Mach Learn (2007) 67:145–168
DOI 10.1007/s10994-007-0715-8

Learning payoff functions in infinite games

Yevgeniy Vorobeychik · Michael P. Wellman ·
Satinder Singh

Received: 1 September 2005 / Revised: 5 September 2006 / Accepted: 13
November 2006 / Published online: 30 January 2007

Abstract We consider a class of games with real-valued strategies and payoff information available only in the form of data from a given sample of strategy profiles. Solving such games with respect to the underlying strategy space requires generalizing from the data to a complete payoff-function representation. We address payoff-function learning as a standard regression problem, with provision for capturing known structure (e.g., symmetry) in the multiagent environment. To measure learning performance, we consider the relative utility of prescribed strategies, rather than the accuracy of payoff functions per se. We demonstrate our approach and evaluate its effectiveness on two examples: a two-player version of the first-price sealed-bid auction (with known analytical form), and a five-player market-based scheduling game (with no known solution). Additionally, we explore the efficacy of using relative utility of strategies as a target of supervised learning and as a learning model selector. Our experiments demonstrate its effectiveness in the former case, though not in the latter.

Keywords Game theory · Learning in games · Nash equilibrium approximation

1 Introduction

Game-theoretic analysis typically begins with a complete description of strategic interactions, that is, *the game*. We consider the prior question of determining what the game actually

Editors: Amy Greenwald and Michael Littman

Revised and extended version of a paper presented at IJCAI-05 (Vorobeychik, Wellman, & Singh, 2005).

Y. Vorobeychik (✉) · M. P. Wellman · S. Singh
University of Michigan, Computer Science & Engineering, 2260 Hayward St, Ann Arbor,
MI 48109-2121, USA
e-mail: yvorobey@umich.edu

M. P. Wellman
e-mail: wellman@umich.edu

S. Singh
e-mail: baveja@umich.edu

is, given a database of game experience rather than any direct specification. This is one possible target of *learning* applied to games (Shoham et al., forthcoming). When agents have few available actions and outcomes are deterministic, the game can be identified through systematic exploration. For instance, we can ask the agents to play each strategy profile in the entire joint strategy set and record the payoffs for each. If the joint action space is small enough, limited nondeterminism can be handled by sampling. Coordinating exploration of the joint set does pose difficult issues. Brafman and Tennenholtz, for example, address these carefully for the case of common-interest stochastic games (Brafman & Tennenholtz, 2003), as well as the general problem of maintaining an equilibrium among learning algorithms (Brafman & Tennenholtz, 2004).

Further difficulties are posed by intractably large (or infinite) strategy sets. We can make this problem tractable by reducing the number of profiles that agents are allowed to play, but this comes at the cost of transforming the game of interest into a different game entirely. Instead, we seek to identify the full game (or at least a less restrictive game) from limited data, entailing some generalization from observed instances. Approximating payoff functions using supervised learning (regression) methods allows us to deal with continuous agent strategy sets, providing a payoff for an arbitrary strategy profile. In so doing, we adopt functional forms consistent with prior knowledge about the game, and also admit biases toward forms facilitating subsequent game analysis (e.g., equilibrium calculation).

In this paper, we present the first study (to our knowledge) focusing on regression methods for inducing payoff functions for normal-form games. We investigate various model forms—low-degree polynomials, local regression, and support vector machines (SVMs)—applied to infinite games with strategy sets defined by real intervals. We explore two example games, both with incomplete information and real-valued actions. First is the standard first-price sealed bid auction, with two players and symmetric value distributions. The solution to this game is well-known (Krishna, 2002), and its availability in analytical form proves useful for benchmarking our learning approach. Our second example is a five-player market-based scheduling game (Reeves et al., 2005), where time slots are allocated by simultaneous ascending auctions (Cramton, 2005; Milgrom, 2000). This game has no known solution, though previous work has identified equilibria on discretized subsets of the strategy space.

2 Preliminaries

2.1 Notation

Our notation follows Mas-Colell, Whinston, and Green (1995). A generic normal-form game is formally expressed as $[I, \{\Delta(S_i)\}, \{u_i(s)\}]$, where I refers to the set of players and $m = |I|$ is the number of players. S_i is the set of strategies available to player $i \in I$, and the set $\Delta(S_i)$ is the simplex of mixed strategies over S_i. The utility function, $u_i(s) : S_1 \times \cdots \times S_m \to \mathcal{R}$ defines the payoff of player i when players jointly play $s = (s_1, \ldots, s_m)$, where each player's strategy s_j is selected from his strategy set, S_j. As is common, we assume von Neumann-Morgenstern utility, allowing an agent i's payoff for a particular mixed-strategy profile to be

$$u_i(\sigma) = \sum_{s \in S} [\sigma_1(s_1) \cdots \sigma_m(s_m)] u_i(s),$$

where $\sigma_j : S_j \to [0, 1]$ is a mixed strategy of player i, assigning a probability to each pure strategy $s_j \in S_j$ such that all probabilities over the agent's strategy set add to 1 (i.e., $\sigma_j \in \Delta(S_j)$).

It is often convenient to refer to the strategy (pure or mixed) of player i separately from that of the remaining players. To accommodate this, we use s_{-i} to denote the joint strategy of all players other than i.

2.2 Nash equilibrium

In this paper, we limit attention to one-shot normal-form games, in which players make decisions about their strategies simultaneously and accrue payoffs, upon which the game ends. This single-shot nature may seem to preclude multi-stage games, but in fact this representation is quite general, since strategies can be arbitrary functions of history of past play. Thus, any multi-stage game may be converted to a one-shot game by making the strategy sets to be sets of functions of all possible histories of play. However, if we do not explicitly represent the dynamic elements of a game, we do lose fidelity in terms of equilibrium concepts (for example, subgame perfection can no longer be represented).

Game payoff data may be obtained from observations of other agents playing the game, or from simulations of hypothetical runs of the game. In any of these cases, learning is relevant despite the fact that the game is to be played only once.

Faced with a one-shot game, an agent would ideally play its best strategy given those played by the other agents. A configuration where all agents play strategies that are best responses to the others constitutes a *Nash equilibrium.*

Definition 1. A strategy profile $s = (s_1, \ldots, s_m)$ constitutes a (pure-strategy) *Nash equilibrium* of game $[I, \{S_i\}, \{u_i(s)\}]$ if for every $i \in I$, $s_i' \in S_i$,

$$u_i(s_i, s_{-i}) \geq u_i(s_i', s_{-i}).$$

A similar definition applies when mixed strategies are allowed.

Definition 2. A strategy profile $\sigma = (\sigma_1, \ldots, \sigma_m)$ constitutes a *mixed-strategy Nash equilibrium* of game $[I, \{\Delta(S_i)\}, \{u_i(s)\}]$ if for every $i \in I$, $\sigma_i' \in \Delta(S_i)$,

$$u_i(\sigma_i, \sigma_{-i}) \geq u_i(\sigma_i', \sigma_{-i}).$$

In this study we devote particular attention to games that exhibit symmetry with respect to payoffs.

Definition 3. A game $[I, \{\Delta(S_i)\}, \{u_i(s)\}]$ is *symmetric* if $\forall i, j \in I$,

- $S_i = S_j$, and
- $u_i(s_i, s_{-i}) = u_j(s_j, s_{-j})$ whenever $s_i = s_j$ and $s_{-i} = s_{-j}$.

Symmetric games have relatively compact descriptions and may present associated computational advantages. Given a symmetric game, we may focus on the subclass of symmetric equilibria, which are arguably most natural (Kreps, 1990), and avoid the need to coordinate on roles.[1] In fairly general settings, symmetric games do possess symmetric equilibria (Nash, 1951; Cheng et al., 2004).

[1] Contention may arise when there are disparities among payoffs in asymmetric equilibrium. Even for symmetric equilibria, coordination issues may still be present with respect to equilibrium selection.

3 Payoff function approximation

3.1 Problem definition

We are given a set of data points (s, v), each describing an instance where agents played a pure strategy profile s and realized value $v = (v_1, \ldots, v_m)$. For deterministic games of complete information, v is simply $u(s)$. With incomplete information or stochastic outcomes, v is a random variable, more specifically an independent draw from a distribution function of s, with expected value $u(s)$.

The *payoff function approximation task* is to select a function $\hat{u}$ from a candidate set $\mathcal{U}$ minimizing some measure of deviation from the true payoff function u. Because the true function u is unknown, of course, we must base our selection on evidence provided by the given data points.

Our goal in approximating payoff functions is typically not predicting payoffs themselves, but rather in evaluating strategic behavior. Therefore, for assessing our results, we measure approximation quality not directly in terms of a distance between $\hat{u}$ and u, but rather in terms of the *strategies dictated by* $\hat{u}$ evaluated with respect to u. For this we appeal to the notion of approximate Nash equilibrium.

Definition 4. A strategy profile $\sigma = (\sigma_1, \ldots, \sigma_m)$ constitutes an ϵ-*Nash equilibrium* of game $[I, \{\Delta(S_i)\}, \{u_i(s)\}]$ if for every $i \in I$, $\sigma_i' \in \Delta(S_i)$,

$$u_i(\sigma_i, \sigma_{-i}) + \epsilon \geq u_i(\sigma_i', \sigma_{-i}).$$

We propose using ϵ in the above definition as a measure of approximation error of $\hat{u}$, and employ it in evaluating our learning methods. When u is known, we can compute ϵ in a straightforward manner. Let s_i^* denote i's *best-response correspondence*, defined by

$$s_i^*(\sigma_{-i}) = \{x : x \in \arg\max_{s_i} u_i(s_i, \sigma_{-i})\}.$$

For clarity of exposition, we take $s_i^*(\sigma_{-i})$ to be single-valued. Let σ be an arbitrary pure or mixed-strategy profile. We define

$$\epsilon(\sigma) = \max_{i \in I} \left[u_i(s_i^*(\sigma_{-i}), \sigma_{-i}) - u_i(\sigma) \right].$$

Now, let $\hat{\sigma}$ be a solution (for example, a Nash equilibrium) of game $[I, \{\Delta(S_i)\}, \{\hat{u}_i(s)\}]$. Then $\hat{\sigma}$ is an ϵ-Nash equilibrium of the true game $[I, \{\Delta(S_i)\}, \{u_i(s)\}]$, for $\epsilon = \epsilon(\hat{\sigma})$. Henceforth, whenever we use ϵ as an evaluation metric for $\hat{u}$, we will be referring to $\epsilon(\hat{\sigma})$, where $\hat{\sigma}$ is a Nash equilibrium of the game defined by $\hat{u}$. Since in general u will either be unknown or not amenable to this analysis, we also developed a method for estimating ϵ from data, described in some detail below.

Although an ϵ-Nash equilibrium is by definition nearly in equilibrium, it need not be close *in profile space* to any exact Nash equilibrium. We nevertheless consider the ϵ measure quite relevant as an evaluation metric. First, in many natural games (as, for example, in first-price sealed-bid auction we study below), ϵ-Nash profiles are indeed close to exact Nash profiles for small values of ϵ. Second, agents may well be content to play an ϵ-Nash equilibrium if ϵ is sufficiently small. In particular, they would if such a profile were a focal point (e.g., based on some external coordination signal), and the perceived cost of finding a better strategy exceeds ϵ. More generally, and especially when agents are inducing payoff functions from noisy data, any assessments of exact equilibrium is subject to error anyway, and so we can view ϵ as directly related to the *likelihood* that such a profile might be adopted in equilibrium.

For the remainder of this report, we focus on a special case of the general problem, where action sets are real-valued intervals, $S_i = [0, 1]$. Moreover, we restrict attention to symmetric games and introduce several forms of aggregation of other agents' actions as another knob for controlling model complexity.[2] The assumption of symmetry allows us to adopt the convention for the remainder of the paper that payoff $u(s_i, s_{-i})$ is to the agent playing s_i.

3.2 Polynomial regression

One class of models we consider are the *nth-degree separable polynomials*:

$$u(s_i, \phi(s_{-i})) = a_n s_i^n + \cdots + a_1 s_i + b_n \cdot (\phi(s_{-i}))^n + \cdots + b_1 \cdot \phi(s_{-i}) + d, \tag{1}$$

where $\phi(s_{-i})$ represents some aggregation of the strategies played by agents other than i.[3] For two-player games, ϕ is simply the identity function. We refer to polynomials of the form (1) as separable, since they lack terms combining s_i and s_{-i}. We also consider models with such terms, for example, the *non-separable quadratic*:

$$u(s_i, \phi(s_{-i})) = a_2 s_i^2 + a_1 s_i + b_2 \cdot (\phi(s_{-i}))^2 + b_1 \cdot \phi(s_{-i}) + c \cdot s_i \phi(s_{-i}) + d. \tag{2}$$

Note that (2) and (1) coincide in the case $n = 2$ and $c = \mathbf{0}$. In the experiments described below, we employ a simpler version of non-separable quadratic that takes $b_1 = b_2 = \mathbf{0}$.

One advantage of the quadratic form is that we can analytically solve for Nash equilibrium. Given a general non-separable quadratic (2), the necessary first-order condition for an interior solution is

$$s_i = -\frac{a_1 + c \cdot \phi(s_{-i})}{2a_2}.$$

This reduces to

$$s_i = -\frac{a_1}{2a_2}$$

in the separable case. For the non-separable case with *additive aggregation*, $\phi_{sum}(s_{-i}) = \sum_{j \neq i} s_j$, we can derive an explicit first-order condition for *symmetric* equilibrium:

$$s_i = -\frac{a_1}{2a_2 + (m-1)c}.$$

It has been observed that any game $[I, \{\Delta(S_i)\}, \{u_i(s)\}]$ in which $u_i(s)$ are continuous separable payoff functions on compact strategy sets has a pure-strategy Nash equilibrium (Balder, 1996). This result is quite intuitive, since the absence of interactions entails the existence of dominant strategies. However, a pure-strategy equilibrium need not exist in non-separable games, even if payoff functions are quadratic. In the experiments that follow, whenever our

[2] Although none of these restrictions are inherent in the approach, one must of course recognize the tradeoffs in complexity of the hypothesis space and generalization performance. Thus, we strive to build in symmetry to the hypothesis space whenever applicable.

[3] Note that $\phi(s_{-i})$ may be a vector, for example, when $\phi(s_{-i}) = s_{-i}$. In this case, $(\phi(s_{-i}))^n$ would be a vector in which every element of $\phi(s_{-i})$ is raised to the nth power.

polynomial approximation yields no pure-strategy Nash equilibrium, we randomly select a symmetric pure strategy profile from the joint strategy set.

Another difficulty arises when a polynomial of a degree higher than three has more than one Nash equilibrium. In such a case we select an equilibrium arbitrarily.

3.3 Local regression

In addition to polynomial models, we explore learning using two local regression methods: locally weighted average and locally weighted quadratic regression (Atkeson, Moore, & Schaal, 1997). Unlike model-based methods such as polynomial regression, local methods do not attempt to infer model coefficients from data. Instead, these methods weigh the training data points by distance from the query point and estimate the answer—in our case, the payoff at the strategy profile point—using some function of the weighted data set. We used a Gaussian weight function:

$$w = e^{-d^2},$$

where d is the distance of the training data point from the query point and w is the weight that is assigned to that training point.

In the case of locally weighted average, we simply take the weighted average of the payoffs of the training data points as our payoffs for a given strategy profile. Locally weighted quadratic regression, on the other hand, fits a quadratic regression to the weighted data set for each query point.

3.4 Support vector machine regression

The third category of learning methods we investigate is Support Vector Machines (SVMs). For details regarding this learning method, we refer an interested reader to Vapnik (1995). In our experiments, we used the *SVM light* package (Joachims, 1999), which is an open-source implementation of SVM classification and regression algorithms, and chose Gaussian radial basis function of the form $e^{-2||x-x'||^2}$ as the kernel.

3.5 Finding mixed-strategy equilibria

In the case of polynomial regression models, we were able to find either analytic or simple and robust numeric methods for computing pure Nash equilibria. With local regression and SVM learning we are not so fortunate, as we do not have access to a closed-form description of the function we have learned.

When a particular learned model is not amenable to a closed-form solution, we use it to approximate a Nash equilibrium of the underlying game as follows. First, we restrict the learned function to a finite strategy subset. Since this restriction produces a finite game, we can thereafter apply any generic finite game solver to find an approximate Nash equilibrium of the *learned* game. For this task, we employed replicator dynamics (Fudenberg & Levine, 1998), which searches for a symmetric mixed equilibrium using an iterative evolutionary algorithm. We treated the result after a fixed number of iterations as an approximate Nash equilibrium of the learned game.[4]

[4] Replicator dynamics does not necessarily converge, but when it does reach a locally asymptotically stable fixed point the result is a Nash equilibrium (Friedman, 1991). For cases where replicator dynamics fails to converge, we still treat the final result as an approximate Nash equilibrium of the game.

To summarize, we use the following procedure:

1. Randomly produce a training set of data points $\mathcal{D}$ in the form (s, v), where s is a pure strategy profile and v is a sample of corresponding payoffs to the players.
2. Run the learning algorithm on $\mathcal{D}$ to obtain a predictor of expected payoffs given a pure strategy profile s (for example, we would thereby obtain the coefficients of polynomial regression or the weights of the local learning methods and SVM)
3. Produce a grid of pure strategy profiles as a discrete approximation of the actual continuous strategy sets.
4. Use the predictor of choice (e.g., SVM) trained as above to obtain the approximate payoff matrix in the discrete strategy subset, thereby obtaining an approximation of the *learned* game.
5. Apply replicator dynamics to the discretized approximation of the learned game to approximate a Nash equilibrium.

3.6 Strategy aggregation

As noted above, we consider payoff functions on two-dimensional strategy profiles in the form $u(s_i, s_{-i}) = f(s_i, \phi(s_{-i}))$. As long as $\phi(s_{-i})$ is invariant under permutation of the strategies in s_{-i}, the payoff function is symmetric. Since the actual payoff functions for our example games are also known to be symmetric, we constrain that $\phi(s_{-i})$ preserve the symmetry of the underlying game.

In our experiments, we compared three variants of $\phi(s_{-i})$. First and most compact is the simple sum, $\phi_{sum}(s_{-i})$. Second is the ordered pair (ϕ_{sum}, ϕ_{ss}), where $\phi_{ss}(s_{-i}) = \sum_{j \neq i}(s_j)^2$. The third variant, $\phi_{identity}(s_{-i}) = s_{-i}$, simply takes the strategies in their direct, unaggregated form. To enforce the symmetry requirement in this last case, we sort the strategies in s_{-i}.

4 First-price sealed-bid auction

In the standard first-price sealed-bid (FPSB) auction game (Krishna, 2002), agents have private valuations for the good for sale, and simultaneously choose a bid price representing their offer to purchase the good. The bidder naming the highest price gets the good and pays the offered price. Other agents receive and pay nothing. In the classic setup first analyzed by Vickrey (1961), agents have identical valuation distributions, uniform on [0, 1], and these distributions are common knowledge. The unique (Bayesian) Nash equilibrium of this game is for agent i to bid $\frac{m-1}{m} x_i$, where x_i is i's valuation for the good.

Note that strategies in this game (and generally for games of incomplete information), $b_i : [0, 1] \to [0, 1]$, are functions of the agent's private information. We consider a restricted case, called "ray bidding" by Selten and Buchta (1994), where bid functions are constrained to the form

$$b_i(x_i) = s_i x_i, \quad s_i \in [0, 1].$$

This constraint transforms the action space to a real interval, corresponding to choice of strategy parameter s_i. Additionally, transformation simplifies representation of the game in normal-form. We can easily see that the restricted strategy space includes the known equilibrium of the full game, with $s_i = \frac{m-1}{m}$ for all i, which is also an equilibrium of the restricted game in which agents are constrained to strategies of the given form. While the fact

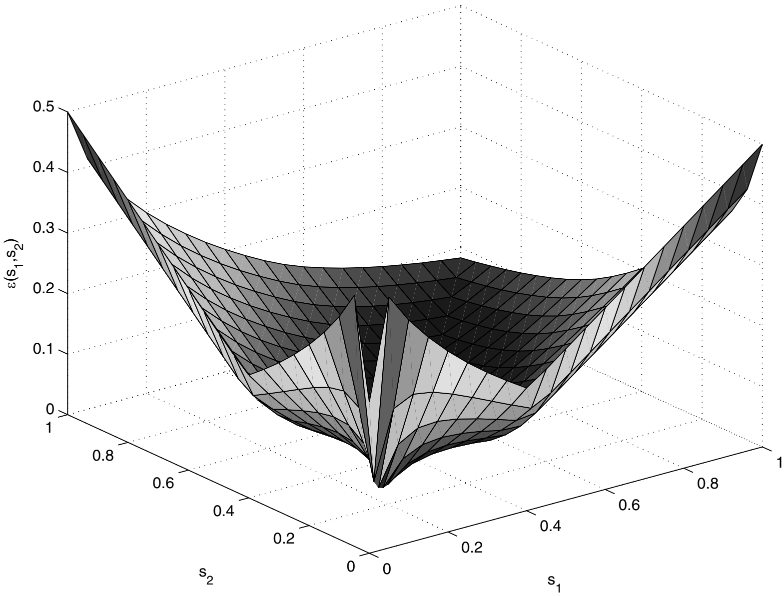


Fig. 1 $\epsilon(s)$ surface in two-player First-Price Sealed-Bid auction

that our restricted strategy space happens to include the actual equilibrium seems extremely fortunate, strategy spaces of this form are intuitively very natural: if we restrict bidders to submit bids in the space of valuations, any valuation that they may wish to bid is also some fraction of their actual value, and they will generally not wish to bid more than the object is worth to them. However, a criticism still applies that we generally do not a priori know if our restriction of the strategy space contains any equilibria of the actual game, and it is certainly not difficult to construct cases in which it is not. Thus, we will at best find an equilibrium of the restricted game, assuming generally that it is a reasonable equilibrium approximation of the actual game. While the question of finding appropriate restrictions of games is an important one, it is beyond the scope of this work.

We further focus on the special case $m = 2$, with corresponding equilibrium at $s_1 = s_2 = 1/2$. For the two-player FPSB, we can also derive a closed-form description of the actual expected payoff function (Reeves, 2005):[5]

$$u(s_1, s_2) = \begin{cases} 0.25 & \text{if } s_1 = s_2 = 0, \\ \dfrac{(s_1 - 1)[(s_2)^2 - 3(s_1)^2]}{6(s_1)^2} & \text{if } s_1 \geq s_2, \\ \dfrac{s_1(1 - s_1)}{3s_2} & \text{otherwise.} \end{cases} \tag{3}$$

The maximum benefit to deviation (ϵ) as a function of joint strategy $s \in [0, 1]^2$ is shown in Fig. 1.

The availability of known solutions for this example facilitates analysis of our learning approach. Our first set of results is summarized in Fig. 2. For each of our methods (classes

[5] Recall that $u(s_1, s_2) = u_1(s_1, s_2)$ and the payoff to the second player is symmetric.

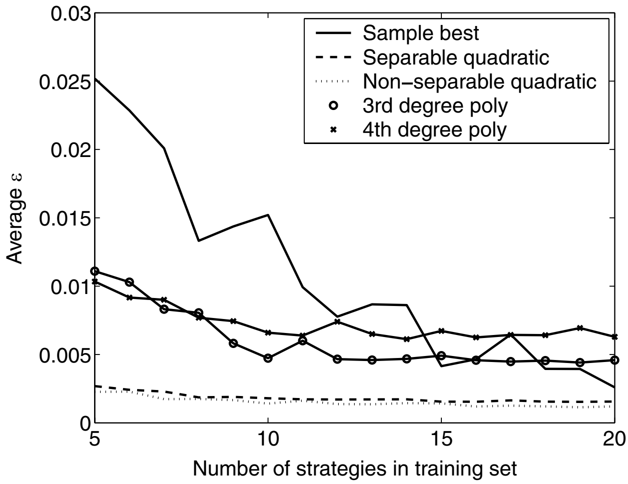


Fig. 2 Performance comparison of discrete (sample best) and polynomial function approximation models trained on noiseless data sets of varying size

of functional forms), we measured average ϵ for varying training set sizes. For instance, to evaluate the performance of separable quadratic approximation with training size N, we independently draw N strategies, $\{s^1, \ldots, s^N\}$, uniformly on [0, 1]. The corresponding training set comprises N^2 points: $((s^i, s^j), u(s^i, s^j))$, for $i, j \in \{1, \ldots, N\}$, with $u(s)$ as given by (3). We find the best separable quadratic fit $\hat{u}$ to these points, and find a Nash equilibrium corresponding to $\hat{u}$. We then calculate the least ϵ for which this strategy profile is an ϵ-Nash equilibrium with respect to the *actual* payoff function u. We repeat this process 200 times, averaging the results over strategy draws, to obtain each value plotted in Fig. 2.

As we can see, both second-degree polynomial forms we tried did quite well on this game. For $N < 20$, quadratic regression outperformed in our experiments the model labeled "sample best", in which the payoff function is approximated by the discrete training set directly. The derived equilibrium in this model is simply a Nash equilibrium over the discrete strategies in the training set. At first, the success of the quadratic model may be surprising, since the actual payoff function (3) is only piecewise differentiable and has a point of discontinuity. However, as we can see from Fig. 3, it appears quite smooth and well approximated by a quadratic polynomial. The higher-degree polynomials apparently overfit the data, as indicated by their inferior learning performance displayed in this game.

The results of this game provide an optimistic view of how well regression might be expected to perform compared to discretization. This game is quite easy for learning since the underlying payoff function is well captured by our lower-degree model.

The results thus far were based on a noiseless dataset. In another set of experiments we artificially added noise to the samples from (3) according to a zero-mean Normal distribution. In order to test the effect of noise on the performance of function approximation, we used a control variable, k, such that variance of noise is $1/k^2$. For each value of k and for each function approximation method, we randomly selected 5 strategies between 0 and 1. We then took noisy samples from (3) as described to produce a data set, $((s^i, s^j), v(s^i, s^j))$, for $i, j \in \{1, \ldots, 5\}$ (where $v = u(s^i, s^j) + \eta$ and $\eta \sim N(0, 1/k^2)$). After the model was fit to the data set, we found the corresponding equilibrium, and evaluated its ϵ with respect to the actual payoff function. The results, averaged over 200 random draws of 5 strategies, are shown

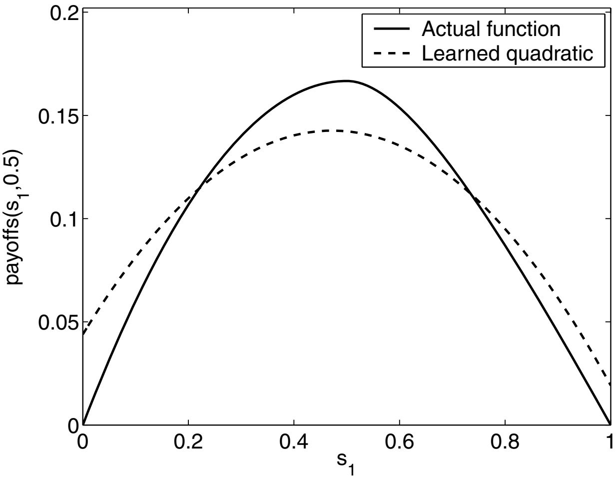
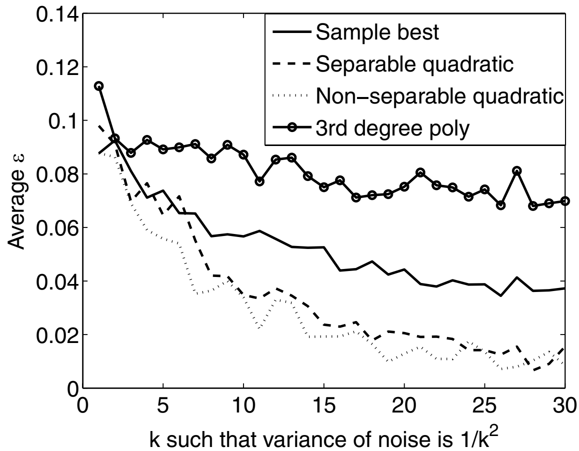


Fig. 3 Learned and actual payoff function when the other agent plays 0.5. The learned function is the separable quadratic, for a particular sample with $N = 5$

Fig. 4 Performance comparison of discrete (sample best) and polynomial function approximation models as the variance of noise in the data decreases



in Fig. 4. It is interesting to observe that when variance is very high, all methods performed extremely poorly, with non-separable quadratic and sample best (discrete approximation) methods approximately tied for best when variance was 1. However, as variance fell, we saw considerable improvement of the quadratic methods compared to discrete approximation, whereas the third-degree polynomial fit remained on average quite poor. Additionally, including the strategy interaction term produced better results in our experiments, as the non-separable quadratic outperformed other methods in most cases.

5 Market-based scheduling game

The second game we investigate presents a significantly more difficult learning challenge. It is a five-player symmetric game, with no analytic characterization, and no (theoretically) known solution. The game hinges on incomplete information, and training data is available only from a simulator that samples from the underlying distribution.

The game is based on a market-based scheduling scenario (Reeves et al., 2005), where agents bid in simultaneous ascending auctions for time-indexed resources necessary to perform their given jobs. Agents have private information about their job lengths, and values for completing their jobs by various deadlines. Note that the full space of strategies is quite complex—dependent on multi-dimensional private information about preferences as well as price histories for all the time slots. As in the FPSB example, we transform this policy space to a real interval by constraining strategies to a parametrized form. In particular, we start from a simple myopic policy—*straightforward bidding* (Milgrom, 2000), in which the agents choose the bundle of goods to bid on based on their current perceived prices. We generalize this strategy by introducing a scalar parameter, $s_i \in [0, 1]$, called "sunk awareness", which modifies the perceived-price calculation for slots the agent is currently winning. Specifically, the agent treats the effective price of such a slot as $s_i\beta$, where β is the current winning price. Playing strategy $s_i = 1$ is therefore the same as straightforward bidding, and $s_i = 0$ treats the goods as sunk costs. Intuitively, the appropriate value should be somewhat in between, since the agent is commited and the cost is sunk if the current price prevails, but may be let off the hook if another agent outbids it for this slot. Lower values of s_i increase the agent's tendency to stick with slots that it is currently winning.

Though sunk awareness represents a crude approximation to a complex tradeoff, including the parameter provides an agent with room for improvement over straightforward bidding. Most significantly for our current purposes, the optimal setting of s_i is generally dependent on other agents' behavior. This is precisely the relationship we aim to induce through learning. Toward this end, we collected data for all strategy profiles over the discrete set of values $s_i \in \{0, 0.05, \ldots, 1\}$. Accounting for symmetry, this represents 53,130 distinct strategy profiles. For evaluation purposes, we treat the sample averages for each discrete profile as the true expected payoffs on this grid.

We again acknowledge that no guarantees can be made about the actual game (bidding in sumultaneous ascending auctions) based solely on analysis of the restricted setting (straighforward bidding with variable sunk awareness). However, in this and many other settings, analysis of the entire strategic space is hopeless, and we generally must rely on restricted analysis to say anything about the actual game. Thus, we proceed to analyze the restricted game, presuming that it sheds light on the strategic interactions in the actual game.

The previous empirical study of this game by Reeves et al. (2005) estimated the payoff function over a discrete grid of profiles assembled from the strategies $\{0.8, 0.85, 0.9, 0.95, 1\}$, computing an approximate Nash equilibrium using replicator dynamics. We therefore used the training set based on the data for these strategies (300,000 samples per profile), regressed to the quadratic forms, and calculated empirical ϵ values with respect to the entire data set of 53,130 profiles by computing the maximum benefit from deviation within the data:

$$\epsilon_{emp} = \max_{i \in I} \max_{s_i \in S_i} [u_i(s_i, \hat{s}_{-i}) - u_i(\hat{s})] ,$$

where S_i is the strategy set of player i represented within the data set.[6] Providing for this restriction of player strategy sets, the definition of ϵ_{emp} above is the same as the definition of

[6] Observe that unlike the first-price sealed-bid auction game, the problem of computing the actual best response in the market-based scheduling game is a computationally intractable POMDP due to both incomplete and imperfect information (player types are private information, and the only observable feature of other agents' bids is the value of the highest for each good). Thus, for the purposes of this study, we treat the data set of 53,130 profiles as the true game.

Table 1 Values of ϵ for the symmetric pure-strategy equilibria of games defined by different payoff function approximation methods. The quadratic models were trained on profiles confined to strategies in $\{0.8, 0.85, 0.9, 0.95, 1\}$. The mixed-strategy equilibrium of the discrete model is presented as the probabilities with which the corresponding strategies are to be played

Method	Equilibrium s_i	ϵ
Separable quadratic	0.876	0.0027
Non-separable quadratic	0.876	0.0027
Discrete approximation	(0, 0.94, 0.06, 0, 0)	0.0238

the ϵ metric in Section 3.1. Since the game is symmetric, the maximization over players can be dropped, and all the agent strategy sets are identical.

If we compare the Nash equilibria of the learned functions presented in Table 1 to the inner product between the restricted strategy set and the corresponding mixed strategy produced using the discrete approximation (a kind of "expected" equilibrium strategy), the results are quite close. However, the learning methods produced much better approximations of Nash equilibria in terms of ϵ_{emp}.[7]

In a more comprehensive trial, we collected 2.2 million additional samples for each of 53,130 profiles, and ran our learning algorithms on 100 training sets, each uniformly randomly selected from the discrete grid $\{0, 0.05, \ldots, 1\}$. Each training set included all profiles generated from between five and ten of the twenty-one agent strategies on the grid. Since in this case an approximate equilibrium produced by a polynomial regression model does not typically appear in the complete data set, we developed a method for estimating ϵ for pure symmetric approximate equilibria in symmetric games based on a mixture of neighbor strategies that do appear in the test set. Let us designate a pure symmetric equilibrium strategy of the approximated game by $\hat{s}$. We first determine the closest neighbors to $\hat{s}$ in the symmetric strategy set S represented within the data. Let these neighbors be denoted by s' and s''. We define a mixed strategy α over support $\{s', s''\}$ as the probability of playing s', computed based on the relative distance of $\hat{s}$ from its neighbors:

$$\alpha = 1 - \frac{|\hat{s} - s'|}{|s' - s''|}.$$

Note that symmetry allows a more compact representation of a payoff function if agents other than i have a choice of only two strategies. Thus, we define $U(s_i, j)$ as the payoff to a (symmetric) player for playing strategy $s_i \in S$ when j other agents play strategy s'. If $m - 1$ agents each independently choose whether to play s' with probability α, then the probability that exactly j will choose s' is given by

$$\Pr(\alpha, j) = \binom{m-1}{j} \alpha^j (1-\alpha)^{m-1-j}.$$

We can thus approximate ϵ of the mixed strategy α by

$$\max_{s_i \in S} \sum_{j=0}^{m-1} \Pr(\alpha, j)[U(s_i, j) - \alpha U(s', j) - (1-\alpha) U(s'', j)].$$

[7] Since 0.876 is not a grid point, we determined ϵ_{emp} post hoc, by running further profile simulations with all agents playing 0.876, and where one agent deviates to any of the strategies in $\{0, 0.05, \ldots, 1\}$.

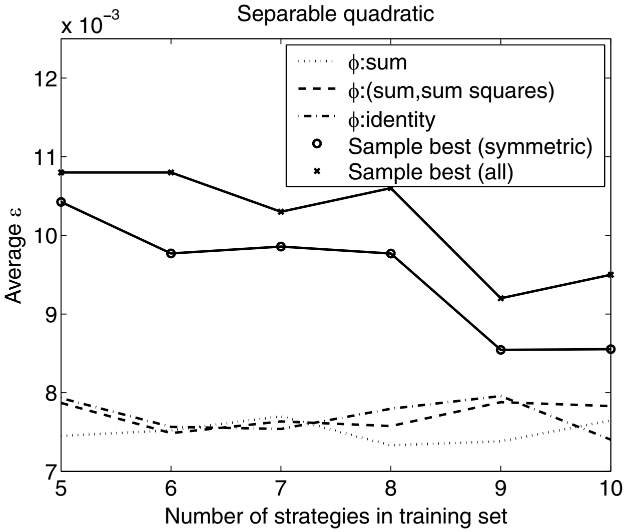


Fig. 5 Performance comparison of discrete and separable quadratic function approximation models with several forms of strategy aggregation

Using this method of estimating ϵ on the complete data set, we compared results from polynomial regression to the method which simply selects from the training set the pure-strategy profile with the smallest value of ϵ. We differentiate between the case where we only consider symmetric pure profiles (labeled "sample best (symmetric)") and all pure profiles (labeled "sample best (all)").[8]

From Fig. 5 we see that regression to a separable quadratic produced on average a considerably better approximate equilibria when the size of the training set was relatively small. Figure 6 shows that the non-separable quadratic performed similarly. The results appear relatively insensitive to the degree of aggregation applied to the representation of other agents' strategies.

The polynomial regression methods we employed yield pure-strategy Nash equilibria. We further evaluated four methods that generally produce mixed-strategy equilibria: two local regression learning methods, SVM with a Gaussian radial basis kernel, and direct estimation using the training data. As discussed above, we computed mixed-strategy equilibria by applying replicator dynamics to discrete approximations (using a fixed ten-strategy grid) of the learned payoff functions. In the case of direct estimation from training data, the data itself was used as input to the replicator dynamics algorithm. Since we ensure that the support of any mixed-strategy equilibrium produced by these methods is in the complete data set, we can compute ϵ of the equilibria directly.

As we can see in Fig. 7, locally weighted average method performed better in our experiments than the other three for most data sets that included between five and ten strategies. Additionally, locally weighted regression performed better than discrete approximation in

[8] Observe in Figs. 5 and 6 that when we restricted the search for a best pure strategy profile to symmetric profiles, we on average did better in terms of ϵ then when this restriction was not imposed.

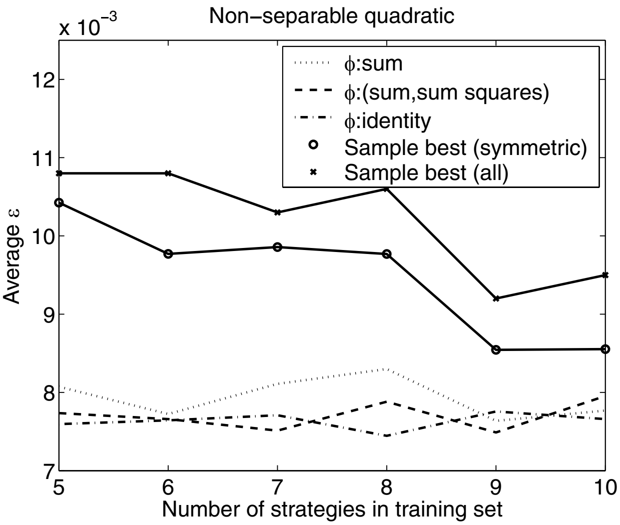
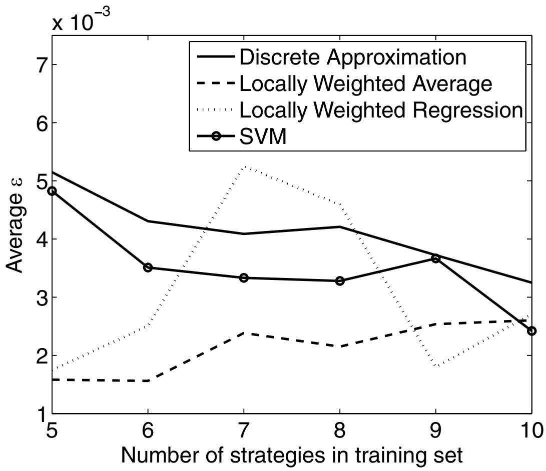


Fig. 6 Performance comparison of discrete and non-separable quadratic function approximation models with several forms of strategy aggregation

Fig. 7 Performance comparison of discrete approximation (with replicator dynamics used to find an equilibrium) and local and SVM regression methods with strategy aggregation of the form $\phi(s_{-i}) = (\phi_{sum}, \phi_{ss})$



our experiments on four of the six data set sizes we considered, and SVM consistently beat discrete approximation for all six data set sizes.[9]

It is somewhat surprising to see how irregular our results appear for the local regression methods. We cannot explain this irregularity, although of course there is no reason for us to expect otherwise: even though increasing the size of the training data set may improve

[9] Note that we do not compare these results to those for the polynomial regression methods. Given noise in the data set, mixed-strategy profiles with larger supports may exhibit lower ϵ simply due to the smoothing effect of the mixtures. Thus, when any profile with the lowest ϵ is desired, mixed-strategy profiles would likely be preferred. However, when pure strategy profiles are preferred, polynomial methods are more desirable as they produce pure strategy approximate equilibria.

the quality of fit, improvement in quality of equilibrium approximation does not necessarily follow.

In order to investigate the effect of noise on the function approximation methods as we had done in the FPSB setting, we ran another set of experiments in which we used the original data set of 300,000 samples per profiles as the training data set, and the data set of 2.5 million samples per profile as the evaluation data set. The remainder of the setup was as above.

The results in Figs. 8 and 9 show the performance of separable and non-separable quadratic models with different forms of strategy aggregation, comparing them to simply selecting the

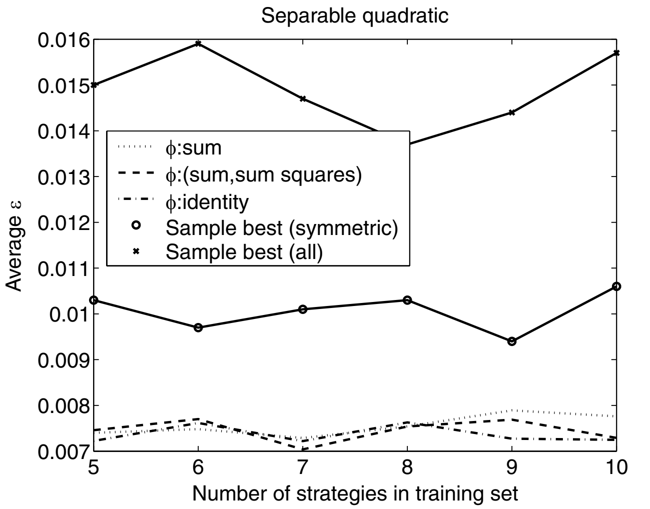


Fig. 8 Performance comparison of discrete and separable quadratic function approximation models with several forms of strategy aggregation when noise is present in training data

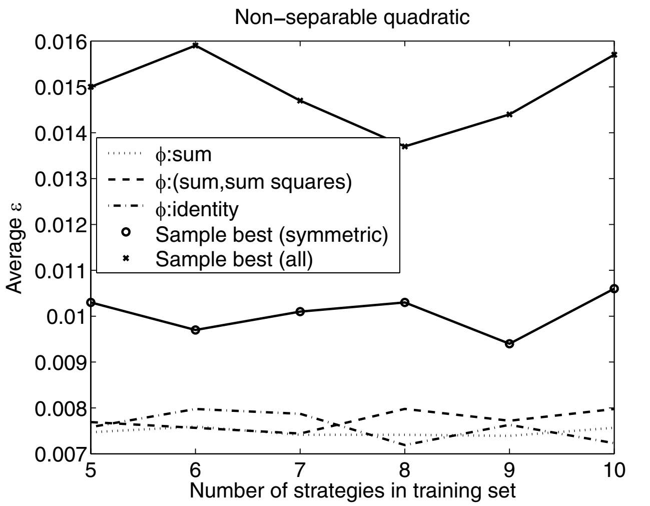


Fig. 9 Performance comparison of discrete and non-separable quadratic function approximation models with several forms of strategy aggregation when noise is present in training data

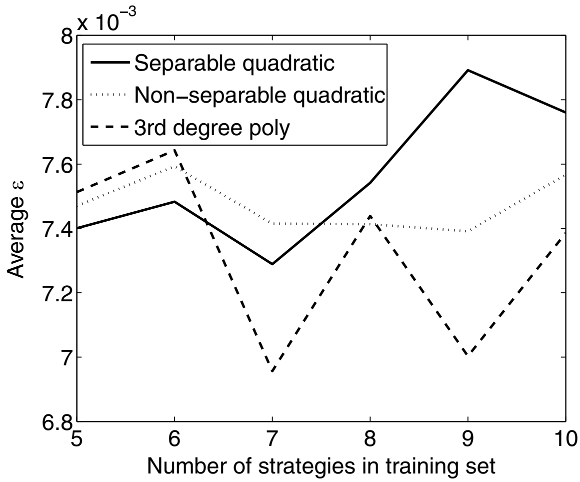


Fig. 10 Performance comparison of low-degree polynomial models trained on a noisy data set. All use strategy aggregation of the form ϕ_{sum}

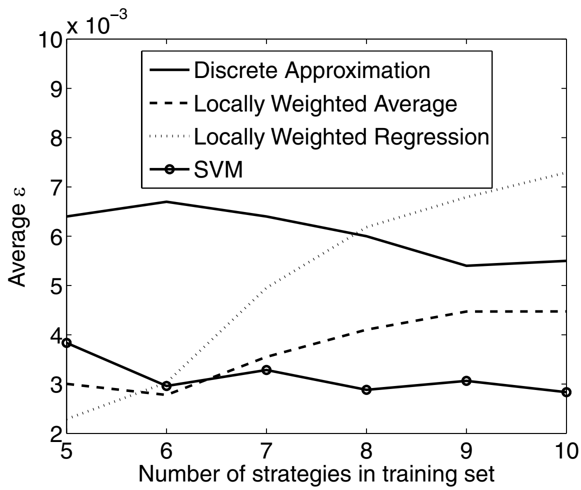


Fig. 11 Performance comparison of discrete approximation (with replicator dynamics used to find an equilibrium) and local and SVM regression methods with strategy aggregation of the form $\phi(s_{-i}) = (\phi_{sum}, \phi_{ss})$ trained on noisy data

best symmetric and asymmetric pure strategy profiles. The quadratic methods still significantly outperformed the methods based on discrete payoff function approximation in our experiments, and we again saw little difference between the three forms of strategy aggregation we tried.

Figure 10 suggests that the third order separable polynomial is relatively robust to noise, having outperformed the quadratic methods in the experiments when training data was more abundant.

The comparison of methods that yield mixed-strategy equilibrium approximations is shown in Fig. 11. As in the noiseless setting, learning methods tended to outperform discrete approximation. SVM produced the best approximations for almost all training data set sizes we had considered, while local learning methods were generally better than discrete approximation.

Finally, we compared the approximation quality of the SVM model with and without strategy aggregation in Fig. 12. We used the noisy training data set for these comparisons.

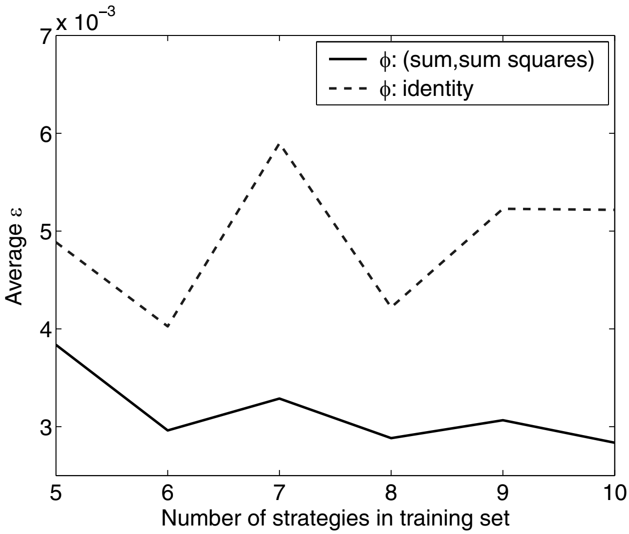


Fig. 12 Performance comparison of SVM model without strategy aggregation (identity) and with aggregation of the form $\phi(s_{-i}) = (\phi_{sum}, \phi_{ss})$

Unlike several comparisons between aggregation functions that we had seen already, in this case we did see a distinct advantage to using the particular form of strategy aggregation we chose here over using none at all.

6 Using ϵ in model selection

By using the ϵ metric in our study, we implicitly suggested that it may also be a useful guide for selecting a learning model to guide the final choice of an approximate equilibrium. A standard method for model selection that has been studied extensively is mean squared error computed on a test data set (see, for example, Hastie, Tibshirani, and Friedman (2001)). An alternative we would like to suggest here is to select the model with lowest ϵ (as defined above) estimated from the union of training and test data.

In the experiments here, we compared three model selection methods in the FPSB setting: mean-squared error on test data and two methods based on ϵ of the model fit on the union of training and test data. The mean-squared error method simply selects a model that has the lowest error on the test data set. The first ϵ-based method (we refer to it as "Best ϵ" in the experiments) finds the exact value of ϵ with respect to the restricted set of deviations represented in training and test data. Since the equilibrium of the learned model may not itself be contained in the training or the test data, we may need to take additional samples from the noisy payoff function.[10]

An alternative method estimates ϵ based only on the available training and test data (requiring no additional payoff samples) by taking a mixture of neighborhood points in the data set as a mixed-strategy candidate equilibrium profile in the place of the actual equilibrium

[10] We add noise artificially to the known FPSB payoff function.

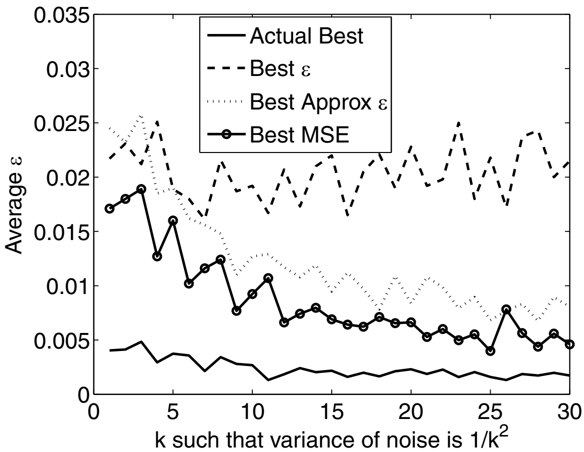


Fig. 13 Comparison of model selection criteria in FPSB as variance of the noise in the data decreases

of the learned game, as we discussed in some detail in Section 5. We refer to this method as "Best Approximate ϵ".

Figure 13 compares the three methods for model selection, varying the parameter, k, of the noise distribution, $N(0, 1/k^2)$, and keeping the size of the training set fixed at 25 profiles, representing all profiles for a random draw of 5 strategies. The model choices are restricted to separable and non-separable quadratics, as well as second- and third-degree separable polynomials. As a baseline for comparisons, we chose the model with the lowest actual ϵ in addition to applying the above selection criteria. This is referred to as "Actual Best" in the figure.

As another comparison between the same model selection methods, we fixed the distribution of noise to be $N(0, 1)$ and varied the number of strategies in the training data set between 5 and 10. The results of this comparison are shown in Fig. 14.

Both plots show that mean-squared error selection criterion consistently outperformed the others. Additionally, we can see from Fig. 13 that "Best Approximate ϵ" method tended to outperform the other ϵ-based method in the experiments. This provides some justification

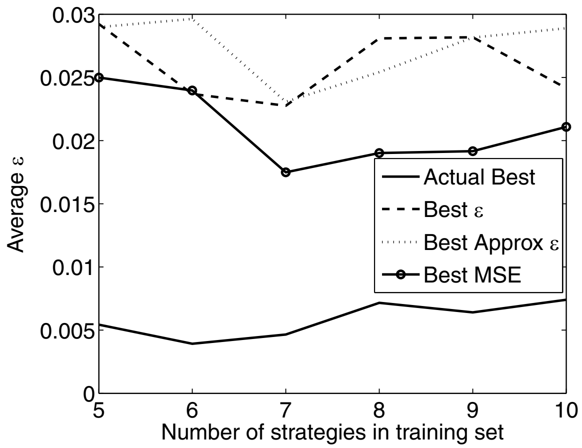


Fig. 14 Comparison of model selection criteria for different training data set sizes, fixing variance of noise at 1

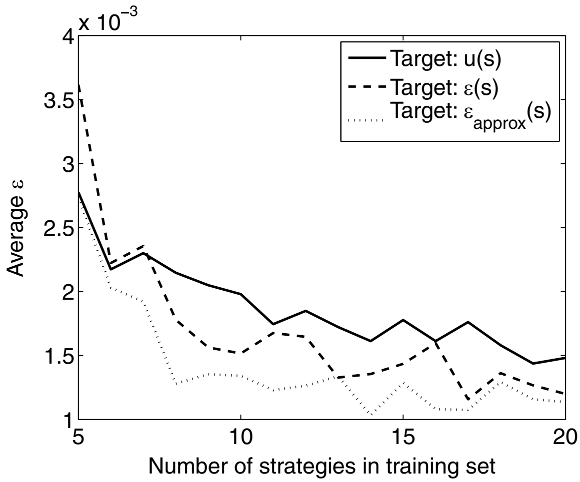


Fig. 15 Comparison of learning different targets on a noiseless data set of varying size

for our use of this as the estimate of actual ϵ in the experiments involving the Market-Based Scheduling game.

7 Using ϵ as the approximation target

Until now, the target of our function approximation endeavors had been the players' payoff functions. In this section, we would like to explore using $\epsilon(s)$ as defined in Section 3.1 as an alternative target, restricting the domain of this function to pure strategy profiles.

Since $\epsilon(s) \geq 0$ for all $s \in S$, and $\epsilon(s) = 0$ in equilibrium, any pure-strategy equilibrium minimizes $\epsilon(s)$. Now, suppose that we have an approximation, $\hat{\epsilon}(s)$ of $\epsilon(s)$. Then we can take a minimizer of $\hat{\epsilon}(s)$ to be an approximate Nash equilibrium.

There are several settings in which we may use $\epsilon(s)$ as a target. The most intuitive involves a data set in the form $(s, \epsilon(s))$ in place of the data set of payoff experience we had assumed thus far. This is a standard noiseless function approximation setting, and in our experiments we refer to it as "Target: $\epsilon(s)$".

In general, obtaining a data set in the form $(s, \epsilon(s))$ seems quite extraordinary. Typically, data representing payoff experience as we had supposed previously will be considerably easier to come by. In such a setting, we can still use an approximate $\epsilon(s)$ for each profile represented in the data set, as long as we restrict the strategy sets S_i to deviations available in data.[11] Making this empirical $\epsilon(s)$ as a target of function approximation is our second setting, which provides a more direct comparison with payoff function approximation techniques we had already explored.

Our first set of experiments, presented in Fig. 15, compares the two function approximation settings that employ $\epsilon(s)$ as the target to the setting of payoff function approximation already explored, using either a noiseless data set of FPSB payoff data points, $(s, u(s))$, or of data

[11] We can, in effect, drop the assumption here that all profiles in the restricted strategy space of players are available in data. Instead, we simply restrict our computation of ϵ to deviations available to the players within data. Note that this approach gives us a lower bound on empirical ϵ. For example, when no deviations are available for some profile s, we can take $\epsilon(s) = 0$. Nevertheless, we use data sets composed of all profiles in the restricted strategy spaces of players in our experiments.

points $(s, \epsilon(s))$, where appropriate. For all methods presented, we used the separable quadratic model described in Section 3.2. The remainder of the setup involved randomly selecting N strategies, for N ranging between 5 and 20, sampling all profiles over these strategies for either the data set of payoff experience or samples from $\epsilon(s)$, and learning the model for the appropriate target function. We repeated this process 200 times in each setting.

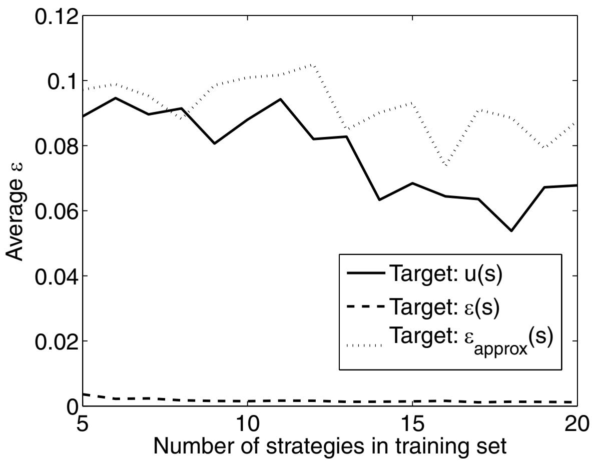
As we can see from Fig. 15, $\epsilon(s)$ approximation target generally performed somewhat better than payoff function approximation. It is noteworthy that even using the $\epsilon(s)$ approximated based on the payoff data as the target tended to produce better equilibrium approximations than those based on learned payoff functions.

Most surprising, perhaps, is the result that the approximate $\epsilon(s)$ provided a better target in our experiments than the actual $\epsilon(s)$. We propose several conjectures to explain this phenomenon. One possibility is that this is a purely idiosyncratic result due to our choice of model and our approximation setting, which manifests itself in a systematic bias of the model that is learned based on the actual examples of $\epsilon(s)$. The fact that this method still outperformed payoff function approximation may be evidence against this conjecture.

Another possibility is that the worst-case nature of $\epsilon(s)$ may tend to introduce irregularities into a data set that make the learning task more difficult in terms of our goals of subsequent equilibrium approximation, and restricting the set of possible deviations has a smoothing effect that actually facilitates learning. From limited exploration, we found some evidence supporting this conjecture. However, we can at this point say very little in general about this phenomenon.

In another set of experiments, we added noise, $\eta \sim N(0, 1)$, to the payoff data, affecting the learning of payoff functions and approximate $\epsilon(s)$. The remaining setup of these experiments was as above, and we present the performance of learning $\epsilon(s)$ based on noiseless samples from the actual function as a part of the plots for calibration. The results are presented in Figs. 16–18 for the settings of $k = 1, 4$, and 10 respectively. According to these plots approximating Nash equilibria based on payoff function approximation yielded results that showed generally better performance when variance was relatively high than using the same data set to derive an $\epsilon(s)$ as the learning target. On the other hand, when variance was small (Fig. 18) or zero (Fig. 15), approximate $\epsilon(s)$ became nearly as good or even better on average.

Fig. 16 Comparison of learning different targets when data has additive noise of the form $N(0, 1)$



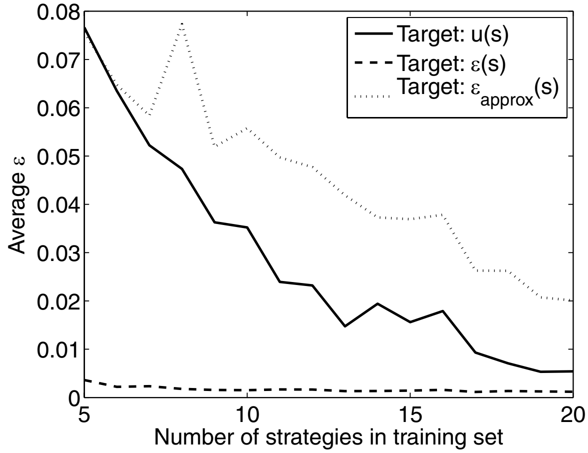


Fig. 17 Comparison of learning different targets when data has additive noise of the form $N(0, 0.0625)$

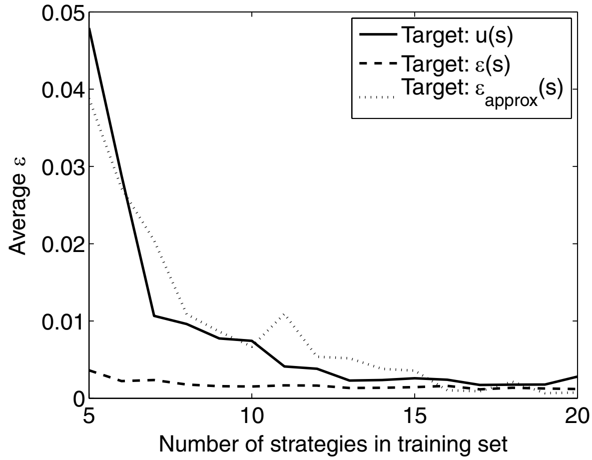


Fig. 18 Comparison of learning different targets when data has additive noise of the form $N(0, 0.01)$

Overall, the results in this section provide a first limited exploration into the relative performance of learning the equilibrium proximity function, $\epsilon(s)$, suggesting that at least in certain settings it could be reasonably effective.

8 Future work

8.1 Active learning

Thus far we had considered a static data set of examples that is available at the time of function approximation and assumed that no further data points can be generated thereafter. Alternatively, we can envision a simulator that produces data points in the desired form on-demand. In such a setting, function approximation is often enhanced when we can select data points to sample in order to minimize the variance of the approximate function (thereby also reducing the mean squared error, since the bias of the model space remains constant). A number of techniques had been developed that leverage the specific properties of the model class under consideration in order to selectively sample the function domain (Cohn, Atlas, & Ladner, 1994). These fall under the rubric of *active learning*.

The primary goal of active learning is to improve the expected quality of function fit in terms of mean squared error. Our goal, however, is somewhat different. We do not necessary care whether the quality of function fit is improved, as long as we can extract a better approximation of a Nash equilibrium from the approximate payoff function. While the two goals are intimately related (a poor function fit will generally not represent the strategic interactions very well), the techniques for achieving each may well be different.

On the other hand, we have considered an alternative setting of approximating the $\epsilon(s)$ function of the game, rather than the payoff functions of players, from data. In this setting, the goals of active learning and equilibrium approximation are considerably closer than in the setting of payoff function approximation. In the future, we intend to explore such connections between the two active learning goals and expect to engage in an experimental study of the relative utility of different active learning methods with respect to our end goals.

8.2 Learning Bayes-Nash equilibria

Let us suppose that we have an extended data set of experience, (s, v, t), where t is the vector of player types that play the strategy profile s and accrue the resulting payoffs, v. Given such a data set, we can extend the input to our function approximation models to include the players' type vector, t as the additional set of inputs. A particularly appealing model class is one studied by Reeves and Wellman (2004). For any model in this class Reeves and Wellman derived a method for computing a best response to a piecewise-linear strategy function (of players' types) exactly. This best-response finder can also be used in an iterative best-response dynamic to find Bayes-Nash equilibria.

If we know the actual distribution of players' types, we can use it directly in an equilibrium computation tool, whereas an unknown type distribution can be estimated from the data.

An implementation and experimental study of this learning framework is underway.

9 Conclusion

Whereas there has been much work in game theory attempting to solve particular games defined by given payoff functions, relatively little attention has been devoted to inducing such functions from data. This work addresses the question of payoff function induction by introducing regression learning techniques and applying them to representative games of interest. We evaluate the quality of a learned game model in terms of the potential gain, ϵ, to deviating from an equilibrium solution of this approximate model, measured in terms of the true underlying game.

Our results in both the FPSB and market-based scheduling games suggest that when data is sparse, regression methods can provide better approximations of the underlying game than direct estimation of a discrete model from the given data set. We observed this in cases where data points corresponded to actual or supposed actual payoff values, as well as cases where the given data comprises noisy samples of such payoffs. In the FPSB setting, we were able to artificially control for the amount of noise present in the data set, observing the dynamics of the payoff function approximation methods as the noise in the system decreased. A particularly interesting result in this case was that low-degree polynomial models exhibited an increasing advantage over discrete approximation as the variance of noise decreased.

In the market-based scheduling setting, we obtained two sets of training data with varying number of samples per profile. As a result, we were able to test our methods both in a noiseless setting, by using the data with the same number of samples per profile for both training and testing, and in a noisy setting, by using data with fewer samples per profile as the training set. We found that local regression and SVM learning methods generally outperformed discrete approximation in our experiments, both on noiseless and noisy training data. While locally weighted average shined when training data contained no noise, SVM model yielded better performance when noise was present.

We also introduced strategy aggregation as a way to control regression model complexity. The comparison between the various degrees of strategy aggregation provided mixed conclusions. In several settings, with and without noise, aggregating the strategies of other players appeared to provide no advantage in function approximation. In one setting, however, we did observe improved quality of equilibrium approximation when a particular form of strategy aggregation was used, as compared to no aggregation at all.

Having used the quality of the approximate equilibrium, ϵ, as a metric for the success of functional fit, we felt it natural to consider it explicitly as a model selection criterion. We evaluated it in comparison to model selection based on mean squared error of the payoffs on a test data set. Our results in the FPSB setting, however, were not favorable, as model selection based on the more direct measure of function accuracy provided better average performance.

Finally, we considered the possibility of learning $\epsilon(s)$ directly, rather than deriving it from a learned payoff-function model. In a series of experiments using FPSB as the game of choice, we observed mixed results. Whereas the noisy environment favored payoff function approximation, $\epsilon(s)$ performed relatively well when the data contained little or no noise.

In summary, we performed a series of experiments comparing several methods of payoff function regression, an alternative approach of learning $\epsilon(s)$, and exploring the possibility of using ϵ as a model selection criterion. Regression or other generalization methods offer the potential to extend game-theoretic analysis to strategy spaces (even infinite sets) beyond directly available experience. By selecting target functions that support tractable equilibrium calculations, we render such analysis analytically convenient. By adopting functional forms that capture known structure of the payoff function (e.g., symmetry or strategy aggregation), we facilitate learnability. By using $\epsilon(s)$ as a target of function approximation, we may in some cases be able to exploit structure that may not be as apparent in the more direct payoff function. This study provides some evidence of the efficacy of the different methods that we considered and the tradeoffs between them that emerge in different environments.

References

Atkeson, C. G., Moore, A. W., & Schaal, S. (1997). Locally weighted learning. *Artificial Intelligence Review*, *11*, 11–73.

Balder, E. J. (1996). Remarks on Nash equilibria for games with additively coupled payoffs. *Economic Theory*, *9*(1), 161–167.

Brafman, R. I., & Tennenholtz, M. (2003). Learning to coordinate efficiently: A model-based approach. *Journal of Artificial Intelligence Research*, *19*, 11–23.

Brafman, R. I., & Tennenholtz, M. (2004). Efficient learning equilibrium. *Artificial Intelligence*, *159*, 27–47.

Cheng, S.-F., Reeves, D. M., Vorobeychik, Y., & Wellman, M. P. (2004). Notes on equilibria in symmetric games. In *AAMAS-04 Workshop on Game-Theoretic and Decision-Theoretic Agents*. New York.

Cohn, D. A., Atlas, L., & Ladner, R. E. (1994). Improving generalization with active learning. *Machine Learning*, *15*(2), 201–221.

Cramton, P. (2005). Simultaneous ascending auctions. In P. Cramton, Y. Shoham, & R. Steinberg (Eds.), *Combinatorial auctions*. MIT Press.

Friedman, D. (1991). Evolutionary games in economics. *Econometrica*, *59*, 637–666.

Fudenberg, D., & Levine, D. (1998). *The theory of learning in games*. MIT Press.
Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The elements of statistical learning: Data mining, inference, and prediction*. Springer.
Joachims, T. (1999). Making large-scale SVM learning practical. In B. Scholkopf, C. Burges, & A. Smola (Eds.), *Advances in Kernel methods—support vector learning*. MIT Press.
Kreps, D. M. (1990). *Game theory and economic modelling*. Oxford University Press.
Krishna, V. (2002). *Auction theory*. Academic Press.
Mas-Colell, A., Whinston M. D., & Green J. R. (1995). *Microeconomic theory*. New York: Oxford University Press.
Milgrom, P. (2000). Putting auction theory to work: The simultaneous ascending auction. *Journal of Political Economy*, *108*, 245–272.
Nash, J. (1951). Non-cooperative games. *Annals of Mathematics*, *54*, 286–295.
Reeves, D. M. (2005). Generating Trading Agent Strategies. Ph.D. thesis, University of Michigan.
Reeves, D. M. & Wellman M. P. (2004). Computing best-response strategies in infinite games of incomplete information. In *Twentieth Conference on Uncertainty in Artificial Intelligence*, (pp. 470–478).
Reeves, D. M., Wellman, M. P., MacKie-Mason, J. K., & Osepayshvili, A. (2005). Exploring bidding strategies for market-based scheduling. *Decision Support Systems*, *39*, 67–85.
Selten, R., & Buchta, J. (1994). Experimental sealed bid first price auctions with directly observed bid functions. Technical Report Discussion Paper B-270, University of Bonn.
Shoham, Y., Powers, R., & Grenager, T. (forthcoming), If multi-agent learning is the answer, what is the question? *Artificial Intelligence*.
Vapnik, V. (1995). *The nature of statistical learning theory*. Springer-Verlag.
Vickrey, W. (1961). Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16, 8–37.
Vorobeychik, Y., Wellman, M. P., & Singh, S. (2005). Learning payoff functions in infinite games. In *Nineteenth International Joint Conference on Artificial Intelligence*, (pp. 977–982).