Computing using Samples: Theoretical Guarantees with the Direct Learning Approach

Doctoral Consortium

Vignesh Viswanathan
University of Massachusetts, Amherst
vviswanathan@umass.edu

ABSTRACT

Machine learning algorithms in the field of economics and game theory usually involve computing an intermediate valuation function from data samples and using this approximate function to compute desired solution concepts. This approach has several problems ranging from a high sample complexity to a lack of provable guarantees about the final solution. In order to avoid these problems, we explore a new method to learn solution concepts from data: instead of learning an intermediate valuation function, we learn the solution concept directly from the samples. This approach provides an alternative way to approximately learn solution concepts using fewer samples. In addition to this, from our study of using this approach to learn market equilibria, we find that, in a lot of settings, it is easier to prove efficiency and fairness guarantees about the learned solutions.

KEYWORDS

Theoretical Machine Learning; Computational Economics; Game Theory

ACM Reference Format:

Vignesh Viswanathan. 2021. Computing using Samples: Theoretical Guarantees with the Direct Learning Approach: Doctoral Consortium. In *Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), Online, May 3–7, 2021*, IFAAMAS, 3 pages.

1 INTRODUCTION

Consider the following problem: we have a dataset with several users' valuations for bundles of items that they purchased in a supermarket. We want to use this dataset to allocate goods to users or set prices for goods such that profits are maximized. The aforementioned problem is very similar in structure to several problems in Economics, Game Theory and Fair Division. The standard approach used to solve this problem is to first learn an approximate valuation function for each user using the available data and then compute the optimal allocation for each user with respect to the approximate valuations. We refer to this approach as *indirect learning*. This approach has several problems. First, the underlying function in most variants of this problem usually has a high sample complexity and would need an exponential number samples to approximate. Furthermore, standard function approximating techniques like deep learning, apart from being time consuming and requiring a large amount of computational resources, do not guarantee an approximate valuation function i.e a function that generalises to new data points with high probability.

As a result of this, it is not immediately clear whether an accurate solution computed with respect to approximate valuations is an approximate solution with respect to the accurate valuations. In addition to this, proving additional guarantees about the final solutions with respect to efficiency and fairness become close to impossible in most settings.

In this work, we explore an alternate approach to learning solution concepts which we refer to as *direct learning*. In this approach, instead of learning the valuation function, we learn an approximate solution concept directly from the data samples. The immediate advantage that can be seen from this approach is that the final solution is guaranteed to be an approximate solution. Furthermore, in several instances, direct learning algorithms are simpler both in terms of computational complexity as well as sample complexity than indirect learning algorithms.

For a better understanding of why this may be the case, consider the problem of finding the global maxima of a function: we have an unknown function $f : \mathbb{R}^n \mapsto \mathbb{R}$ and m samples $\{(x_1, f(x_1)), (x_2, f(x_2)), \dots, (x_m, f(x_m))\}$ where m is polynomial in n and we want to find an approximate global maxima of the function i.e. we want to find an approximate value of $\arg\max_{x \in \mathbb{R}^n} f(x)$. Since the function could be arbitrarily complex, we may not be able to approximate it using polynomial samples. Therefore, we may not be able to find an approximate solution using indirect learning. However, there exists a simple direct learning algorithm for this problem which works with any function f: output $\arg\max_{x_i \in \{x_1, x_2, \dots, x_m\}} f(x_i)$. This value can be shown to be a probably approximately correct value of the global maxima of the function [3].

While the above example should provide some intuition about how direct learning can be applied and why, in some cases, it has an advantage over indirect learning, the problem of finding a global maxima is fairly straightforward. In Section 2, we discuss the more complex problem of learning market equilibria using samples. Before that, we discuss previous works that use this approach.

1.1 Related Work

Direct Learning for complex solution concepts is a relatively unexplored area. [1, 2, 5] use direct learning to learn game theoretic solution concepts from data. Jha and Zick [3] propose a learning framework for game theoretic solution concepts and analyze the sample complexity of several solution concepts.

Our work builds on previous research and analyzes the specific case of Fisher markets with various valuation functions.

Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), U. Endriss, A. Nowé, F. Dignum, A. Lomuscio (eds.), May 3–7, 2021, Online.

2 LEARNING MARKET EQUILIBIRA FROM SAMPLES

This section discusses the results in Lev et al. [4]. In this paper, we consider a *Fisher market* with a set of *players* $N = \{1, 2, \ldots, n\}$ and a set of *indivisible goods* $G = \{g_1, g_2, \ldots, g_k\}$. Each player has a *budget* b_i and a *valuation function* $2^G \mapsto \mathbb{R}^+ \cup \{0\}$ which assigns a value $v_i(S)$ to each bundle of goods $S \subseteq G$.

We assume that the valuation function is unknown but we have m bundles of goods $\mathcal{S} = \{S_1, S_2, \ldots, S_m\}$ drawn from an unknown distribution $\mathcal{D}$ and we know each player's valuation for each bundle of goods.

An allocation is denoted by $(\mathcal{A}, \vec{p})$ where $\mathcal{A} = \{A_1, A_2, \ldots, A_n\}$ is the allocation vector and $\vec{p}$ is the price vector. Our goal is to compute an allocation such that it is an approximate equilibrium with high probability i.e. the probability that a bundle drawn from the unknown distribution can be afforded by a player who prefers this bundle to their allocated bundle is low. Mathematically, for any $\epsilon > 0$, the following condition is satisfied when m is polynomial in n and $\frac{1}{\epsilon}$

$$\Pr_{S \sim \mathcal{D}}[\exists i \in N, v_i(S) > v_i(A_i) \wedge \sum_{g \in S} p_g \leq b_i] \leq \epsilon$$

It is easy to see why this condition approximates the conditions imposed by the *Walrasian Equilibrium*. A Walrasian equilibrium satisfies the above condition for all bundles of goods $S \in 2^G$ whereas an approximate equilibrium only satisfies the condition with a high probability for a randomly sampled bundle of goods. We call an allocation which satisfies the above condition a Probably Approximately Correct Equilibrium or a *PAC Equilibrium*.

In addition to computing PAC equilibria in markets, we also prove efficiency guarantees for the allocations output by our algorithms. We define efficiency as the ratio of the total welfare of our allocation to the total welfare of the welfare maximizing equilibrium allocation i.e.

$$ER_v(\mathcal{A}) = \frac{\sum_{i=1}^{n} v_i(A_i)}{\sum_{i=1}^{n} v_i(A_i^*)}$$

where A^* is the welfare maximizing equilibrium allocation.

We study the computability and efficiency of PAC equilibria in markets with different classes of common valuation functions and discuss our findings below.

2.1 Unit Demand Valuations

In markets with unit demand valuations, the value of a bundle is equal to the value of the most valuable good in the bundle i.e. $v(S) = \max_{g \in S} v(\{g\})$. This class of valuation functions can be efficiently learned in polynomial samples and when all the players in the market have unit demand valuations, an equilibrium can be computed in polynomial time. However, we show that an indirectly learned equilibrium is not guaranteed to be a PAC equilibrium.

We then provide a direct learning algorithm to compute an equilibrium and show that our algorithm guarantees an efficiency bound of $\frac{1}{\min\{n,k\}}$. We also show that this efficiency bound is tight i.e. no algorithm can guarantee a better efficiency bound.

Furthermore, We show that, under product distributions and relatively sparse valuations, our algorithm outputs an allocation with efficiency 1 with exponentially high probability.

2.2 Additive Valuations

In markets with additive valuations, the value of a bundle is equal to the sum of the values of every good in the bundle i.e. $v(S) = \sum_{g \in S} v(\{g\})$.

Additive valuations can be efficiently learned but computing an equilibrium even when the valuations are known is an open problem. To work around this, we provide a direct learning algorithm which may not be market clearing but always outputs a PAC equilibrium. In addition to this we prove that our algorithm guarantees an efficiency of $\frac{1}{k}$ and show that this bound is tight.

2.3 Monotone Submodular Valuations

The class of monotone submodular valuations can be characterized by three equations. First, the empty set has a value of zero i.e. $v(\emptyset) = 0$. Second, the function is monotone i.e. if $S \subseteq T$, then $v(S) \leq v(T)$. Third, the valuation function satisfies the submodular inequality i.e. for any two sets S, T, we have $v(S) + v(T) \geq v(S \cup T) + v(S \cap T)$. This class of valuations is a superset of both the class of additive valuations as well as the class of unit demand valuations.

Monotone submodular valuations cannot be learned efficiently since they have an exponentially high sample complexity. Therefore, indirect learning cannot be used here. We provide a direct learning algorithm for markets with monotone submodular valuations and show that our algorithm guarantees an efficiency of $\frac{1}{k}$. Similar to previous valuation classes, we show that this bound is tight.

3 CONCLUSION AND FUTURE WORK

Our work shows the potential benefit of directly learning solutions, instead of learning utility functions and calculating solutions from them. We study Fisher markets with several valuation function families, and in all of them, use direct learning to construct algorithms which guarantee generalisation of the final solution to new samples as well as the highest possible efficiency.

We believe that this work is the tip of the iceberg in showing how direct learning can help in computing solution concepts in computer science and economics, directly from the data, without using the data to construct intermediate steps (such as learning utility functions). Even in the field of game theory, plenty of problems are still open – from expanding results to a larger family of functions (XOS, gross substitutes), to further type of results (e.g., other desirable states beyond equilibria). Outside of game theory, fields like recommender systems and fair division remain to be explored as well with potential results in both explainability and fairness.

REFERENCES

[1] M.F. Balcan, A.D. Procaccia, and Y. Zick. 2015. Learning cooperative games. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*.

[2] A. Igarashi, J. Sliwinski, and Y. Zick. 2019. Forming Probably Stable Communities with Limited Interactions. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI)*.

[3] T. Jha and Y. Zick. 2020. A Learning Framework for Distribution-Based Game-Theoretic Solution Concepts. In *Proceedings of the 21st ACM Conference on Economics and Computation (EC)*.
[4] O. Lev, N. Patel, V. Viswanathan, and Y. Zick. 2021. The Price is (Probably) Right: Learning Market Equilibria from Samples. In *Proceedings of the 20th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*.
[5] J. Sliwinski and Y. Zick. 2017. Learning Hedonic Games. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*.