2011 50th IEEE Conference on Decision and Control and
European Control Conference (CDC-ECC)
Orlando, FL, USA, December 12-15, 2011

Learning User Preferences in Mechanism Design

Anil Kumar Chorppath and Tansu Alpcan

***Abstract*—In designing a mechanism for allocation of a divisible resource, the designer needs to know the player utility functions, which are often infinitely dimensional, in order to choose the appropriate pricing and allocation rules. This paper utilizes Gaussian process regression learning techniques to infer general player preferences by a designer in a mechanism design setting. In pricing mechanisms, the price taking players are charged with the appropriate value of Lagrange multiplier, in order to achieve efficiency. This value is obtained iteratively through learning. Likewise, the reserve price in auction mechanisms with price anticipating players, a parameter in allocation and pricing rules, is modified iteratively using online learning to move the system solution to near efficiency. A numerical example illustrates the approach and demonstrates the online learning algorithm.**

I. INTRODUCTION

In strategic (noncooperative) games, the players (users) interact with each other over time and each user responds to the actions of others. While operating under limited information, learning methods are useful tools for a player to learn and estimate its environment. *Learning theory* has been utilized in the context of game theory for many years [1]. Also, in recent years, there has been substantial work on algorithmic aspects of games, in the area of algorithmic game theory [2]. In *mechanism design*, in addition to the competing players with their own objectives, there is a designer with a system level or social goal. In order to achieve this goal, the designer, however, needs to know about the player preferences. In this paper, we use learning concepts for the *preference elicitation* of players in mechanism design.

We consider a scenario in which unlike most of the previous works in mechanism design, the designer explicitly models and learns the preferences or utilities of players through observations. In this setting, mechanisms can be interpreted as a method for learning players utility functions [3]. The concave or non-concave utility functions of players are private information to them, yet the designer requires them for optimal resource allocation and pricing. Here, the interactions in the iterative process in mechanisms are used to learn the marginal utilities of the users by the designer. The designer is assumed to be more powerful and can learn the *hidden information* observing the action of the players.

In this paper, the mechanism designer learns these utility functions using regression techniques based on the bids reported or actions taken by the users. Specifically, a *Gaussian process regression learning* [4] technique is used to estimate the marginal utilities of the players. The best response of the players to the prices and rules imposed by the designer constitutes the training data set. Once the marginal utilities are estimated with small number of training data points, the optimal point is searched in order to satisfy the optimality conditions. The nature of the optimal point may depend on the specific problem formulation. In the specific problem considered here, the optimality condition becomes full utilization of the resource given the marginal utilities of the users.

Two types of mechanisms are considered in this paper. In pricing mechanisms, the price taking players take best response actions to the the price charged by the designer. We use Gaussian process regression learning to approximate the utility function of players from the their actions, which are considered to be the input data points. Once the marginal utilities of players are learned, the space of the Lagrange multiplier of the total resource constraint in the designer problem is searched to obtain the optimal point.

In auctions the players bid as a response to the price and allocation designed by the designer. In a similar way as in pricing, the marginal utilities are learned through Gaussian process regression. Then, the reserve bid parameter in the price and allocation functions is updated until the optimality conditions are satisfied.

The main contributions of this paper is the application of regression learning methods to infer user utilities in mechanism design. Such learning schemes decrease the communication requirements considerably and allow usage of successive scalar bids or actions from the users for divisible resource allocation, even though the users have infinitely-dimensional utility functions. We apply these learning schemes to both pricing and auction mechanisms.

In the area of mechanism design for allocation of divisible resources, there have been many works to approximate the infinite dimensional utility function with finite quantities. In [5], the players are asked to bid on a scalar parameter of an allowable class of scalar parameterized utility functions which are named as surrogate utility functions. Since the payment and allocation is using Vicrey-Clarke-Groves(VCG) mechanism but based on scalar parameterized surrogate utility functions, they call it scalar parameterized VCG mechanism. The outcome results in at least one efficient Nash equilibrium (NE), when the marginal utility from actual utility function of user and marginal utility from using the declared parameter of surrogate utility function become equal. However, there can be other multiple Nash Equilibria, which are not efficient, due to the approximation by surrogate valuation functions. In [6], an iterative algorithm is proposed for VCG and scalar parameterized VCG mechanisms which reduces the amount

This work has been supported by Deutsche Telekom Laboratories.

Anil Kumar Chorppath is with Technical University of Berlin, Deutsche Telekom Laboratories, 10587 Berlin, Germany. Tansu Alpcan is with the Department of Electrical and Electronics Engineering, The University of Melbourne, Australia. He was with Technical University of Berlin, Deutsche Telekom Laboratories during this research. anil.chorppath@sec.t-labs.tu-berlin.de and tansualpcan@gmail.com

of overhead information by appropriate selection of the initial value of bids. The appropriate selection takes into account the previous bids and plays an important role in convergence to a NE. But they do not attempt any learning of the utility function.

An iterative auction *iBundle* [7] is proposed in a setting in which users take myopic best-response bidding as response to the bid of other users and rules set by the designer. The optimality of proposed iterative auction is proved with connection to primal-dual optimization theory.

In [3], the authors reduce mechanism design problems to standard algorithmic problems using techniques from sample complexity. They considered a prior-free setting for revenue maximization. The approach in [8] considers a learning phase followed by an accepting phase, and is careful to handle incentive issues for agents in both the phases. They study a limited-supply online auction problem, and construct value- and time-strategyproof auctions. The scenario when the users are strategic and they may manipulate the labeling for their individual benefit is considered in [9].

The next section presents the underlying game and the learning model. Section III analyses pricing mechanisms using regression learning. In Section IV auction mechanisms are considered. Numerical simulations and their results are shown in Section V. The paper concludes with remarks of Section VI.

II. Game and Learning Model

At the center of the game and mechanism design model is the *designer* $\mathcal{D}$ who influences N *players*, denoted by the set $\mathcal{A}$, and participating in a **strategic (non cooperative) game**. These players are autonomous and independent decision makers, who share and compete for limited resources under the given constraints of the environment. Concurrently, the designer tries to ensure that the outcome of the game satisfies the desirable properties of efficiency and truthfulness.

Let us define an N-player strategic game, $\mathcal{G}$, where each player $i \in \mathcal{A}$ has a respective **decision variable** x_i such that

$$x = [x_1, \ldots, x_N] \in \mathcal{X} \subset \mathbb{R}^N,$$

where $\mathcal{X}$ is the decision space of all players. Let

$$x_{-i} = [x_1, \ldots, x_{i-1}, x_{i+1}, \ldots x_N] \in \mathcal{X}_{-i} \subset \mathbb{R}^{N-1},$$

be the profile of decision variable of players other than i^{th} player and $\mathcal{X}_{-i}$ is the respective decision space. As a starting point, this paper assumes scalar decision variables and a compact and convex decision space. The decision variables may represent, depending on the specific problem formulation, player flow rate, power level, investment, or bidding in an auction. Due to the inherent coupling between the players, the decisions of players directly affect each other's performance as well as the aggregate allocation of limited resources.

The users have separable utility functions

$$U_i(x_i) : \mathcal{X} \to \mathbb{R}, \quad \forall i \in \mathcal{A},$$

which are chosen to be continuous and differentiable for analytical tractability. In this paper, the utility functions are assumed to be concave which is an assumption that captures marginal diminishing utility from the resource.

We consider two specific examples of concave, non-decreasing utility functions in order to demonstrate the results,

1) θ-Utility function, i.e.,

$$U_i(x_i; \alpha_i) = \begin{cases} \alpha_i \frac{x_i^{(1-\theta)}-1}{1-\theta}, & \text{if } \theta \geq 0 \text{ and } \theta \neq 1 \\ \alpha_i \log(x_i), & \text{if } \theta = 1 \end{cases}$$

2) Exponential:

$$U_i = 1 - e^{-\alpha_i x_i} \ \ \forall i \in \mathcal{A}.$$

We note however that the presented approach is applicable to any nonparametric utility function. The techniques are also independent whether the utility functions are concave or not and separable or non-separable. Therefore, the approach here case can be generalized to more general cases.

The designer $\mathcal{D}$ devises a **mechanism** M, which can be represented by the mapping $M : \mathcal{X} \to \mathbb{R}^N$, implemented by introducing incentives in the form of *rules and prices* to players. The latter can be formulated by adding it as a cost term such that the player i has the cost function

$$J_i(x) = c_i(x) - U_i(x). \tag{1}$$

Thus, the **player objective** is to solve the following individual optimization problem in the strategic game

$$\min_{x_i} J_i(x), \tag{2}$$

under the given constraints of the strategic game, and rules and prices imposed by the designer.

We consider two different types of mechanisms, pricing and auctions, which vary on the assumption on the nature of users and the rules imposed by the designer.

In the case of **pricing mechanism**, designer $\mathcal{D}$ introduces incentives in the form of *prices* P_i to the price taking users. The incentives can be formulated by adding it as a cost term such that the player i. With the pricing mechanism in place, each player i's cost is given by

$$J_i(x) = P_i x_i - U_i(x_i), \tag{3}$$

which is strictly convex in x_i.

In the case of **auctions**, the designer $\mathcal{D}$ imposes on a price anticipating user $i \in \mathcal{A}$ a user-specific

- resource allocation rule, $Q_i(x)$,
- resource pricing, $P_i(x)$,

where x denotes the vector of player bids. Thus, the cost function of player i can be written as

$$J_i(x) = P_i(x)\, Q_i(x) - U_i(Q_i(x)), \tag{4}$$

which is strictly convex with respect to Q_i under the assumptions made.

The **designer objective**, e.g. maximization of aggregate user utilities or social welfare, can be formulated using a smooth objective function V for the designer:

$$V(x, U_i(x), c_i(x)) : \mathcal{X} \to \mathbb{R},$$

where $c_i(x)$ and $U_i(x)$, $i = 1, \ldots, N$ are player-specific pricing terms and player utilities, respectively. Hence, the global optimization problem of the designer is simply $\max_x V(x, U_i(x), c_i(x))$, which it solves *indirectly* by setting rules and prices. The formal definitions of properties of mechanisms considered in this paper are given in the Appendix.

The users have utility functions private to them which are functions on their actions x. Therefore, the designer can learn these utility functions from the actions taken by the users. Consider any function $f(.)$ and a set of M data points $\mathcal{D} = \{x_1, \ldots, x_M\}$, and the corresponding vector of scalar values is $\{f(x_1), f(x_2)...., f(x_M)\}$. A regression learning algorithm uses the training data set to give a learned function $\hat{f}$ which minimizes the error from f and follows the real shape of f. Assume that the observations are distorted by a zero-mean Gaussian noise, n with variance $\sigma \sim \mathcal{N}(0, \sigma)$. Then, the resulting observations is a vector of Gaussian $y = f(x) + n \sim \mathcal{N}(f(x), \sigma)$.

A Gaussian Process (GP) is formally defined as a collection of random variables, any finite number of which have a joint Gaussian distribution. It is completely specified by its mean function $m(x)$ and covariance function $C(x, \tilde{x})$, where

$$m(x) = E[\hat{f}(x)]$$

and

$$C(x, \tilde{x}) = E[(\hat{f}(x) - m(x))(\hat{f}(\tilde{x}) - m(\tilde{x}))],\ \forall x, \tilde{x} \in \mathcal{D}.$$

Let us for simplicity choose $m(x) = 0$. Then, the GP is characterized entirely by its covariance function $C(x, \tilde{x})$. Since the noise in observation vector y is also Gaussian, the covariance function can be defined as the sum of a *kernel function* $W(x, \tilde{x})$ and the diagonal noise variance

$$C(x, \tilde{x}) = W(x, \tilde{x}) + \sigma I,\ \forall\, x, \tilde{x} \in \mathcal{D}, \tag{5}$$

where I is the identity matrix. While it is possible to choose here any (positive definite) kernel $W(\cdot, \cdot)$, one classical choice is

$$W(x, \tilde{x}) = \exp\left[-\frac{1}{2}\left\|x - \tilde{x}\right\|^2\right]. \tag{6}$$

Note that GP makes use of the well-known *kernel trick* here by representing an infinite dimensional continuous function using a (finite) set of continuous basis functions and associated vector of real parameters in accordance with the *representer theorem* [10].

The training set $(\mathcal{D}, y)$ is used to define the corresponding GP, $\mathcal{GP}(0, C(\mathcal{D}))$, through the $M \times M$ covariance function $C(\mathcal{D}) = W + \sigma I$, where the conditional Gaussian distribution of any point outside the training set, $\bar{y} \in \mathcal{X}, \bar{y} \notin \mathcal{D}$, given the training data $(\mathcal{D}, t)$ can be computed as follows. Define the vector

$$k(\bar{x}) = [W(x_1, \bar{x}), \ldots W(x_M, \bar{x})] \tag{7}$$

and scalar

$$\kappa = W(\bar{x}, \bar{x}) + \sigma. \tag{8}$$

Then, the conditional distribution $p(\bar{y}|y)$ that characterizes the $\mathcal{GP}(0, C)$ is a Gaussian $\mathcal{N}(\hat{f}, v)$ with mean $\hat{f}$ and variance v,

$$\hat{f}(\bar{x}) = k^T C^{-1} y \text{ and } v(\bar{x}) = \kappa - k^T C^{-1} k. \tag{9}$$

This is a key result that defines GP regression. The mean function $\hat{f}(x)$ of the GP provides a prediction of the objective function $f(x)$. Furthermore, the variance function $v(x)$ can be used to measure the uncertainty level of the predictions with the mean value $\hat{f}$.

Here the designer learns the marginal utility functions U_i' of each user using their best response bids or actions as data points.

III. Learning in Pricing Mechanisms

In this section, regression techniques are used to learn the user private marginal utilities by the designer for implementation of pricing mechanisms. The users are not considered to be price anticipating here because we consider a distributed network in which there is an information asymmetry between the users and the designer. The users do not know the action and utility function of other users or the nature of pricing function. Hence, they cannot anticipate the exact impact of their action on the pricing function and they just adopt a best response strategy by taking the price as a constant given by the designer.

We consider the case of a divisible resource of amount B is allocated among users having separable utility functions. The resource can be spectrum in wireless communication systems or bandwidth in the Internet. Due to the selfish nature of the individual users, without designer intervention there will be an inefficient distribution of the divisible resource (Price of Anarchy). The prices are designed to bring the Nash Equilibrium of the resulting game to an efficient point.

The user optimization problem will be to find the action level which minimizes his individual cost given in equation(3), i.e.,

$$\min_{x_i} \quad P_i x_i - U_i(x_i).$$

Consequently, the general condition for player best response obtained from first order derivative is

$$P_i - \frac{dU_i(x_i)}{dx_i} = 0,\ \forall\ i \in \mathcal{A}. \tag{10}$$

The best response will be,

$$x_i = (\frac{dU_i}{dx_i})^{-1}(P_i),\ \forall\ i \in \mathcal{A}. \tag{11}$$

The designer want to achieve the maximum social welfare, i.e the net utility of users is to be maximized. Therefore, the social objective is,

$$V = \max_x \sum_i U_i(x_i), \text{such that} \sum_i x_i \leq B.$$

The Lagrangian is given by

$$L = \sum_i U_i(x_i) + \lambda(\sum_i x_i - B).$$

where $\lambda > 0$ is the unique Lagrange multiplier.

The resulting Karush-Kuhn-Tucker (KKT) conditions will give,

$$U_i'(x_i) = \lambda, \ \forall i \in \mathcal{A}, \tag{12}$$

and

$$\lambda(\sum_i x_i - B) = 0, \ \forall i,$$

Since the individual user utility functions are concave and non-decreasing, the optimum point will ensure boundary solution. By comparing (12) and (10), we conclude that for aligning designer and user objectives, the designer needs to set λ as the price for every user for solving designer and user problems. Therefore, from the criterion of full resource usage, it follows that

$$\sum_i x_i^* = \sum_i \left(U_i'\right)^{-1}(\lambda^*) = B. \tag{13}$$

where x^* and λ^* are the optimal points.

Each user i sends a response to the sample prices $\{P_{i1}, \cdots, P_{iM}\}$ set by the social planner which contains the action vector $\{x_{i1}, \cdots, x_{iM}\}$. The corresponding scalar marginal utility values at those points are $U_i'(x_{i1}, \cdots, U_i'(x_{iM}) \ , \forall i$. Assume that the observations are distorted by a zero-mean Gaussian noise, n with variance $\sigma \sim \mathcal{N}(0, \sigma)$. Now let the Gaussian vector obtained in the case of user i is $\{y_{i1}, \cdots, y_{iM}\}$, where

$$y_{im} = U_i'(x_{im}) + n_i \ \forall i.$$

A Gaussian regression technique as described in Section II is used to estimate the marginal utility functions $\tilde{U_i'}$. After that, the λ values are obtained by an online learning algorithm. The optimal points λ^* and x^* are selected at which

$$\lambda^* = \tilde{U_i'} = \tilde{U_j'}, \forall i, j$$

and $\sum_i x_i^* = B$.

The algorithm which also shows the information flow for the regression learning method is given below in Algorithm 1. First an initial estimation of marginal utilities are obtained using M data points. Then the best value of λ is found using an iterative search by the designer

$$\lambda_{n+1} = \lambda_n + \kappa_D\big(\sum_i x_i - B\big), \tag{14}$$

where n is the time step and κ_D is the step size. The corresponding values of x are obtained using the estimated marginal utility curves by setting λ_n as the marginal utility values. By checking the full utilization condition the converging value λ_{new} is obtained. It is important to note that this computation is done by the designer alone and does not require any player involvement. The converged value λ_{new} is sent to the players as the new prices, and new actions x_{new} are observed. The noisy version of value of λ_{new} (which is the value of the function at x_{new}) and x_{new} are added next to the initial data set. Using the regression this new data set gives a better estimate of marginal utilities near the optimal point. From this new estimate of marginal utilities the iteration given by equation (14) is run to obtain a new converging value of λ and corresponding values of x. This online learning and estimation is repeated till end of iteration.

Algorithm 1: Regression Learning of User utilities in Pricing Mechanisms

Input: *Designer*: Global objective.
Input: *Players (users)*: Utility functions $U_i(x_i)$
Result: Learned utility functions $\tilde{U_i}(x) \ \forall i$, optimal prices, and efficient allocation vector x^*

Initialization: The designer obtains initial data points by selecting values for the Lagrangian λ and makes an initial estimate of $\tilde{U_i}$ for each user i using GP by setting the prices accordingly and observing user responses;
repeat
  begin *Designer:*
    Update the value of λ using $\lambda_{n+1} = \lambda_n + \kappa_D\big(\sum_i x_i - B\big)$;
    Using $\tilde{U_i}$, find the corresponding values of x;
    Continue until $\sum_i x_i = B$ **and denote the corresponding** λ_n **as** λ_{new};
    Set λ_{new} **as the user prices,** P_i ;
    begin *Players:*
      foreach *Player* i **do**
        Take action x_{inew} as response to the prices P_i;
      end
    end
    Observe the player actions $x_{inew} \ \forall i, m$;
    Add the values of λ_{new} and x_{new} to the initial data set points;
    Update user utility estimates $\tilde{U_i}$ and variances v_i for all the users based on the updated data set using GP;
  end
until *convergence*;

Note that by using the online learning algorithm as above, when the user preferences or parameters in utility function change in the course of time, the designer can estimate the new functions and can move the system to optimal point. The numerical results which illustrate the learned functions and convergence of the algorithm are given in Section V. We can see that by using this online learning algorithm, the designer can adapt the estimation if the utility functions or utility parameters of the players change in the course of time.

IV. Learning in Auctions

We consider next iterative auctions similar to iterative combinatorial auction or English auction for a divisible good. The players decide on their bids or actions by minimizing their cost which is a combination of their own utilities and prices imposed by the designer. Specifically, the designer $\mathcal{D}$ imposes on a player $i \in \mathcal{A}$ a user-specific resource allocation rule, $Q_i(x)$, pricing, $P_i(x)$ from the the vector of player bids x. We consider here an additive resource sharing scenario where the

players bid for a fixed divisible resource B and are allocated their share captured by the vector $Q = [Q_1, \ldots, Q_N]$ subject to the resource constraint $\sum_i Q_i \leq B$.

The total payment by the i^{th} player is $c_i(x) = P_i(x)Q_i(x)$. The player utility function U_i is separable, i.e. it depends only on the individual allocation of the player. It is also assumed to be continuous, strictly concave, and twice differentiable in terms of its argument Q_i.

From a player's perspective, who takes myopic best response to the price and allocation given by the designer and tries to minimize its cost in terms of the actual resources obtained, the condition

$$\frac{\partial J_i}{\partial Q_i} = \frac{\partial c_i}{\partial Q_i} - \frac{\partial U_i}{\partial Q_i} = c_i' - U_i'$$

is necessary and sufficient for optimality. Suppressing the dependence of user cost on bids x, for the cost minimization, it has to satisfy

$$P_i(Q) = \frac{\partial U_i(Q_i)}{\partial Q_i} \quad \forall i \in \mathcal{A}. \tag{15}$$

Furthermore, if additional assumptions are made on $J_i(x)$, it can be shown that the game admits a unique NE, Q^* (or x^*) [11].

Different from players, the designer $\mathcal{D}$ has two objectives: maximizing the sum of utilities of players and allocating all of the existing resource B, i.e. its full utilization. Hence, the designer $\mathcal{D}$ solves the constrained optimization problem

$$\max_Q V(Q) \Leftrightarrow \max_Q \sum_i U_i(Q_i) \text{ such that } \sum_i Q_i \leq B, \tag{16}$$

in order to find a globally optimal allocation Q that satisfies this **efficiency criterion**. The associated Lagrangian function is then

$$L(Q) = \sum_i U_i(Q_i) + \lambda \left(B - \sum_i Q_i \right),$$

where $\lambda > 0$ is a scalar Lagrange multiplier. Under the convexity assumptions made, this leads to

$$\frac{\partial L}{\partial Q_i} \Rightarrow U_i'(Q_i) = \lambda, \ \forall i \in \mathcal{A}, \tag{17}$$

and the efficiency constraint

$$\frac{\partial L}{\partial \lambda} \Rightarrow \sum_i Q_i = B. \tag{18}$$

In the specific resource sharing setting defined, an auction-based mechanism, $\mathcal{M}^a$, can be defined based on the bid of player i,

$$x_i := P_i(x)Q_i(x), \tag{19}$$

the pricing function

$$P_i := \frac{\sum_{j \neq i} x_j + \omega}{B}, \tag{20}$$

for a scalar $\omega > 0$ sufficiently large such that $\sum_i Q_i \leq B$, and the resource allocation rule

$$Q_i := \frac{x_i}{\sum_{j \neq i} x_j + \omega} B. \tag{21}$$

which is differentiable. It is also possible to interpret the scalar ω as a *reserve bid* [12]. Note that the bid of each player is her willingness to pay i.e. the total amount she pays is her bid $c_i(x) = x_i$. The cost function for the mechanism $\mathcal{M}^a$ becomes in this case,

$$J_i(x) = x_i - U_i(Q_i(x)). \tag{22}$$

Let us denote

$$S_i = \sum_{j \neq i} x_j + \omega,$$

and then equations (20) and (21) become

$$P_i := \frac{S_i}{B}, \tag{23}$$

and

$$Q_i := \frac{x_i}{S_i} B. \tag{24}$$

We obtain the best response as,

$$Q_i^* = \left(\frac{\partial U_i}{\partial Q_i} \right)^{-1} \left(\frac{S_i}{B} \right), \tag{25}$$

where S_i/B is the argument of the inverse marginal utility function.

From the general condition in equation (15), the marginal utility is equal to the price

$$\frac{\partial U_i}{\partial Q_i} = \frac{S_i}{B} = \frac{\sum_{j \neq i} x_j + \omega}{B}. \tag{26}$$

This equation can be rewritten as following

$$\frac{\partial U_i}{\partial Q_i}(x_i) = \psi(x) - \frac{x_i}{B}. \tag{27}$$

where

$$\psi(x) = \frac{\sum_j x_j + \omega}{B}. \tag{28}$$

As in pricing, GP regression learning is used now to learn the marginal utilities in auctions. For an initial value of ω, an initial estimate of the marginal utilities of players are constructed. The values of Q_i's will give the corresponding values of x_i's for this initial value of ω.

Next, the value of ψ is varied over space of all possible values, by changing the value of the reserve bid ω. This search algorithm provides the value of λ for which $\sum_i Q_i = B$ for any general utility function and the corresponding value of ω. This ω is then used to set the price and allocation, using which the bids will converge to the efficient point. Since the reserve bid is an independent parameter which does not depend on user bids, the incentive compatible property of the mechanism still holds.

To illustrate the approach, consider the case of logarithmic utility function weighted by a positive scalar parameter α, i.e.,

$$U_i = \alpha_i \log Q_i \ \ \forall i \in \mathcal{A}.$$

The best response is $x_i^* = \alpha_i$. The unknown α's are then learned in single step from the bid which corresponds to optimal point.

Consider next the alternative case of exponential user utilities,

$$U_i = 1 - e^{-\alpha_i Q_i} \quad \forall i \in \mathcal{A}.$$

In this case

$$x_i^* = \frac{S_i}{B\alpha_i} \log \frac{B}{S_i}.$$

So to learn α's an iteration is needed and the optimal prices based on these correct α's will take the system to approximately efficient point.

In the case of general user utilities, however, multiple steps of the Algorithm 1 are required in order for the designer to characterize user utilities with sufficient accuracy and the outcome converges to the optimal solution.

V. Numerical Results

In this section we provide some numerical results that illustrate our theoretical analysis. We consider a system with 5 users having scalar parameterized logarithmic utility functions in order to visualize the results. The learned marginal utility curves of the users are plotted and compared with the actual curves in Figure 1 for the pricing mechanism given in section III with 5 initial data points. Next, learned marginal utility curves of the users are plotted and compared with the actual curves in Figure 2 with 15 initial data points. We observe that as the number of data points increases from 5 to 15 the marginal utility curves are estimated more accurately.

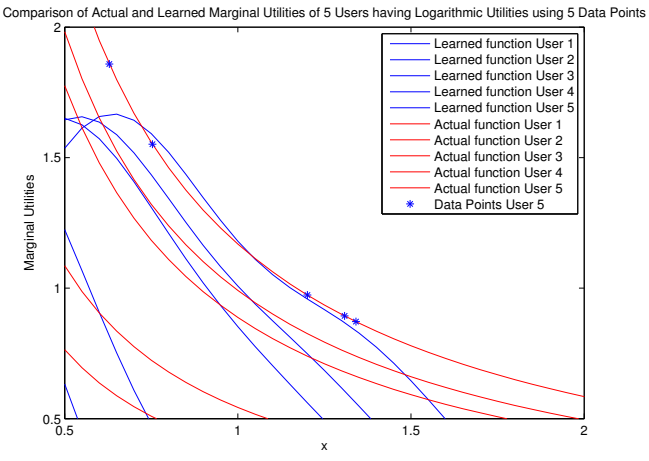

Fig. 1. Marginal Utility curve for logarithmic utilities constructed by GP regression and actual ones for $M = 5$.

In Figure 3, the actual marginal utility curves for 3 users with logarithmic utilities are compared with marginal utility curves constructed using initial data points and the online algorithm given in Algorithm 1. We can observe that near the optimal lambda value the estimation of the function is better with the online algorithm than with only initial data points, as expected.

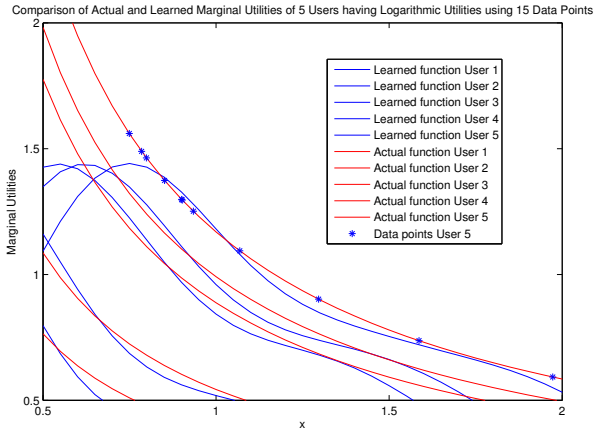

Fig. 2. Marginal Utility curve for logarithmic utilities constructed by GP regression and actual ones for $M = 15$.

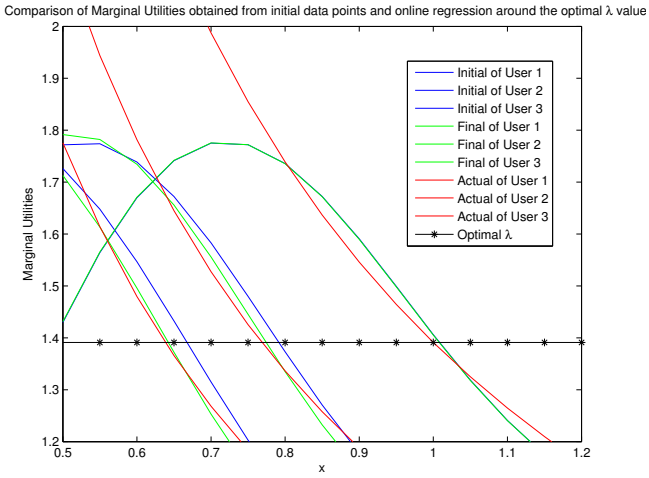

Fig. 3. Marginal Utility curve for logarithmic utilities constructed using initial data points and the online algorithm given in Algorithm 1.

We next plot in Figure 4 social welfare (SW) and the value of λ of 5 users having scalar parameterized logarithmic utility functions with the number of data points varying from 3 to 25. We observe that the estimated social welfare and λ using the learned marginal utility functions improve and approaches the analytically obtained social welfare as the number of data points increase.

VI. Conclusion

This paper used learning techniques as a tool for estimating the utilities of the players by a mechanism designer. We considered the problem of allocation of a divisible resource by a designer to a number of players having infinite dimensional utility functions and designer employing Gaussian process regression method to obtain the marginal utility functions of the players. In the pricing mechanisms, the Lagrange multiplier of the total resource constraint, which is set as the price for

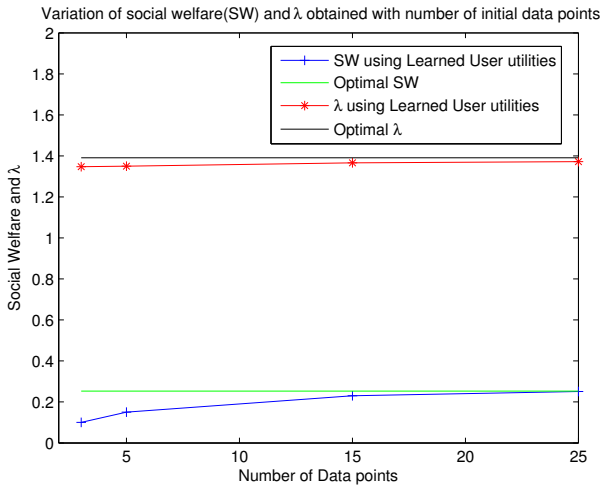

Fig. 4. Marginal Utility curve for logarithmic utilities constructed by GP regression.

all the users, is used to navigate the allocation to the efficient point using the estimated marginal utilities. In auctions, the reserve bid parameter in the pricing and allocation rule is varied for obtaining near efficient point. We proposed an online algorithm which uses the best response action of users in each time instance to give a better estimate of the utilities, near the efficient point.

As future work we plan to extend the results in this paper to the case of mechanisms with non-separable user utilities.

REFERENCES

[1] D. Fudenberg and D. Levine, *The Theory of Learning in Games*. The MIT Press, 1998.

[2] N. Nissan, T. Roughgarden, and E. Tardos, *Algorithmic Game Theory*. Cambridge University Press, 2007.

[3] M.-F. Balcan, A. Blum, J. D. Hartline, and Y. Mansour, "Reducing mechanism design to algorithm design via machine learning," *J. Comput. Syst. Sci.*, vol. 74, pp. 1245–1270, December 2008. [Online]. Available: http://portal.acm.org/citation.cfm?id=1460945.1461324

[4] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005. [Online]. Available: http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/026218253X

[5] R. Johari, S. Mannor, and J. Tsitsiklis, "Efficiency loss in a network resource allocation game: the case of elastic supply," *IEEE Transactions on Automatic Control*, vol. 50, no. 11, pp. 1712–1724, November 2005.

[6] S. J. Kang, Y. Won, S. Lim, and M. van der Schaar, "Efficient resource management with reduced overhead information," in *in IEEE Conference on Personal, Indoor and Mobile Radio Communications*, September 2009, pp. 1452– 1456.

[7] D. C. Parkes and L. H. Ungar, "Iterative combinatorial auctions: Theory and practice," in *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*. AAAI Press, 2000, pp. 74–81. [Online]. Available: http://portal.acm.org/citation.cfm?id=647288.721579

[8] M. T. Hajiaghayi, R. Kleinberg, and D. C. Parkes, "Adaptive limited-supply online auctions," in *In Proceedings of the 5th ACM Conference on Electronic Commerce*. ACM Press, 2004, pp. 71–80.

[9] O. Dekel, F. Fischer, and A. D. Procaccia, "Incentive compatible regression learning," in *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, ser. SODA '08. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2008, pp. 884–893. [Online]. Available: http://portal.acm.org/citation.cfm?id=1347082.1347179

[10] B. Scholkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press, 2001.

[11] T. Başar and G. J. Olsder, *Dynamic Noncooperative Game Theory*, 2nd ed. Philadelphia, PA: SIAM, 1999.

[12] J. Huang, R. Berry, and M. Honig, "Auction–based Spectrum Sharing," *ACM Mobile Networks and Applications Journal*, vol. 24, no. 5, pp. 405–418, June 2006.

VII. APPENDIX

Definitions:

The properties of mechanisms considered in this paper can be formally defined as follows.

Definition 1: **Efficiency:** Efficient mechanisms maximize designer objective, i.e. they solve the problem $\max_x V(x, U_i(x), c_i(x))$.

Definition 2: **Nash Equilibrium:** The strategy profile $x^* = [x_1^*, \ldots, x_N^*]$ is in Nash Equilibrium if the cost of each player is minimized at the equilibrium given the best strategies of other players.

$$J_i(x_i^*, x_{-i}^*) \leq J_i(x_i, x_{-i}^*), \forall i \in \mathcal{A}, x_i \in \mathcal{X}_i$$

Definition 3: **Dominant Strategy Equilibrium:** The strategy profile $\tilde{x} = [\tilde{x}_1, \ldots, \tilde{x}_N]$ is in Dominant Strategy Equilibrium if the cost of each player is minimized at the equilibrium irrespective of the strategies of other players.

$$J_i(\tilde{x_i}, x_{-i}) \leq J_i(x_i, x_{-i}), \forall i \in \mathcal{A}, x_i \in \mathcal{X}_i, x_{-i} \in \mathcal{X}_{-i}$$