PROVABLE SAMPLE COMPLEXITY GUARANTEES FOR LEARNING OF CONTINUOUS-ACTION GRAPHICAL GAMES WITH NONPARAMETRIC UTILITIES

Adarsh Barik, Jean Honorio

Purdue University
Department of Computer Science
West Lafayette, Indiana, USA

ABSTRACT

In this paper, we study the problem of learning the exact structure of continuous-action games with non-parametric utility functions. We propose an ℓ_1-regularized method which encourages sparsity of the coefficients of the Fourier transform of the recovered utilities. Our method works by accessing very few Nash equilibria and their noisy utilities. Under certain technical conditions, our method also recovers the exact structure of these utility functions, and thus, the exact structure of the game. Furthermore, our method only needs a logarithmic number of samples in terms of the number of players and runs in polynomial time. We follow the primal-dual witness framework to provide provable theoretical guarantees. *A full version of this paper is accessible at:* `https://www.cs.purdue.edu/homes/abarik/abarik_nonpara_icassp_full.pdf`

***Index Terms*—** Graphical Games, Machine Learning, Continuous Action Games, Non-Parametric Payoffs

1. INTRODUCTION

Game theory has been extensively used as a framework to model and study the strategic interactions amongst rational but selfish individual players who are trying to maximize their payoffs. Game theory has been applied in many fields including but not limited to social and political science, economics, communication, system design and computer science. In non-cooperative games, each player decides its action based on the actions of others players. These games are characterized by the equilibrium solution concept such as *Nash equilibrium (NE)* [1] which serves a descriptive role of the stable outcome of the overall behavior of self-interested players (e.g., people, companies, governments, groups or autonomous systems) interacting strategically with each other in distributed settings.

Graphical games [2], introduced within the AI community about two decades ago, are a representation of multiplayer games which capture and exploit locality or sparsity of direct influences. They are most appropriate for large-scale population games in which the payoffs of each player are determined by the actions of only a small number of other players. Indeed, graphical games played a prominent role in establishing the computational complexity of computing NE in normal-form games as well as in succinctly representable multiplayer games [3–5]. Graphical games have been studied for both discrete and continuous actions.

Inference in Graphical Games. There has been a large body of work on computing classical equilibrium solution concepts. The Nash equilibria and *correlated equilibria* [6] in graphical games have been studied by [2, 7–16]. The computation of the *price of anarchy* was studied in [17]. In addition, [18] identified the most influential players, i.e., a small set of players whose collective behavior forces every other player to a unique choice of action. All the works above focus on inference problems for graphical games, and fall in the field of algorithmic game theory. All of these methods assume access to graphical game network and payoffs of the games under consideration.

Learning Graphical Games. In order to answer the *inference* problems discussed above, we would need to recover the structure of graphical game. In particular we ask: Given that we have access to a noisy observation of the player's revenue and few joint action vectors at equilibria, can we learn the neighbors of a player? Of course, once we learn the local structure for each individual player then we can combine them in order to obtain the entire structure of the graphical game. Learning the structure of a game is essential to the development, potential use and success of game-theoretic models in large-scale practical applications. It is also interesting to study the relationship between the pure strategy Nash equilibria (PSNE) set of the recovered game and the true game. In this paper, we study the problem of learning the graph structure in a continuous-action graphical game with non-parametric utility functions. We also provide the complete characterization and comparison of the payoff functions and the PSNE set of the recovered game and the true game.

A Motivating Real-World Case. To understand the difference between learning and inference, we provide a real world example of potato trade among various countries between 1973 to 2019. In this case, it is necessary to learn the structure of the graphical game before we can infer global efficiency quantities such as the price of anarchy (See subsection A.8 for details). We divided dataset into two time periods: 1973-1995 and 1996-2019. The structure of the graphical game is learnt using our method (See Appendix C for details). Once the structure is learnt, we computed the price of anarchy which is the ratio between the maximum welfare across all strategy profiles and the minimum welfare across all strategy profiles in the ϵ-PSNE set, where welfare was defined as the sum of payoffs of all players. The price of anarchy for the duration 1973-1995 and 1996-2019 were 1.1793 and 1.0685 respectively. We observe that the price of anarchy in period 1973-1995 is greater than that of period 1996-2019. From this, we infer that in the past, the total welfare across European countries could have been increased by deviating away from Nash equilibria, however now the gain in total welfare by deviating away from Nash equilibria is not too much. We speculate that this correlates with the formation of the European union in 1993.

Related Work. None of the prior literature has dealt with either continuous actions or non-parametric utilities. In discrete-action games, [19] proposed a maximum-likelihood approach to learn "linear influence games" - a class of parametric graphical games with binary actions and linear payoffs. However, their method runs in exponential time and the authors assumed a specific observation model for the strategy profiles. For the same specific observation model, [20] proposed a polynomial time algorithm, based on ℓ_1-regularized logistic regression, for learning linear influence games. Their strategy profiles (or joint actions) were drawn from a mixture of uniform distributions: one over the pure-strategy Nash equilibria (PSNE) set, and the other over its complement. [21] obtained necessary and sufficient conditions for learning linear influence games under arbitrary observation model. [22] use a discriminative, max-margin based approach, to learn tree structured polymatrix games[1]. Their method runs in exponential time and the authors show that learning polymatrix games is NP-hard under this max-margin setting, even when the class of graphs is restricted to trees. Finally, [23] proposed a polynomial time algorithm for learning sparse polymatrix games in the discrete-action setting.

Contribution. Our goal is to come up with a provably correct method which has a polynomial time and sample complexity. We propose a novel yet simple method to learn graphical games with non-parametric payoffs. Our proposed mathematical model, in its final form, resembles a constrained lasso problem. But before reaching this simple formulation, it carefully handles non-parametric payoffs by modeling them using an infinite weighted sum of orthonormal basis function and then truncating it for estimation. We use ℓ_1-norm regularizer to encourage sparsity of the coefficients of the Fourier transform of the recovered payoff functions. We go beyond solving the optimization problem and fully characterize a payoff function which is at most $\frac{\epsilon}{2}$ away from the true payoff function. Using these payoff functions, we recover a game whose ϵ-PSNE is contained in the PSNE of the true game (and vice versa). If the true game satisfies some technical conditions, then the PSNE of our recovered game matches exactly with the PSNE of the true game. For n players and at most d in-neighbors per player, we show that $\Omega(d^3 \log(n))$ *samples* are sufficient to recover the exact structure of the true game.

[1]Polymatrix games are graphical games where each player's utility is a sum of unary (single player) and pairwise (two players) potential functions.

2. PRELIMINARIES

In this section, we collect notations and describe our problem.

2.1. Notation

Consider a directed graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ are set of vertices and edges respectively. We define $\mathcal{V} \triangleq \{1, \ldots, n\}$, where each vertex corresponds to one player. We denote the set of in-neighbors of a player i by S_i, i.e., $\mathrm{S}_i = \{j \mid (j, i) \in \mathcal{E}\}$. An in-neighbor of player i is a node that points to node i in the graph. All the other players are denoted by S_i^c, i.e., $\mathrm{S}_i^c = \{1, \ldots, n\} \backslash (\mathrm{S}_i \cup \{i\})$. Let $|\mathrm{S}_i| \leqslant d$ and $|\mathrm{S}_i^c| \geqslant n-1-d$. For each player $i \in \mathcal{V}$, there is a set of actions or *pure-strategies* $\mathcal{A}_i$. That is, player i can take action $x_i \in \mathcal{A}_i$. Each action x_i consists of making k decisions on a limited budget $\mathrm{b} \in \mathbb{R}$. We consider games with continuous actions. Mathematically, $x_i \in \mathcal{A}_i = \mathbb{R}^k$ and $\|x_i\|_2 \leqslant \mathrm{b}$. We use $x_{\text{-}i}$ to denote the collection of actions of all players but i. For each player i, there is also a local payoff function $u_i : \mathcal{A}_i \times (\times_{j \in \mathrm{S}_i} \mathcal{A}_j) \to \mathbb{R}$ mapping the joint action of player i and its in-neighbors S_i, to a real number. A joint action $\mathbf{x}^* \in \times_{i \in \mathcal{V}} \mathcal{A}_i$ is a *pure-strategy Nash equilibrium (PSNE)* of a graphical game iff, no player i has any incentive to unilaterally deviate from the prescribed action $x_i^* \in \mathcal{A}_i$, given the joint action of its in-neighbors $x_{\mathrm{S}_i}^* \in \times_{j \in \mathrm{S}_i} \mathcal{A}_j$ in the equilibrium. We denote a game by $\mathscr{G}$, and the set of all PSNE and ϵ-PSNE are denoted by $\mathrm{NE}(\mathscr{G})$ and $\mathrm{NE}_\epsilon(\mathscr{G})$ respectively, for a constant $\epsilon > 0$. Mathematically, $\mathrm{NE}(\mathscr{G}) \triangleq \{x^* \in \times_{i \in \mathcal{V}} \mathcal{A}_i \mid x_i^* \in \arg\max_{x_i \in \mathcal{A}_i} u_i(x_i, x_{\text{-}i}^*), \forall i \in \mathcal{V}\}$ and $\mathrm{NE}_\epsilon(\mathscr{G}) \triangleq \{x^* \in \times_{i \in \mathcal{V}} \mathcal{A}_i \mid u_i(x_i^*, x_{\text{-}i}^*) \geqslant -\epsilon + \max_{x_i \in \mathcal{A}_i} u_i(x_i, x_{\text{-}i}^*), \forall i \in \mathcal{V}\}$.

2.2. Model

Now we describe the basic setup of our problem. Our first challenge arises due to not having access to parametric payoffs. To tackle this, we decompose our non-parametric payoffs as a weighted sum of orthonormal basis functions. We consider a graphical game with the property that utility function of each player i is decomposable into sum of pairwise functions which only depend on the in-neighbors of i, i.e.,

$$u_i(x) = \sum_{j \in \mathrm{S}_i} u_{ij}(x_i, x_j) \;, \tag{1}$$

where S_i is the set of neighbors of player i. For two real valued functions $f(x)$ and $g(x)$ on the domain $x \in \mathcal{X}$, we define their inner product as $\langle f, g\rangle = \int_{x\in\mathcal{X}} f(x)g(x)dx$. Let $\psi_k(.,.), \forall k = \{1,\ldots,\infty\}$ be a set of uniformly bounded, orthonormal basis functions such that $u_{ij}(x_i,x_j) = \sum_{k=0}^{\infty} \beta^*_{ijk} \psi_k(x_i,x_j)$, where $\beta^*_{ijk} = \langle u_{ij}, \psi_k\rangle$. We assume that for all $i,j \in \{1,\ldots,n\}, i \neq j$, the weight magnitudes $|\beta^*_{ijk}|, \forall k \in \{r+1,\ldots,\infty\}$ form a convergent series for a sufficiently large r. For example, for large enough r, Fourier coefficients of a periodic and twice continuously differentiable function form a convergent series [24]. For all $i,j \in \{1,\ldots,n\}$ we assume that $\sup_{x_i\in\mathcal{A}_i, x_j\in\mathcal{A}_j} |\psi_k(x_i,x_j)| \leqslant \overline{\psi}$. Outside the context of graphical games, a setup involving orthonormal basis functions was used by [25]. However, their setup considers additive combination of univariate functions, while we use additive combination of bivariate functions. Our setup also fundamentally differs in the generative model and the use of ℓ_1 regularizer. Even contextually, we go beyond the proof techniques and provide provable insights about the learned game.

2.3. Sampling Mechanism

Treating the outcomes of the game as "samples" observed across multiple "plays" of the same game is a recurring theme in the literature for learning games [19, 21, 23]. All of these works assume access to Nash equilibria and a noise mechanism. Noise could be added to each player's strategy at a local level or by mixing the sample with a non-Nash equilibria set at a global level. Since none of the prior literature has dealt with either continuous actions or non-parametric utilities, we propose a novel sampling mechanism. We assume that we have access to a set of joint actions $\mathbf{x}$ along with noisy black-box access to their payoffs. For a joint action x, the blackbox outputs a noisy payoff $\widetilde{u}_i(x)$. The blackbox computes this noisy payoff by first computing noisy basis function values $\widetilde{\psi}_k(x_i,x_j)$ and then finally taking their weighted sum to get the final noisy payoff. Mathematically, $\widetilde{\psi}_k(x_i,x_j) = \psi_k(x_i,x_j) + \gamma_k(x_i,x_j)$. where $\gamma_k(x_i,x_j)$ are independent zero mean sub-Gaussian noise with variance proxy σ^2. The vectors $\boldsymbol{\psi}(x)$ and $\boldsymbol{\gamma}(x)$ collect $\psi_k(x_i,x_j)$ and $\gamma_k(x_i,x_j)$ indexed by $j \in \{1,\ldots,n\}, j \neq i$ and $k \in \{1,\ldots,r\}$. The class of sub-Gaussian random variables includes for instance Gaussian random variables, any bounded random variable (e.g. Bernoulli, multinomial, uniform), any random variable with strictly log-concave density, and any finite mixture of sub-Gaussian variables. Thus, using sub-Gaussian noise makes our setup quite general in nature. Correspondingly, the resulting noisy payoff function that we observe can be written as $\widetilde{u}_i(x) = \sum_{j\in\mathrm{S}_i} \widetilde{u}_{ij}(x_i,x_j) = \sum_{j\in\mathrm{S}_i}\sum_{k=0}^{\infty} \beta^*_{ijk}\widetilde{\psi}_k(x_i,x_j)$, where $\widetilde{u}_{ij}(x_i,x_j) = \sum_{k=0}^{\infty} \beta^*_{ijk}\widetilde{\psi}_k(x_i,x_j)$.

Estimation. Let $\overline{u}_{ij}(x_i,x_j)$ be our estimation of $u_{ij}(x_i,x_j)$. Ideally, we would like to estimate an infinitely long vector with entries indexed by (j,k) for $j \in \{1,\ldots,n\}, j \neq i$ and $k \in \{1,\ldots,\infty\}$ for every player i. However, this is impractical with finite computing resources. Rather, we estimate an $(n-1) \times r$ dimensional vector $\boldsymbol{\beta}$ with entries indexed by (j,k) for $j \in \{1,\ldots,n\}, j \neq i$ and $k \in \{1,\ldots,r\}$ for every player i. Using these finite number of coefficients, we estimate pairwise utility in the following manner, $\overline{u}_{ij}(x_i,x_j) = \sum_{k=0}^{r} \beta_{ijk}\psi_k(x_i,x_j)$ and thus, the resulting estimation of the utility function is $\overline{u}_i(x) = \sum_{j\in\mathrm{S}_i}\sum_{k=0}^{r} \beta_{ijk}\psi_k(x_i,x_j)$.

3. THEORETICAL RESULTS

In this section, we setup our estimation problem. We also mention some technical assumptions for our setup. Let D be a collection of N samples. We estimate $\boldsymbol{\beta}^*$ by solving the following optimization problem for some $\lambda > 0$:

$$\begin{aligned} &\min_{\beta} \frac{1}{N}\sum_{x\in D}(\widetilde{u}_i(x) - \overline{u}_i(x))^2 + \lambda\sum_{j\neq i}\sum_{k=0}^{r}|\beta_{ijk}| \\ &\text{such that } \sum_{j\neq i}\sum_{k=0}^{r}|\beta_{ijk}| \leqslant C\ , \end{aligned} \tag{2}$$

where $C > 0$ is a constant that acts as a budget on the coefficients β_{ijk} to ensure that they are not unbounded. We would like to prove that $\beta_{ijk} = 0, \forall j \notin \mathrm{S}_i$ and $\beta_{ijk} \neq 0, \forall j \in \mathrm{S}_i$. This gives us a straight forward way of picking in-neighbors of player i by solving optimization problem (2).

3.1. Assumptions

In this subsection, we will discuss the key technical assumptions which are required for our theoretical results. For notational clarity, we define the following two quantities: $\mathbf{H} \triangleq \frac{1}{N}\sum_{x\in D}\boldsymbol{\psi}(x)\boldsymbol{\psi}(x)^\intercal$ and $\widehat{\mathbf{H}} \triangleq \frac{1}{N}\sum_{x\in D}\big(\boldsymbol{\psi}(x)\boldsymbol{\psi}(x)^\intercal + \boldsymbol{\psi}(x)\boldsymbol{\gamma}(x)^\intercal\big)$. It should be noted that $\mathbf{H} = \mathrm{E}(\widehat{\mathbf{H}})$. For our first assumption, we want $\mathbf{H}$ restricted to in-neighbors to be an invertible matrix. Formally,

Assumption 1. *We assume that* $\mathbf{H}$ *is positive definite matrix, i.e.,* $\Lambda_{\min}([\mathbf{H}]_{\mathrm{S}_i\mathrm{S}_i}) = C_{\min} > 0$ *where* $\Lambda_{\min}$ *denotes the minimum eigenvalue.*

This assumption ensures that our optimization problem (2) has a unique solution and all our theoretical guarantees hold for this particular solution. As for our second assumption, we require that non-in-neighbors of a player do not affect the player's action too much. We achieve this by proposing a mutual incoherence assumption.

Assumption 2. *We assume that* $|||[\mathbf{H}]_{\mathrm{S}_i^c\mathrm{S}_i}[\mathbf{H}]^{-1}_{\mathrm{S}_i\mathrm{S}_i}|||_\infty \leqslant 1-\alpha$ *for some* $0 < \alpha < 1$.

While mutual incoherence is new to graphical games, it has been a standard assumption in various estimation problems such as compressed sensing [26], Markov random fields [27],

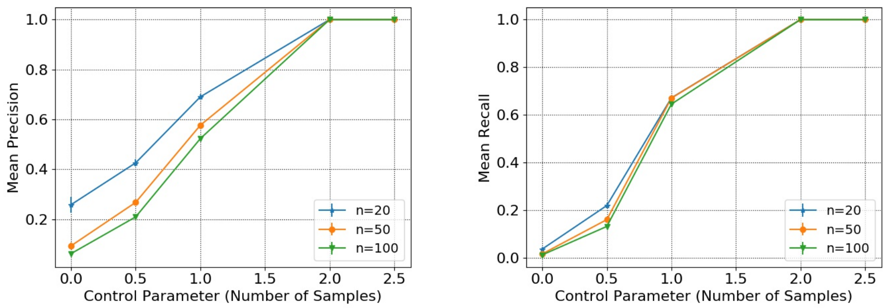

Fig. 1: Exact Structure Recovery in Non-parametric Games. Precision (Left) and Recall (Right) for recovered games across 30 runs against varying number of samples N. N is varied as $10^{C_p}\log(rn)$ where $r = 2$ and C_p is a control parameter.

non-parametric regression [25], diffusion networks [28], among others. It should be noted that Assumptions 1 and 2 also hold in the sample setting with sufficient number of samples (See Lemma 2 and 3 in Appendix).

Assumption 3. *We assume that* $\min_{j\in S_i}|\beta^*_{ijk}| > \Delta(\lambda, C_{\min}, d, r, \alpha)$ *where* $\Delta(\lambda, C_{\min}, d, r, \alpha) = \frac{2}{C_{\min}}(\frac{\alpha}{24} + \sqrt{rd})\lambda$.

The minimum weight assumption is a standard practice in the literature employing the primal-dual witness technique [26, 27]. It ensures that the coefficients β^*_{ijk} are not arbitrarily close to zero which can make inference very difficult for any method.

3.2. Main Theorem

Here, we state our main result. We use primal-dual witness framework [26] to prove it (Details in Appendix A).

Theorem 1. *Consider a continuous-action graphical game $\mathscr{G}$ such that Assumptions 1, 2 and 3 are satisfied for each player. Furthermore, assume that the payoff function for each player is decomposable according to Equation* (1)*. Let* $\lambda > \frac{48\sqrt{2\psi}\max(C,\sqrt{\delta})}{\alpha}\sqrt{\frac{d^2\log(2n)}{N}}$ *and* $N = \Omega(\frac{\bar{\psi}^2\sigma^2C^2\delta r^3 d^3}{\epsilon^2}\log(rn))$, *if we solve the optimization problem* (2) *then with probability at least* $1 - \exp(-c\lambda^2 N)$ *for some positive constant c,*

1. *We recover the correct non-neighbors for each player.*
2. *We recover the exact structure of* $\mathscr{G}$.
3. *We estimate a payoff function which is* $\frac{\epsilon(\lambda, C_{\min}, d, \bar{\psi}, \delta, r, \alpha)}{2}$ *close to the true payoff by estimating* $\boldsymbol{\beta}$ *for each player* i *by solving the optimization problem* (2) *where* $\epsilon(\lambda, C_{\min}, d, \bar{\psi}, \delta, r, \alpha) = \frac{2}{C_{\min}}(\frac{\alpha}{24} + \sqrt{rd})\lambda\sqrt{d}\bar{\psi} + \bar{\psi}\delta$.
4. *Let* $\widehat{\mathscr{G}}$ *be the estimated game which uses estimated payoff functions then* $\mathrm{NE}(\widehat{\mathscr{G}}) \subseteq \mathrm{NE}_\epsilon(\mathscr{G})$ *and* $\mathrm{NE}(\mathscr{G}) \subseteq \mathrm{NE}_\epsilon(\widehat{\mathscr{G}})$.
5. *If* $\forall i \in \mathcal{V}, \forall (x_i, x_{-i}), (x'_i, x_{-i}) \in \mathcal{A}$ *such that* $(x_i, x_{-i}) \in \mathrm{NE}(\mathscr{G})$ *and* $(x'_i, x_{-i}) \notin \mathrm{NE}(\mathscr{G})$ *implies that* $u^*_i(x_i, x_{-i}) > u^*_i(x'_i, x_{-i}) + \epsilon$ *then* $\mathrm{NE}(\mathscr{G}) \subseteq \mathrm{NE}(\widehat{\mathscr{G}})$*. Furthermore, if* $\mathrm{NE}_\epsilon(\mathscr{G}) = \mathrm{NE}(\mathscr{G})$ *then* $\mathrm{NE}(\mathscr{G}) = \mathrm{NE}(\widehat{\mathscr{G}})$.

where $C_{\min} = \Lambda([\mathbf{H}]_{S_i S_i})$ *and* δ *is an arbitrary parameter which depends on* r.

In simple words, we show that by choosing an appropriate regularizer and with access to sufficient number of samples we can not only learn the structure of the graphical game but also estimate a game which is close to the true game.

4. EXPERIMENTAL RESULTS

In this section, we validate our theorem by running computational experiments on synthetic data. Our goal is to recover the correct in-neighbors of all the players, by using noisy utilities and basis functions. The performance of our method is measured using precision and recall. We defer the details of experiments to Appendix B. We provide results of our experiments in Figure 1. We can see that both precision and recall go to 1 as we increase the number of samples, obtaining perfect recovery with enough samples. Notice that the different curves for different number of players ($n = 20, 50$ and 100) line up with one another quite well. This matches with our theoretical results and shows that with a constant number of in-neighbors $\Omega(\log(rn))$ samples are sufficient to recover the exact structure of the graphical games.

5. REFERENCES

[1] J. Nash, "Non-cooperative games," *Annals of Mathematics*, vol. 54, no. 2, pp. 286–295, 1951.

[2] M. Kearns, M. Littman, and S. Singh, "Graphical models for game theory," *Uncertainty in Artificial Intelligence*, pp. 253–260, 2001.

[3] C. Daskalakis, A. Fabrikant, and C. Papadimitriou, "The game world is flat: The complexity of Nash equilibria in succinct games," *International Colloquium on Automata,*

Languages, and Programming, vol. 4051, pp. 513–524, 2006.

[4] C. Daskalakis, P. Goldberg, and C. Papadimitriou, “The complexity of computing a Nash equilibrium,” *Communications of the ACM*, vol. 52, no. 2, pp. 89–97, 2009.

[5] C. Daskalakis, G. Schoenebeckt, G. Valiant, and P. Valiant, “On the complexity of Nash equilibria of action-graph games,” *ACM-SIAM Symposium on Discrete Algorithms*, pp. 710–719, 2009.

[6] R. Aumann, “Subjectivity and correlation in randomized strategies,” *Journal of Mathematical Economics*, vol. 1, pp. 67–96, 1974.

[7] B. Blum, C.R. Shelton, and D. Koller, “A continuation method for Nash equilibria in structured games,” *Journal of Artificial Intelligence Research*, vol. 25, pp. 457–502, 2006.

[8] S. Kakade, M. Kearns, J. Langford, and L. Ortiz, “Correlated equilibria in graphical games,” *ACM Conference on Economics and Computation*, pp. 42–47, 2003.

[9] L. Ortiz and M. Kearns, “Nash propagation for loopy graphical games,” *Neural Information Processing Systems*, vol. 15, pp. 817–824, 2002.

[10] C. Papadimitriou and T. Roughgarden, “Computing correlated equilibria in multi-player games,” *Journal of the ACM*, vol. 55, no. 3, pp. 1–29, 2008.

[11] D. Vickrey and D. Koller, “Multi-agent algorithms for solving graphical games,” *Association for the Advancement of Artificial Intelligence Conference*, pp. 345–351, 2002.

[12] Francisco Facchinei and Christian Kanzow, “Generalized nash equilibrium problems,” *4or*, vol. 5, no. 3, pp. 173–210, 2007.

[13] Gesualdo Scutari, Daniel P Palomar, Francisco Facchinei, and Jong-shi Pang, “Convex optimization, game theory, and variational inequality theory,” *IEEE Signal Processing Magazine*, vol. 27, no. 3, pp. 35–49, 2010.

[14] Steven Perkins and David S Leslie, “Asynchronous stochastic approximation with differential inclusions,” *Stochastic Systems*, vol. 2, no. 2, pp. 409–446, 2013.

[15] Steven Perkins, Panayotis Mertikopoulos, and David S Leslie, “Mixed-strategy learning with continuous action sets,” *IEEE Transactions on Automatic Control*, vol. 62, no. 1, pp. 379–384, 2015.

[16] Panayotis Mertikopoulos and Zhengyuan Zhou, “Learning in games with continuous action sets and unknown payoff functions,” *Mathematical Programming*, vol. 173, no. 1-2, pp. 465–507, 2019.

[17] O. Ben-Zwi and A. Ronen, “Local and global price of anarchy of graphical games,” *Theoretical Computer Science*, vol. 412, pp. 1196–1207, 2011.

[18] M. Irfan and L. Ortiz, “On influence, stable behavior, and the most influential individuals in networks: A game-theoretic approach,” *Artificial Intelligence*, vol. 215, pp. 79–119, 2014.

[19] J. Honorio and L. Ortiz, “Learning the structure and parameters of large-population graphical games from behavioral data,” *Journal of Machine Learning Research*, vol. 16, no. Jun, pp. 1157–1210, 2015.

[20] A. Ghoshal and J. Honorio, “From behavior to sparse graphical games: Efficient recovery of equilibria,” *IEEE Allerton Conference on Communication, Control, and Computing*, vol. 54, 2016.

[21] A. Ghoshal and J. Honorio, “Learning graphical games from behavioral data: Sufficient and necessary conditions,” *International Conference on Artificial Intelligence and Statistics*, vol. 54, pp. 1532–1540, 2017.

[22] V. Garg and T. Jaakkola, “Learning tree structured potential games,” *Neural Information Processing Systems*, vol. 29, pp. 1552–1560, 2016.

[23] A. Ghoshal and J. Honorio, “Learning sparse polymatrix games in polynomial time and sample complexity,” *International Conference on Artificial Intelligence and Statistics*, pp. 1486–1494, 2018.

[24] Bryan Rust, “Convergence of Fourier Series,” 2013.

[25] Pradeep Ravikumar, Han Liu, John Lafferty, and Larry Wasserman, “Spam: Sparse Additive Models,” in *Proceedings of the 20th International Conference on Neural Information Processing Systems*. Curran Associates Inc., 2007, pp. 1201–1208.

[26] Martin J Wainwright, “Sharp Thresholds for High-Dimensional and Noisy Sparsity Recovery Using L1-Constrained Quadratic Programming (Lasso),” *IEEE transactions on information theory*, vol. 55, no. 5, pp. 2183–2202, 2009.

[27] Pradeep Ravikumar, Martin J Wainwright, John D Lafferty, et al., “High-dimensional Ising Model Selection Using L1-Regularized Logistic Regression,” *The Annals of Statistics*, vol. 38, no. 3, pp. 1287–1319, 2010.

[28] Hadi Daneshmand, Manuel Gomez-Rodriguez, Le Song, and Bernhard Schoelkopf, “Estimating Diffusion Network Structures: Recovery Conditions, Sample Complexity & Soft-Thresholding Algorithm,” in *International Conference on Machine Learning*, 2014, pp. 793–801.