\mathcal{X}$ and $\mathcal{Y}$ are both ℓ_1-norm unit balls, Grigoriadis and Khachiyan (1995) can solve the matrix game in time $O((n+d)\log(n+d)/\epsilon^2)$. When $\mathcal{X}$ is the ℓ_2-norm unit ball in $\mathbb{R}^d$ and $\mathcal{Y}$ is the ℓ_1-norm unit ball in $\mathbb{R}^n$, Clarkson, Hazan, and Woodruff (2012) can solve the matrix game in time $O((n+d)\log n/\epsilon^2)$.

As far as we know, the ℓ_1-ℓ_1 and ℓ_2-ℓ_1 matrix games are the only two cases where sublinear algorithms are known. However, there is general interest of solving matrix games with general norms. For instance, matrix games are closely related to the Carathéodory problem for finding a sparse linear combination in the convex hull of given data points, where all the ℓ_p-metrics with $p\geq 2$ have been well-studied (Barman 2015; Mirrokni et al. 2017; Combettes and Pokutta 2019). In addition, matrix games are common in machine learning especially support vector machines (SVMs), and general ℓ_p-margin SVMs have also been considered by previous literature, see e.g. the book by Deng, Tian, and Zhang (2012). In all, it is a natural question to investigate *sublinear algorithms for general matrix games*. In addition, quantum computing has been rapidly advancing and current technology has reached "quantum supremacy" for some specific tasks (Arute et al. 2019); since previous works have given sublinear quantum algorithms for ℓ_1-ℓ_1 matrix games (Li, Chakrabarti, and Wu 2019; Apeldoorn and Gilyén 2019) and ℓ_2-ℓ_1 matrix games (Li, Chakrabarti, and Wu 2019) with running time $(\sqrt{n}+\sqrt{d})\,\mathrm{poly}(1/\epsilon)$, it is also natural to explore *sublinear quantum algorithms for general matrix games*.

Contributions. We conduct a systematic study of ℓ_q-ℓ_1 matrix games for any $q\in(1,2]$ which corresponds to ℓ_q-margin SVMs and the ℓ_p-Carathéodory problem for any $p\geq 2$. We use the following entry-wise input model, the standard assumption in the sublinear algorithms in Grigoriadis and Khachiyan (1995); Clarkson, Hazan, and Woodruff (2012):

Input model: Given any $i\in[n]$ and $j\in[d]$, the j^{th} entry of A_i can be recovered in $O(1)$ time.

Quantumly, we consider an almost same oracle:

Quantum input model: Given any $i\in[n]$ and $j\in[d]$, the j^{th} entry of A_i can be recovered in $O(1)$ time *coherently*.

The only difference is to allow *coherent* queries, which

[*]Equal contribution. Full version of the paper (including the supplementary material) is at https://arxiv.org/abs/2012.06519

give quantum algorithms the ability to query different locations in superposition, and have been the *standard* quantization of the classical inputs and commonly adopted in previous works (Li, Chakrabarti, and Wu 2019; Apeldoorn and Gilyén 2019).

Theorem 1 (Main Theorem). *Given $q \in (1, 2]$. Define $p \geq 2$ such that $\frac{1}{p} + \frac{1}{q} = 1$. Consider the ℓ_q-ℓ_1 matrix game*[1]*:*

$$\sigma := \max_{x \in \mathbb{B}_q^d} \min_{\mathbf{p} \in \Delta_n} \mathbf{p}^\top A x, \tag{2}$$

where $\mathbb{B}_q^d$ is the ℓ_q-unit ball in $\mathbb{R}^d$ and Δ_n is the ℓ_1-simplex in $\mathbb{R}^n$. Then we can find an $\bar{x} \in \mathbb{B}_q^d$ s.t.[2]

$$\min_{i \in [n]} A_i \bar{x} \geq \sigma - \epsilon \tag{3}$$

with success probability at least $2/3$, using

- $O\big(\frac{(n+d)(p+\log n)}{\epsilon^2}\big)$ *classical queries (Theorem 2); or*
- $\tilde{O}\big(\frac{p^2\sqrt{n}}{\epsilon^4} + \frac{p^{3.5}\sqrt{d}}{\epsilon^7}\big)$ *quantum queries*[3] *(Theorem 3).*

When $p = \Omega(\log d/\epsilon)$, the above bounds can be improved (by Lemma 1) to respectively

- $O\big(\frac{(n+d)(\frac{\log d}{\epsilon}+\log n)}{\epsilon^2}\big)$ *queries to the classical input model;*
- $\tilde{O}\big(\frac{\sqrt{n}}{\epsilon^6} + \frac{\sqrt{d}}{\epsilon^{10.5}}\big)$ *queries to the quantum input model.*

Both results are optimal in n and d up to poly-log factors as we show $\Omega(n + d)$ and $\Omega(\sqrt{n} + \sqrt{d})$ classical and quantum lower bounds respectively when $\epsilon = \Theta(1)$ (Theorem 4).

Conceptually, our classical and quantum algorithms for general matrix games enjoy quite a few nice properties. On the one hand, they can be directly applied to

- **Convex geometry:** We give the *first* sublinear classical and quantum algorithms for the approximate Carathéodory problem (Corollary 1), improving the previous linear-time algorithms of Mirrokni et al. (2017); Combettes and Pokutta (2019);
- **Supervised learning:** We provide the *first* sublinear algorithms for general ℓ_q-margin support vector machines (SVMs) (Corollary 2).

On the other hand, our quantum algorithm is **friendly for near-term applications**. It uses the *standard quantum input model* and needs not to use any sophisticated quantum data structures. It is *classical-quantum hybrid* where the quantum part is isolated by pieces of state preparations connected by classical processing. Its output is completely *classical*.

Technique-wise, we are deeply inspired by Clarkson, Hazan, and Woodruff (2012), which serves as the starting point of our algorithm design. At a high level, Clarkson et al.'s algorithm follows a primal-dual framework where the primal part applies (ℓ_2-norm) online gradient descent (OGD) by Zinkevich (2003), and the dual part applies multiplicative weight updates (MWU) by ℓ_2-sampling. The choice of the ℓ_2-norm metric greatly facilitates the design and analysis of the algorithms for both parts. However, it is conceivable that *more sophisticated design and analysis* will be required to handle general ℓ_q-ℓ_1 matrix games.

Classically, our main technical contribution is to expand the primal-dual approach of Clarkson, Hazan, and Woodruff (2012) to work for more general metrics for the ℓ_q-ℓ_1 matrix game. Specifically, in the primal we replace OGD by a generalized p-norm OGD due to Shalev-Shwartz (2012), and in the dual we replace the ℓ_2-sampling by ℓ_q-sampling. We conduct a careful algorithm design and analysis to ensure that this strategy only incurs an $O(p/\epsilon^2)$ overhead in the number of iterations, and the error of the ℓ_q-ℓ_1 matrix game is still bounded by ϵ as in (3). In a nutshell, our algorithm can be viewed as an *interpolation* between the ℓ_2-ℓ_1 matrix game (Clarkson, Hazan, and Woodruff 2012) and the ℓ_1-ℓ_1 matrix game (Grigoriadis and Khachiyan 1995): when q is close to 2 the algorithm is more similar to Clarkson, Hazan, and Woodruff (2012), whereas when q is close to 1, p is large and the p-norm GD becomes closer to the normalized exponentiated gradient (Shalev-Shwartz 2012), which is exactly the update rule in Grigoriadis and Khachiyan (1995).

Quantumly, our **main contribution** is the systematic improvement of the previous quantum algorithm for ℓ_2-ℓ_1 matrix games by Li, Chakrabarti, and Wu (2019). They achieved a quantum speedup of $\tilde{O}(\sqrt{n} + \sqrt{d})$ for solving ℓ_2-ℓ_1 matrix games by levering *quantum amplitude amplification* and observing that ℓ_2-sampling can be readily accomplished by *quantum state preparation* as quantum states refer to ℓ_2 unit vectors. For general ℓ_q-ℓ_1 matrix game ($q \in (1, 2]$), we likewise upgrade both primal and dual parts as in our classical algorithm: specifically, in the primal, we apply the p-norm OGD in $\tilde{O}(\sqrt{d})$ time, whereas in the dual, we apply the multiplicative weight update via an ℓ_q-sampling in $\tilde{O}(\sqrt{n})$ time. To that end, we contribute to the following technical improvements, which may be of independent interest:

- In our algorithm, we cannot directly leverage quantum state preparation in the ℓ_q metric because it corresponds to ℓ_2-normalized vectors. Instead, we propose Algorithm 2 for **quantum ℓ_q-sampling** with $O(\sqrt{n})$ oracle calls which works with states whose amplitudes follow ℓ_q-norm proportion. Measuring such states is equivalent to performing ℓ_q-sampling.
- When $p = q = 2$, we improved the **ϵ-dependence** from the $1/\epsilon^8$ in the prior art by Li, Chakrabarti, and Wu (2019) to $1/\epsilon^7$. This is achieved by deriving a better upper bound on the entries of the vectors in the p-norm OGD (i.e., $y_{t,j}$ as in Eq. (38)); see the supplementary material (Eqs. (38)-(40)) for details.
- In our **lower bounds**, although the hard cases are motivated by Li, Chakrabarti, and Wu (2019), the matrix game values are much more complicated in the ℓ_q-ℓ_1 case. In the supplementary material, we figure out two functions f_1 and f_2 that not only separate the game values of two specifically-constructed ℓ_q-ℓ_1 matrix games but also have

[1] Throughout the paper, we use the bold font $\mathbf{p}$ to denote a vector and the math font p to denote a real number.

[2] $\bar{x} \in \mathbb{B}_q^d$ is the standard objective quantity under the ℓ_q-norm. Also note that once we have the $\bar{x}$ in (3), any $\mathbf{p} \in \Delta_n$ satisfies $\mathbf{p}^\top A\bar{x} \geq \sigma - \epsilon$.

[3] Here $\tilde{O}$ omits poly-logarithmic factors.

monotone and nonnegative properties, which are crucial factors in our proof.

These improvements together result in Theorem 1.

Related work. Matrix games were probably first studied as zero-sum games by Neumann (1928). The seminal work (Nemirovski and Yudin 1983) proposed the mirror descent method and gave an algorithm for solving matrix games in time $\tilde{O}(nd/\epsilon^2)$. This was later improved to $\tilde{O}(nd/\epsilon)$ by the prox-method due to Nemirovski (2004) and the dual extrapolation method due to Nesterov (2007). To further improve the cost, there have been two main focuses:

- Sampling-based methods: They focus on achieving sublinear cost in nd, the size of the matrix A. Grigoriadis and Khachiyan (1995); Clarkson, Hazan, and Woodruff (2012) mentioned above are seminal examples; these sublinear algorithms can also be used to solve semidefinite programs (Garber and Hazan 2011), SVMs (Hazan, Koren, and Srebro 2011), etc.
- Variance-reduced methods: They focus on the cost in $1/\epsilon$, in particular its decoupling with nd. Palaniappan and Bach (2016) showed how to apply the standard SVRG (Johnson and Zhang 2013) technique for solving ℓ_2-ℓ_2 matrix games; this idea can also be extended to smooth functions using general Bregman divergences (Shi, Zhang, and Yu 2017). Variance-reduced methods for solving matrix games culminate in Carmon et al. (2019), where they show how to solve ℓ_1-ℓ_1 and ℓ_2-ℓ_1 matrix games in time $\tilde{O}(\mathrm{nnz}(A)+\sqrt{\mathrm{nnz}(A)\cdot(n+d)}/\epsilon)$, where $\mathrm{nnz}(A)$ is the number of nonzero elements in A.

There have been relatively few quantum results for solving matrix games. Kapoor, Wiebe, and Svore (2016) solved the ℓ_2-ℓ_1 matrix game with cost $\tilde{O}(\sqrt{n}d/\epsilon^2)$ using an unusual input model where the representation of a data point in $\mathbb{R}^d$ is the concatenation of d floating point numbers. More recently, Apeldoorn and Gilyén (2019) was able to solve the ℓ_1-ℓ_1 matrix game with cost $\tilde{O}(\sqrt{n}/\epsilon^3+\sqrt{d}/\epsilon^3)$ using the standard input model above, and Li, Chakrabarti, and Wu (2019) solved the ℓ_2-ℓ_1 matrix game with cost $\tilde{O}(\sqrt{n}/\epsilon^4+\sqrt{d}/\epsilon^8)$ also using the standard input model.

Preliminaries and Notations

To facilitate the reading of this paper, we introduce necessary definitions and notations here.

Preliminaries for quantum computing. Quantum mechanics can be formulated in terms of linear algebra. For the space $\mathbb{C}^d$, we denote $\{\vec{e}_0,\dots,\vec{e}_{d-1}\}$ as its computational basis, where $\vec{e}_i=(0,\dots,1,\dots,0)^\top$ where 1 only appears in the $(i+1)^{\text{th}}$ coordinate. These basic vectors can be written by the *Dirac notation*: $\vec{e}_i := |i\rangle$ (called a "ket"), and $\vec{e}_i^\top := \langle i|$ (called a "bra"). A d-dimensional *quantum state* is a unit vector in $\mathbb{C}^d$: i.e., $|v\rangle=(v_0,\dots,v_{d-1})^\top$ such that $\sum_{i=0}^{d-1}|v_i|^2=1$.

Tensor product of quantum states is their Kronecker product: if $|u\rangle\in\mathbb{C}^{d_1}$ and $|v\rangle\in\mathbb{C}^{d_2}$, then

$$|u\rangle\otimes|v\rangle := (u_0v_0,u_0v_1,\dots,u_{d_1-1}v_{d_2-1})^\top, \qquad (4)$$

which is a vector in $\mathbb{C}^{d_1}\otimes\mathbb{C}^{d_2}$.

Quantum access to an input matrix, also known as a *quantum oracle*, is reversible and allows access to coordinates of the matrix in *superposition*, this is the essence of quantum speedups. In particular, to access entries of a matrix $A\in\mathbb{R}^{n\times d}$, we exploit a quantum oracle O_A, which is a unitary transformation on $\mathbb{C}^n\otimes\mathbb{C}^d\otimes\mathbb{C}^{d_{\text{acc}}}$ (d_{acc} being the dimension of a floating-point register) such that

$$O_A(|i\rangle\otimes|j\rangle\otimes|z\rangle)=|i\rangle\otimes|j\rangle\otimes|z\oplus A_{ij}\rangle \qquad (5)$$

for any $i\in[n]$, $j\in[d]$, and $z\in\mathbb{C}^{d_{\text{acc}}}$. Intuitively, O_A reads the entry A_{ij} and stores it in the third register as a floating-point number. However, to promise that O_A a unitary transformation, O_A applies the XOR operation ($\oplus$) on the third register. This is a natural generalization of classical reversible computation, when each entry of A can be recovered in $O(1)$ time. Subsequently, a common assumption is that a single query to O_A takes $O(1)$ cost.

Interpolation for large p. If p is large, we prove the following lemma showing that we can restrict without loss of generality to cases where p such that $\frac{1}{p}+\frac{1}{q}=1$ is $O(\log d/\epsilon)$, since in this case the ℓ_q-ℓ_1 matrix game is ϵ-close to the ℓ_1-ℓ_1 matrix game in the following sense:

Lemma 1. *An ℓ_q-ℓ_1 matrix game where p such that $\frac{1}{p}+\frac{1}{q}=1$ is greater than $\log d/\epsilon$ can be solved using an algorithm for solving ℓ_1-ℓ_1 games. This introduces an error $O(\epsilon)$ in the objective value.*

Proof. Assume without loss of generality that $\epsilon\le 1/2$. Let $p\ge\log d/\epsilon\ge\log d/(-\log(1-\epsilon))$. It can be easily verified that $\mathbb{B}_1^d\subset\mathbb{B}_q^d\subset\mathbb{B}_1^d+\left(1-d^{-1/p}\right)\mathbb{B}_q^d$. Thus $\mathbb{B}_q^d\subset\mathbb{B}_1^d+\epsilon\mathbb{B}_q^d$.

Consider applying an algorithm to solve an ℓ_1-ℓ_1 matrix game instead of the ℓ_q-ℓ_1 matrix game as required in (2). Let the optimal solution to (2) be $x^*\in\mathbb{B}_q^d, p^*\in\Delta_n$. By the previous analysis, there is a point $x\in\mathbb{B}_1^d$ such that $\|x-x^*\|_q\le\epsilon$. Thus the solution x,p^* has an error at most $O(\epsilon)$ from the true objective, and the algorithm for solving ℓ_1-ℓ_1 games finds a solution at least as good as this. □

Notations. Throughout the paper, we denote $p,q>1$ to be two real numbers such that $\frac{1}{p}+\frac{1}{q}=1$; $p\in[2,+\infty)$ and $q\in(1,2]$. For any $s>1$, we use $\mathbb{B}_s^d$ to denote the d-dimensional unit ball in ℓ_s-norm, i.e., $\mathbb{B}_s^d:=\{x:\sum_{i\in[d]}|x_i|^s\le 1\}$; we use Δ_n to denote the n-dimensional unit simplex $\{p\in\mathbb{R}^n: p_i\ge 0,\sum_i p_i=1\}$, and use $\mathbf{1}_n$ to denote the n-dimensional all-one vector. We denote $A\in\mathbb{R}^{n\times d}$ to be the matrix whose i^{th} row is $A_i^\top$ for all $i\in[n]$. We define $\mathrm{sgn}\colon\mathbb{R}\to\{-1,0,1\}$ such that $\mathrm{sgn}(x)=-1$ if $x<0$, $\mathrm{sgn}(x)=1$ if $x>0$, and $\mathrm{sgn}(0)=0$.

A Sublinear Classical Algorithm for General Matrix Games

For any $q \in (1,2]$, we consider the ℓ_q-ℓ_1 matrix game:

$$\sigma := \max_{x \in \mathbb{B}_q^d} \min_{\mathbf{p} \in \Delta_n} \mathbf{p}^\top A x. \tag{6}$$

The goal is to find a $\bar{x}$ that approximates the equilibrium of the matrix game within additive error ϵ:

$$\min_{i \in [n]} A_i \bar{x} \geq \sigma - \epsilon. \tag{7}$$

Throughout the paper, we assume $A_1, \ldots, A_n \in \mathbb{B}_p^d$, i.e., all the n data points are normalized to have ℓ_p-norm at most 1.

Algorithm 1: A sublinear algorithm for ℓ_q-ℓ_1 games.

Input: $\epsilon > 0$; $p \in [2, +\infty)$, $q \in (1,2]$ such that $\frac{1}{p} + \frac{1}{q} = 1$; $A \in \mathbb{R}^{n \times d}$ with $A_i \in \mathbb{B}_p^d \; \forall i \in [n]$.
Output: $\bar{x}$ that satisfies (7).

1 Let $T = \lceil \frac{895 \log n + 4p}{\epsilon^2} \rceil$, $y_1 = \mathbf{0}_d$, $\eta = \sqrt{\frac{11 \log n}{12T}}$, $w_1 = \mathbf{1}_n$;
2 **for** $t = 1$ **to** T **do**
3 $\quad p_t \leftarrow \frac{w_t}{\|w_t\|_1}$, $x_t \leftarrow \frac{y_t}{\max\{1, \|y_t\|_q\}}$;
4 $\quad$ Choose $i_t \in [n]$ by $i_t \leftarrow i$ with probability $p_t(i)$;
5 $\quad$ Define y_{t+1} where for any $j \in [d]$, $y_{t+1,j} \leftarrow y_t + \sqrt{\frac{q-1}{2T}} \frac{\operatorname{sgn}(A_{i_t,j}) |A_{i_t,j}|^{p-1}}{\|A_{i_t}\|_p^{p-2}}$;
6 $\quad$ Choose $j_t \in [d]$ by $j_t \leftarrow j$ with probability $\frac{x_t(j)^q}{\|x_t\|_q^q}$;
7 $\quad$ **for** $i = 1$ **to** n **do**
8 $\quad\quad \tilde{v}_t(i) \leftarrow A_i(j_t) \|x_t\|_q^q / x_t(j_t)^{q-1}$;
9 $\quad\quad v_t(i) \leftarrow \operatorname{clip}(\tilde{v}_t(i), \frac{1}{\eta})$ where $\operatorname{clip}(v, M) := \min\{M, \max\{-M, v\}\}$ $\forall v, M \in \mathbb{R}$;
10 $\quad\quad w_{t+1}(i) \leftarrow w_t(i)(1 - \eta v_t(i) + \eta^2 v_t(i)^2)$;
11 Return $\bar{x} = \frac{1}{T} \sum_{t=1}^T x_t$.

Theorem 2. *The output of Algorithm 1 satisfies (7) with probability at least $2/3$, and its total running time is $O(\frac{(n+d)(p+\log n)}{\epsilon^2})$ where $p \geq 2$ such that $\frac{1}{p} + \frac{1}{q} = 1$.*

Our sublinear algorithm follows the primal-dual approach of Algorithm 1 of Clarkson, Hazan, and Woodruff (2012), which solves ℓ_1-ℓ_2 matrix games. Here for ℓ_q-ℓ_1 matrix games, the solution vector x now lies in $\mathbb{B}_q^d$. Hence, the most natural adaptations are to use ℓ_q-sampling instead of ℓ_2-sampling in the primal updates, and to use a p-norm OGD by Shalev-Shwartz (2012) which generalizes the online gradient descent by Zinkevich (2003) in ℓ_2-norm. In the following, we use various technical tools to show these natural adaptations actually work.

Proposition 1 (Shalev-Shwartz 2012, Corollary 2.18)**.** *Consider a set of vectors $u_1, \ldots, u_T \in \mathbb{R}^d$ such that $\|u_i\|_p \leq 1$. Set $\iota = \sqrt{\frac{q-1}{2T}}$. Let $x_0 \leftarrow \mathbf{0}_d$, $\tilde{x}_{t+1,i} \leftarrow x_{t,i} + \iota \frac{\operatorname{sgn}(u_{t,i}) |u_{t,i}|^{p-1}}{\|u_t\|_p^{p-2}}$ for all $i \in [d]$, and $x_{t+1} \leftarrow \frac{\tilde{x}_{t+1}}{\max\{1, \|\tilde{x}_{t+1}\|_q\}}$. Then*

$$\max_{x \in \mathbb{B}_q^d} \sum_{t=1}^T u_t^\top x - \sum_{t=1}^T u_t^\top x_t \leq \sqrt{\frac{2T}{q-1}}. \tag{8}$$

The analysis of Algorithm 1 uses the following lemma, adapted from the variance multiplicative weight lemma and martingale tail bounds in Clarkson, Hazan, and Woodruff (2012)[4]:

Lemma 2 (Section 2 of Clarkson, Hazan, and Woodruff 2012)**.** *In Algorithm 1, the parameters p_t in Line 3 and v_t in Line 9 satisfy*

$$\sum_{t \in [T]} p_t^\top v_t \leq \min_{i \in [n]} \sum_{t \in [T]} v_t(i) + \eta \sum_{t \in [T]} p_t^\top v_t^2 + \frac{\log n}{\eta} \tag{9}$$

where v_t^2 is defined as $(v_t^2)_i := (v_t)_i^2$ for all $i \in [n]$, as long as the update rule of w_t is as in Line 10 and $\operatorname{Var}[v_t(i)^2] \leq 1$ for all $t \in [T]$ and $i \in [n]$. Furthermore, with probability at least $1 - O(1/n)$,

$$\max_{i \in [n]} \sum_{t \in [T]} \left[v_t(i) - A_i x_t \right] \leq 4\eta T; \tag{10}$$

$$\Big| \sum_{t \in [T]} A_{i_t} x_t - \sum_{t \in [T]} p_t^\top v_t \Big| \leq 10 \eta T, \tag{11}$$

with probability at least $5/7$, $\sum_{t \in [T]} p_t^\top v_t^2 \leq 7T$.

We also need to prove the following inequality on different moments of random variables.

Lemma 3. *Suppose that X is a random variable on $\mathbb{R}$, and $p \geq 2$. If $\mathbb{E}[|X|^p] \leq 1$, then $\mathbb{E}[X^2] \leq 1$.*

Proof. Denote the probability density of X as μ. Then $\int_{-\infty}^{+\infty} |x|^p \mathrm{d}\mu_x = \mathbb{E}[|X|^p] \leq 1$. By Hölder's inequality, we have

$$\begin{aligned} 1 &\geq \Big(\int_{-\infty}^{+\infty} |x|^p \mathrm{d}\mu_x \Big)^{2/p} \Big(\int_{-\infty}^{+\infty} 1 \mathrm{d}\mu_x \Big)^{1-2/p} \\ &\geq \int_{-\infty}^{+\infty} |x|^2 \cdot 1^{1-2/p} \mathrm{d}\mu_x = \int_{-\infty}^{+\infty} x^2 \mathrm{d}\mu_x, \end{aligned} \tag{12}$$

hence the result follows. □

Now we are ready to prove our main theorem.

Proof of Theorem 2. First, $\tilde{v}_t(i)$ is an unbiased estimator of $A_i x_t$ as

$$\mathbb{E}[\tilde{v}_t(i)] = \sum_{j_t=1}^d \frac{x_t(j_t)^q}{\|x_t\|_q^q} \cdot \frac{A_i(j_t) \|x_t\|_q^q}{x_t(j_t)^{q-1}} = A_i x_t. \tag{13}$$

[4] The proof follows from the proofs of Lemmas 2.3, 2.4, 2.5, and 2.6 in Section 2 and Appendix B of Clarkson, Hazan, and Woodruff (2012), with only small modifications to fit our new parameter choices. For instance, the original statement requires that $\eta \geq \sqrt{\frac{\log n}{T}}$, but the proofs actually work for $\eta \geq \sqrt{\frac{11 \log n}{12T}}$.

Furthermore,

$$\mathbb{E}[|\tilde{v}_t(i)|^p] = \sum_{j_t=1}^{d} \frac{x_t(j_t)^q}{\|x_t\|_q^q} \cdot \frac{|A_i(j_t)|^p \|x_t\|_q^{pq}}{x_t(j_t)^{p(q-1)}} = \|A_i\|_p^p \|x_t\|_q^p \le 1, \quad (14)$$

where the second equality follows from the identities $q = p(q-1)$ and $p = q(p-1)$, and the last inequality follows from the assumption that $A_i \in \mathbb{B}_p^d \; \forall i \in [n]$. By Lemma 3, $\mathbb{E}[\tilde{v}_t(i)^2] \le 1$. Because the clip function in Line 9 only makes variance smaller, this means that the conditions of Lemma 2 are satisfied and we hence have (9), rewritten below:

$$\sum_{t\in[T]} p_t^\top v_t \le \min_{i\in[n]} \sum_{t\in[T]} v_t(i) + \eta \sum_{t\in[T]} p_t^\top v_t^2 + \frac{\log n}{\eta}. \quad (15)$$

Furthermore, Lemma 2 implies that with probability $5/7 - O(1/n)$ we have

$$\sum_{t\in[T]} A_{i_t} x_t \le \min_{i\in[n]} \sum_{t\in[T]} v_t(i) + 17\eta T + \frac{\log n}{\eta}. \quad (16)$$

Moreover, (10) gives $\sum_{t\in[T]} \big[v_t(i) - A_i x_t\big] \le 4\eta T$, and hence $\min_{i\in[n]} \sum_{t\in[T]} v_t(i) \le 4\eta T + \min_{i\in[n]} \sum_{t\in[T]} A_i x_t$. Plugging this into (16), we have

$$\sum_{t\in[T]} A_{i_t} x_t \le \sum_{t\in[T]} p_t^\top v_t + 10\eta T \le \min_{i\in[n]} \sum_{t\in[T]} A_i x_t + 21\eta T + \frac{\log n}{\eta}, \quad (17)$$

with probability $(5/7 - O(1/n)) \cdot (1 - O(1/n)) \ge 2/3$.

On the other hand, by taking $u_t = A_{i_t}$ in Proposition 1,

$$T\sigma \le \max_{x\in\mathbb{B}_q^d} \sum_{t=1}^{T} A_{i_t} x \le \sum_{t=1}^{T} A_{i_t} x_t + \sqrt{2Tp}, \quad (18)$$

since $\frac{1}{q-1} = \frac{p}{q} \le p$. Combining (17) and (18), we have

$$\min_{i\in[n]} \sum_{t\in[T]} A_i x_t \ge T\sigma - \sqrt{2Tp} - 21\eta T - \frac{\log n}{\eta}. \quad (19)$$

Consequently, the return $\bar{x} = \frac{1}{T}\sum_{t=1}^{T} x_t$ of Algorithm 1 in Line 11 satisfies

$$\min_{i\in[n]} A_i \bar{x} \ge \sigma - \sqrt{\frac{2p}{T}} - 21\eta - \frac{\log n}{\eta T}. \quad (20)$$

To prove (7), it remains to show that $\sqrt{\frac{2p}{T}} + 21\eta + \frac{\log n}{\eta T} \le \epsilon$, which is equivalent to $\sqrt{2p} + 21\sqrt{\frac{11\log n}{12}} + \sqrt{\frac{12\log n}{11}} \le \sqrt{T}\epsilon$ by the definition of η. This is true because the AM-GM inequality implies that that LHS is at most $2(\sqrt{2p})^2 + 2\big(21\sqrt{\frac{11\log n}{12}} + \sqrt{\frac{12\log n}{11}}\big)^2 \le 4p + 895\log n \le T\epsilon^2$. □

Lemma 1 combined with Theorem 2 yields the classical result in Theorem 1.

A Sublinear Quantum Algorithm for General Matrix Games

In this section, we give a quantum algorithm for solving the general ℓ_q-ℓ_1 matrix games. It closely follows our classical algorithm because they both use a primal-dual approach, where the primal part is composed of p-norm online gradient descent and the dual part is composed of multiplicative weight updates. However, we adopt quantum techniques to achieve speedup on both.

The intuition behind the quantum algorithm and the quantum speedup is that we measure quantum states to obtain random samples. These quantum states can be efficiently prepared (with cost $\tilde{O}(\sqrt{n})$ and $\tilde{O}(\sqrt{d})$). Mathematically, A quantum state can be represented by an ℓ_2-normalized complex vector ψ in the sense that measuring this quantum states yields outcome i with probability $|\psi_i|^2$ (thus for every probability distribution there is a quantum state corresponding to it). Let us denote the quantum state for sampling from w by $|w\rangle$ and the quantum state for sampling from x by $|x\rangle$ (different from the notation in Algorithm 3). If we can maintain $|w\rangle$ and $|x\rangle$ in each iteration, then there is no need for classical updates, and preparing $|w\rangle$ and $|x\rangle$ becomes the bottleneck of the quantum algorithm.

The source of our quantum speedup comes from an important subroutine, Algorithm 2, which is designed to prepare states for ℓ_q-sampling. It uses standard Grover-based techniques to prepare states but we carefully keep track of the normalizing factor to facilitate ℓ_q-sampling. We showed (in Proposition 2 in the supplementary material) that preparing $|w\rangle$ costs $\tilde{O}(\sqrt{n})$ and preparing $|x\rangle$ costs $\tilde{O}(\sqrt{d})$. In the following, we give the high-level ideas of Algorithm 2.

1. We first create a quantum state corresponding to the uniform distribution, which is easy using Hadamard gates.
2. For each entry, we create a state with the desired amplitude associated with 0, and an undesired amplitude associated to 1 (the unitarity of quantum operations necessitates the existence of this undesired term).
3. Finally we use a technique called amplitude amplification to amplify the portion of the state corresponding to 0 for each entry, to get a state with only the desired amplitudes.

The details of our quantum algorithm for solving the general ℓ_q-ℓ_1 matrix games, Algorithm 3, is rather technical. To simplify the presentation, we postpone its pseudocode (Algorithm 3) to the supplementary material and highlight how it is different from Algorithm 1 in the following.

- For the primal part, we prepare a quantum state $|y_t\rangle$ for the q-norm OGD and measure it (in Line 7) to obtain a sample $j_t \in [d]$. The subtlety here is that we need to perform the ℓ_q-sampling to the vector y_t; this is different from the ℓ_2-sampling in Li, Chakrabarti, and Wu (2019) which uses the fact that pure quantum states are ℓ_2-normalized. To this end, we design Algorithm 2 for *ℓ_q-quantum state sampling*, which may be of independent interest; this algorithm is built upon a clever use of *quantum amplitude amplification*, the technique behind the Grover search (Grover 1996). Note that sampling according to y_t is equivalent to sampling according to x_t

Algorithm 2: Prepare an ℓ_q-pure state given an oracle to its coefficients.

1 Find $a_{\|q\|} := \max_{i\in[n]} |a_i|^{q/2}$ in $O(\sqrt{n})$ time by the minimum-finding algorithm (Dürr and Høyer 1996);

2 Prepare the uniform superposition $\frac{1}{\sqrt{n}}\sum_{r\in[n]} |i\rangle$;

3 Perform the following unitary transformations:

$$\begin{aligned}
&\frac{1}{\sqrt{n}}\sum_{i\in[n]} |i\rangle \xmapsto{O_a} \frac{1}{\sqrt{n}}\sum_{i\in[n]} |i\rangle|a_i\rangle \\
&\mapsto \frac{1}{\sqrt{n}}\sum_{i\in[n]} |i\rangle|a_i\rangle \left(\frac{a_i^{q/2}}{a_{\|q\|}}|0\rangle + \sqrt{1-\frac{|a_i|^q}{a_{\|q\|}^2}}|1\rangle\right) \\
&\xmapsto{O_a^{-1}} \frac{1}{\sqrt{n}}\sum_{i\in[n]} |i\rangle|0\rangle \left(\frac{a_i^{q/2}}{a_{\|q\|}}|0\rangle + \sqrt{1-\frac{|a_i|^q}{a_{\|q\|}^2}}|1\rangle\right);
\end{aligned}$$

4

5 Discard the second register above; rewrite the state as

$$\frac{\|a\|_q^{q/2}}{\sqrt{n}a_{\|q\|}}\left(\frac{1}{\|a\|_q^{q/2}}\sum_{i\in[n]} a_i^{q/2}|i\rangle\right)|0\rangle + |a^{\perp}\rangle|1\rangle, \tag{21}$$

where $|a^{\perp}\rangle := \frac{1}{\sqrt{n}}\sum_{i\in[n]}\sqrt{1-\frac{|a_i|^q}{a_{\|q\|}^2}}|i\rangle$;

6 Apply amplitude amplification (Brassard et al. 2002) for the state in (21) conditioned on the second register being 0. Return the output.

in Algorithm 1, because $x_t(j)^q/\|x_t\|_q^q = y_t(j)^q/\|y_t\|_q^q$. Moreover, it suffices to replace $\|x_t\|_q^q/x_t(j_t)^{q-1}$ with $\|y_t\|_q^q/(y_t(j_t)^{q-1}\max\{1,\|y_t\|_q\})$ in Line 8 of Algorithm 3. Similar to preparing $|p_t\rangle$, we use $\tilde{O}(\sqrt{d})$ queries to O_A to prepare y_t, while classically we need to compute all the entries of y_t, which takes $O(d)$ queries.

- For the dual part, we prepare the multiplicative weight vector as a quantum state $|p_t\rangle$ and measure it (in Line 3) to obtain a sample $i_t \in [n]$. This adaption enables us to achieve the $\tilde{O}(\sqrt{n})$ dependence by using quantum amplitude amplification in the quantum state preparation: in Line 8, we implement the oracle O_t and in Line 9 we use $\tilde{O}(\sqrt{n})$ queries to O_t to prepare the state $|p_{t+1}\rangle$ for the next iteration. In contrast, classically we need to compute all the entries of w_{t+1} to obtain the probability distribution p_{t+1} for the next iteration, which takes $O(n)$ queries.

In general, Algorithm 3 can be viewed as a template for achieving quantum speedups for online mirror descent methods: In this work, we focus on the general matrix games where the primal and dual are in the special relationship of ℓ_p and ℓ_q norms, but in principle it may be applicable to study other dualities in online learning.

We summarize the main quantum result as the following theorem, which states the correctness and time complexity of Algorithm 3. The relevant technical proofs are deferred to the supplementary material.

Theorem 3. *Algorithm 3 returns a succinct classical representation*[5] *of a vector $\bar{w}\in\mathbb{R}^d$ such that*

$$A_i\bar{x} \geq \max_{x\in\mathbb{B}_q^d}\min_{i'\in[n]} A_{i'}x - \epsilon \quad \forall i\in[n], \tag{22}$$

with probability at least $2/3$, and its total running time is $\tilde{O}\big(\frac{p^2\sqrt{n}}{\epsilon^4} + \frac{p^{3.5}\sqrt{d}}{\epsilon^7}\big)$. We can also assume $p = O(\log d/\epsilon)$ (Lemma 1) and result in running time $\tilde{O}\big(\frac{\sqrt{n}}{\epsilon^6} + \frac{\sqrt{d}}{\epsilon^{10.5}}\big)$.

Moreover, Algorithm 3 enjoys the following features:

- **Simple quantum input:** Algorithm 3 uses the standard quantum input model and needs not to use any sophisticated quantum data structures, such as quantum random access memory (QRAM) in some other quantum machine learning applications, to achieve speedups.
- **Hybrid classical-quantum feature:** Algorithm 3 is also highly classical-quantum hybrid: the quantum part is isolated by pieces of state preparations connected by classical processing. In addition, it only has $O(\frac{\log n+p}{\epsilon^2})$ iterations, which implies that the corresponding quantum circuit is shallow and can potentially be implemented even on near-term quantum machines (Preskill 2018).
- **Classical output:** The output of Theorem 3 is completely *classical*. Compared to quantum algorithms whose output is a quantum state and may incur overheads (Aaronson 2015), Algorithm 3 guarantees minimal overheads and can be directly used for classical applications.

Applications

We give two applications that generically follow from our classical and quantum ℓ_q-ℓ_1 matrix game solvers.

Approximate Carathéodory Problem

The exact Carathéodory problem is a fundamental result in linear algebra and convex geometry: every point $u\in\mathbb{R}^d$ in the convex hull of a vertex set $S\subset\mathbb{R}^d$ can be expressed as a convex combination of $d+1$ vertices in S. Recently, a breakthrough result by Barman (2015) shows that if $S\subset\mathbb{B}_p^d$, i.e., S is in the ℓ_p-norm unit ball, then there exists a point u' s.t. $\|u-u'\|_p\leq\epsilon$ and u' is a convex combination of $O(p/\epsilon^2)$ vertices in S. The follow-up work by Mirrokni et al. (2017) proved a matching lower bound $\Omega(p/\epsilon^2)$, and Combettes and Pokutta (2019) can give better bounds under stronger assumptions on S or u.

Currently, the best-known time complexity of solving the approximate Carathéodory problem is $O(ndp/\epsilon^2)$ by Theorem 3.5 of Mirrokni et al. (2017). We give classical and quantum sublinear algorithms:

Corollary 1. *Suppose that $S\subset\mathbb{B}_p^d$, $|S|=n$, and u is in the convex full of S. Then we can find a convex combination $\sum_{i=1}^k x_iv_i$ such that $v_i\in S$ for all $i\in[k]$,*

[5] The algorithm stores $T=\tilde{O}(p/\epsilon^2)$ real numbers classically: $i_1,\ldots,i_T$ obtained from Line 3 and $\widetilde{\|y_1\|}_q,\ldots,\widetilde{\|y_T\|}_q$ obtained from Line 6. After that, each coordinate of $\bar{x}$ can be computed in time $\tilde{O}(p/\epsilon^2)$.

$k = O((p + \log n)/\epsilon^2)$, *and* $\| \sum_{i=1}^{k} x_i v_i \|_p \leq \epsilon$, *using a classical algorithm with running time* $O\big(\frac{(n+d)(p+\log n)}{\epsilon^2}\big)$ *or a quantum algorithm with running time* $\tilde{O}\big(\frac{p^2\sqrt{n}}{\epsilon^4} + \frac{p^{3.5}\sqrt{d}}{\epsilon^7}\big)$. *We can also assume* $p = O(\log d/\epsilon)$ *(Lemma 1) and result in running time* $O\big(\frac{(n+d)(\log d/\epsilon + \log n)}{\epsilon^2}\big)$ *and* $\tilde{O}\big(\frac{\sqrt{n}}{\epsilon^6} + \frac{\sqrt{d}}{\epsilon^{10.5}}\big)$, *respectively.*

Proof. We denote the matrix $V := (v_1; v_2; \cdots; v_n)$ where v_i is the i^{th} element in S. Note that the approximate Carathéodory problem can be formed as $\min_{\mathbf{p}\in\Delta_n} \|V^\top \mathbf{p} - u\|_p$. In addition, by Hölder's inequality $\|y\|_p = \max_{x:\|x\|_q \leq 1} y^\top x$; therefore, we obtain the following minimax matrix game:

$$\min_{\mathbf{p}\in\Delta_n} \max_{x\in\mathbb{B}_q^d} (\mathbf{p}^\top V - u^\top)x. \tag{23}$$

We denote $U = (u; u; \cdots; u) \in \mathbb{R}^{n\times d}$, i.e., all the n rows of U are u. Then we have $(\mathbf{p}^\top V - u^\top)x = 2\mathbf{p}^\top \frac{V-U}{2} x$. Furthermore, since $u, v_i \in \mathbb{B}_p^d$ for all $i \in [n]$, each row of $\frac{V-U}{2}$ is also in $u, v_i \in \mathbb{B}_p^d$. Finally, by the Sion's Theorem (Sion 1958) we can switch the order of the $\min$ and $\max$ in (23). In all, to solve the approximate Carathéodory problem with precision ϵ, it suffices to solve the maximin game

$$\max_{x\in\mathbb{B}_q^d} \min_{\mathbf{p}\in\Delta_n} \mathbf{p}^\top \frac{V-U}{2} x \tag{24}$$

with precision $\frac{\epsilon}{2}$. This is exactly (6), thus the result follows from Theorem 2 and Theorem 3. □

Compared to Mirrokni et al. (2017), we pay a $\log n$ overhead in the cardinality of the convex combination, but in time complexity the dominating term nd is significantly improved to $n + d$. We also give the first sublinear quantum algorithm. Note that as Mirrokni et al. (2017) pointed out, the approximate Carathéodory problem has wide applications in machine learning and optimization, including support vector machines (SVMs), rounding in polytopes, submodular function minimization, etc. We elaborate the details of SVMs below, and leave out the details of other applications as the reductions are direct.

The ℓ_q-Margin Support Vector Machine (SVM)

When we solve the ℓ_q-ℓ_1 matrix game in Algorithm 1, we apply ℓ_q-sampling where $j_t = j$ with probability $x(j)^q/\|x\|_q^q$ for any $j \in [d]$. The key reason of the success of Algorithm 1 is because the expectation of the random variable $A_i(j_t)\|x_t\|_q^q / x_t(j_t)^{q-1}$ in Line 8 is $A_i x$, which is unbiased.

If we consider some alternate random variables, we can potentially solve a maximin game in ℓ_q-ℓ_1 norm with respect to some nonlinear functions of the matrix. A specific problem of significant interest is the ℓ_q-margin support vector machine (SVM), where we are given n data points $X_1, \ldots, X_n$ in $\mathbb{R}^d$ and a label vector $y \in \{1, -1\}^n$. The goal is to find a separating hyperplane $w \in \mathbb{R}^d$ of these data points with the largest margin under the ℓ_q-norm loss, i.e.,

$$\sigma_{\text{SVM}} := \max_{w\in\mathbb{R}^d} \min_{i\in[n]} 2y_i \cdot X_i^\top w - \|w\|_q^q. \tag{25}$$

Without loss of generality, we assume $y_i = 1$ for all $i \in [n]$, otherwise we take $X_i \leftarrow (-1)^{y_i} \cdot X_i$. In this case, the random variable $2X_i(j)\|w\|_q^q / w(j)^{q-1} - \|w\|_q^q$ is unbiased under ℓ_q-sampling on j:

$$\mathbb{E}\Big[\frac{2X_i(j)\|w\|_q^q}{w(j)^{q-1}} - \|w\|_q^q\Big] = 2X_i^\top w - \|w\|_q^q. \tag{26}$$

Note that $\sigma_{\text{SVM}} \geq 0$ since $2X_i^\top w - \|w\|_q^q = 0$ for all $i \in [n]$ when $w = 0$. For the case $\sigma_{\text{SVM}} > 0$ and taking $0 < \epsilon < \sigma_{\text{SVM}}$, similar to Theorem 2 and Theorem 3 we have:

Corollary 2. *To return a vector* $\bar{w} \in \mathbb{R}^d$ *such that with probability at least* $2/3$,

$$\min_{i\in[n]} 2X_i\bar{w} - \|\bar{w}\|_q^q \geq \sigma_{\text{SVM}} - \epsilon > 0, \tag{27}$$

there is a classical algorithm that achieves this with $O\big(\frac{(n+d)(p+\log n)}{\epsilon^2}\big)$ *time and a quantum algorithm that achieves this with* $\tilde{O}\big(\frac{p^2\sqrt{n}}{\epsilon^4} + \frac{p^{3.5}\sqrt{d}}{\epsilon^7}\big)$ *time. We can also assume* $p = O(\log d/\epsilon)$ *(Lemma 1) and result in running time* $O\big(\frac{(n+d)(\log d/\epsilon + \log n)}{\epsilon^2}\big)$ *and* $\tilde{O}\big(\frac{\sqrt{n}}{\epsilon^6} + \frac{\sqrt{d}}{\epsilon^{10.5}}\big)$, *respectively.*

Notice that classical sublinear algorithms for ℓ_2-SVMs have been given (Clarkson, Hazan, and Woodruff 2012; Hazan, Koren, and Srebro 2011), and there is also a sublinear quantum algorithm for ℓ_2-SVMs in Li, Chakrabarti, and Wu (2019). We essentially generalize their results to the l_q-norm cases based on our new general matrix game solvers in Theorem 2 and Theorem 3.

Classical and Quantum Lower Bounds

For both our classical and quantum algorithms for general matrix games, we can prove matching classical and quantum lower bounds in n and d for constant ϵ:

Theorem 4. *Assume* $0 < \epsilon < 0.04$. *Then to return an* $\bar{x} \in \mathbb{B}_q^d$ *satisfying*

$$A_j\bar{x} \geq \max_{x\in\mathbb{B}_q^d} \min_{i\in[n]} A_i x - \epsilon \quad \forall\, j \in [n], \tag{28}$$

with probability at least $2/3$, *we need* $\Omega(n + d)$ *classical queries or* $\Omega(\sqrt{n} + \sqrt{d})$ *quantum queries.*

Due to the space limitation, we postpone the proof details of Theorem 4 to the supplementary material.

Conclusions

We give sublinear algorithms for solving general ℓ_q-ℓ_1 matrix games for any $q \in (1, 2]$. Our classical and algorithms run in time $O(\frac{(n+d)(p+\log n)}{\epsilon^2})$ and $\tilde{O}(\frac{p^2\sqrt{n}}{\epsilon^4} + \frac{p^{3.5}\sqrt{d}}{\epsilon^7})$, respectively; both bounds are tight up to poly-logarithmic factors in n and d. Our results can be applied to solve the approximate Carathéodory problem and the ℓ_q-margin SVMs.

Our paper raises a couple of natural open questions for future work. For instance:

- Can we give sublinear algorithms for ℓ_p-ℓ_1 matrix games where $p > 2$? Technically, this will probably require a q^{th} moment multiplicative weight lemma to replace Lemma 2.
- Can we give quantum algorithms that achieve speedup of variance-reduced methods for solving matrix games, such as the state-of-the-art result in Carmon et al. (2019)?

Acknowledgements

TL thanks Adrian Vladu for many helpful discussions, as well as Yair Carmon for the discussions about his paper (Carmon et al. 2019). TL was supported by an IBM PhD Fellowship, an QISE-NET Triplet Award (NSF grant DMR-1747426), the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Quantum Algorithms Teams program, ARO contract W911NF-17-1-0433, and NSF grant PHY-1818914. CW was supported by Scott Aaronson's Vannevar Bush Faculty Fellowship. SC and XW were partially supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Quantum Algorithms Team program, and were also partially supported by the U.S. National Science Foundation grant CCF-1755800, CCF-1816695, and CCF-1942837 (CAREER).

Ethics Statement

This work is purely theoretical. Researchers working on learning theory and quantum computing may benefit from our results. In the long term, once fault-tolerant quantum computers have been built, our results may find practical applications in matrix game scenarios arising in the real world. As far as we are aware, our work does not have immediate negative ethical impact.

References

Aaronson, S. 2015. Read the fine print. *Nature Physics* 11(4): 291.

Apeldoorn, J. v.; and Gilyén, A. 2019. Quantum algorithms for zero-sum games. arXiv:1904.03180

Arute et al., F. 2019. Quantum supremacy using a programmable superconducting processor. *Nature* 574(7779): 505–510. arXiv:1910.11333

Barman, S. 2015. Approximating Nash equilibria and dense bipartite subgraphs via an approximate version of Carathéodory theorem. In *Proceedings of the 47th Annual ACM Symposium on Theory of Computing*, 361–369. arXiv:1406.2296

Bennett, C. H.; Bernstein, E.; Brassard, G.; and Vazirani, U. 1997. Strengths and weaknesses of quantum computing. *SIAM Journal on Computing* 26(5): 1510–1523. arXiv:quant-ph/9701001

Brassard, G.; Høyer, P.; Mosca, M.; and Tapp, A. 2002. Quantum amplitude amplification and estimation. *Contemporary Mathematics* 305: 53–74. arXiv:quant-ph/0005055

Carmon, Y.; Jin, Y.; Sidford, A.; and Tian, K. 2019. Variance reduction for matrix games. In *Advances in Neural Information Processing Systems*, 11377–11388. arXiv:1907.02056

Clarkson, K. L.; Hazan, E.; and Woodruff, D. P. 2012. Sublinear optimization for machine learning. *Journal of the ACM (JACM)* 59(5): 23. arXiv:1010.4408

Combettes, C. W.; and Pokutta, S. 2019. Revisiting the Approximate Carathéodory Problem via the Frank-Wolfe Algorithm. arXiv:1911.04415

Dantzig, G. B. 1998. *Linear programming and extensions*, volume 48. Princeton University Press.

Deng, N.; Tian, Y.; and Zhang, C. 2012. *Support vector machines: optimization based theory, algorithms, and extensions*. CRC press.

Dürr, C.; and Høyer, P. 1996. A quantum algorithm for finding the minimum. arXiv:quant-ph/9607014

Garber, D.; and Hazan, E. 2011. Approximating semidefinite programs in sublinear time. In *Advances in Neural Information Processing Systems*, 1080–1088.

Grigoriadis, M. D.; and Khachiyan, L. G. 1995. A sublinear-time randomized approximation algorithm for matrix games. *Operations Research Letters* 18(2): 53–58.

Grover, L. K. 1996. A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing*, 212–219. ACM. arXiv:quant-ph/9605043

Hazan, E.; Koren, T.; and Srebro, N. 2011. Beating SGD: Learning SVMs in sublinear time. In *Advances in Neural Information Processing Systems*, 1233–1241.

Johnson, R.; and Zhang, T. 2013. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, 315–323.

Kapoor, A.; Wiebe, N.; and Svore, K. 2016. Quantum perceptron models. In *Proceedings of the 30th Conference on Neural Information Processing Systems*, 3999–4007. arXiv:1602.04799

Li, T.; Chakrabarti, S.; and Wu, X. 2019. Sublinear quantum algorithms for training linear and kernel-based classifiers. In *Proceedings of the 36th International Conference on Machine Learning*, 3815–3824. arXiv:1904.02276

Minsky, M.; and Papert, S. A. 1988. *Perceptrons: An introduction to computational geometry*. MIT Press.

Mirrokni, V.; Leme, R. P.; Vladu, A.; and Wong, S. C.-w. 2017. Tight bounds for approximate Carathéodory and beyond. In *Proceedings of the 34th International Conference on Machine Learning*, 2440–2448. arXiv:1512.08602

Nemirovski, A. 2004. Prox-method with rate of convergence $O(1/t)$ for variational inequalities with Lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization* 15(1): 229–251.

Nemirovski, A. S.; and Yudin, D. B. 1983. Problem complexity and method efficiency in optimization .

Nesterov, Y. 2007. Dual extrapolation and its applications to solving variational inequalities and related problems. *Mathematical Programming* 109(2-3): 319–344.

Neumann, J. v. 1928. Zur theorie der gesellschaftsspiele. *Mathematische Annalen* 100(1): 295–320.

Novikoff, A. B. 1963. On convergence proofs for perceptrons. In *Proceedings of the Symposium on the Mathematical Theory of Automata*, volume 12, 615–622.

Palaniappan, B.; and Bach, F. 2016. Stochastic variance reduction methods for saddle-point problems. In *Advances*

in Neural Information Processing Systems, 1416–1424. arXiv:1605.06398

Preskill, J. 2018. Quantum Computing in the NISQ era and beyond. *Quantum* 2: 79. ISSN 2521-327X. arXiv:1801.00862

Shalev-Shwartz, S. 2012. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning* 4(2): 107–194.

Shi, Z.; Zhang, X.; and Yu, Y. 2017. Bregman divergence for stochastic variance reduction: saddle-point and adversarial prediction. In *Advances in Neural Information Processing Systems*, 6031–6041.

Sion, M. 1958. On general minimax theorems. *Pacific Journal of Mathematics* 8(1): 171–176.

Zinkevich, M. 2003. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning*, 928–936.