Inverse Modeling of Non-Cooperative Agents via Mixture of Utilities

Ioannis C. Konstantakopoulos, Lillian J. Ratliff, Ming Jin, Costas Spanos, S. Shankar Sastry

Abstract— We describe a new method of parametric utility learning for non-cooperative, continuous games using a probabilistic interpretation for combining multiple utility functions—thereby creating a *mixture of utilities*—under non-spherical noise terms. This framework allows for the estimated parameters of the learned utility functions to depend on the historical actions of the players and allows us to capture the fact that players' utility functions are not static. In particular, we present an adaptation of *mixture of regression models* that takes in to account heteroskedasticity. We show the performance of the proposed method by estimating the utility functions of players using data from a social game experiment designed to encourage energy efficient behavior amongst building occupants. Using occupant voting data we simulate the new game defined by the estimated mixture of utilities and show that the resulting forecast is more accurate than robust utility learning methods such as constrained Feasible Generalized Least Squares (cFGLS), ensemble methods such as bagging, and classical methods such as Ordinary Least Squares (OLS).

I. INTRODUCTION

Due to an increase in technology, there are new sensing and actuation platforms that are being deployed. As a result, humans are being integrated into the decision-making process for operations and management of large-scale systems such as the smart grid and intelligent transportation network. We can now observe how end-users in these systems are consuming resources and, moreover, what their individual preferences are with regard to resource consumption. Methods of learning how humans make decisions and how they interact with their environment are needed in order to account for their impact on the system and control for it via cyber-physical control schemes or economic mechanisms such as incentives.

In our past work [13], [14], we developed tools for estimating parameters of users' utility functions and designing incentives to encourage socially optimal and efficient behaviors. At the core of our approach is the fact that we modeled the agents as non-cooperative agents who play according to a *Nash equilibrium strategy*. This serves the purpose of modeling the agents as strategic entities who make decisions based on their own preferences in spite of others. The game theoretic framework both allows for qualitative insights to be made about the outcome of such selfish behavior and, more importantly, can be leveraged in designing *incentives* for adjusting agents strategies.

We extended those results to a robust utility estimation framework in [15]. In particular, assuming a parametric form of utility function for each agent, we utilized constrained Feasible Generalized Least Squares (cFGLS) to formulate a parameter estimation scheme in which the estimator variance is reduced, unbiased, and consistent. We explored wild bootstraping, a powerful technique for asymptotic approximation of the bias and standard error of an estimator in a complex and noisy statistical model. Strengthened by the bootstrap estimators, we improved the parameter estimation scheme by using ensemble methods such as bagging.

In the present work, we develop a novel framework for estimating a mixture of utility functions by extendeding the classical *mixture of regression models* framework to inverse modeling of utility functions in non–cooperative, continuous games. We show that the proposed method outperforms many other robust utility learning frameworks including those in our previous works such as cFGLS, enemble methods such as bagging, and several other standard techniques such as OLS and our preliminary estimation scheme in [13].

In particular, we present the theoretical formulation of a new parametric utility learning method that uses a probabilistic interpretation—i.e. a mixture of utilities—of agent utility functions. The mixture of utilities modeling paradigm allows us to account for variations in agents' parameters over time. The resulting scheme is a *Mixture of constrained Feasible Generalized Least Squares* (Mix-cFGLS) that uses heteroskedasticity inference for correlated errors in the resulting regression model. Mix-cFGLS is a complex statistical model that we show is a powerful tool for utility learning that providing greater accuracy in the prediction of players responses. Furthermore, captured in this framework is the fact that players' utility functions are not static; instead, the parameters of players utility functions can depend on historical data. Inherent to the Mix-cFGLS framework is the fact that we can explore the tradeoff between minimizing bias and minimizing the variance of predictions. We show that by using Mix-cFGLS for utility learning, allowing for a small amount of bias results in a substantial decrease in variance and forecasting accuracy.

The rest of the paper is organized as follows. We begin in Section II by describing a social game framework for an experiment that we conducted on the UC Berkeley campus and and formulating the decision making model for agents.

In Section III we propose the robust parametric learning framework for a mixture of utilities which uses probabilistic softmax gates. We present the results of our estimation scheme in Section IV using data from the social game for energy efficiency. We conclude with some discussion and proposal for future work in Section V. We remark that while we present our results in the context of the social game for energy efficient behavior in buildings, it is general enough to apply to many other applications.

II. USER DECISION MAKING FRAMEWORK

In this section, we briefly describe the social game experimental setup and we introduce the non–cooperative game framework between users in the social game experiment. We refer the reader to our previous works [13], [14] for a more detailed description of the social game.

A. Social Game

We designed and implemented a social game for encouraging energy efficiency in a collaboratory which resides in the Center for Research in Energy Systems Transformation (CREST) on the Berkeley campus. We have deployed an automated lighting control system (Lutron system[1]), which enables its occupants to adjust the lighting through a web portal. In particular, using this web portal, the social game consists of occupants who select a lighting setting by balancing their preferences over comfort, productivity, desired to be *green* and desire to win a prize. The portal also allows for users to visualize the social game—in particular, the dim level of the lights and the energy efficiency level of occupants—as well as view the point levels and historical votes of all occupants.

The game is designed to leverage interactions amongst occupants, who win points based on how energy efficient their vote is compared to others. The occupants select their desired lighting dim level in the continuous interval $[0, 100]$ (0 being off, and 100 being the maximum level of lighting). The occupants can vote as frequently as they like and the average of all the occupants' current votes sets the implemented dim level in the collaboratory.

Three persistent user behavior profiles have emerged: those who actively participate through voting, those who are present yet keep their vote at the default value, and those who are absent (i.e. do not participate in the game). Experiments with four default levels, namely $\{10, 20, 60, 90\}$, have been conducted, covering a spectrum of lighting conditions from being dim to bright. To enforce the rule that those who are not present in the space cannot vote remotely, we executed a simple presence detection algorithm based on their power usage [16].

B. Occupant Decision Making

In the previous work [13], we modeled the interaction between the building manager and the occupants as a leader–follower(s) game. We designed the incentives by estimating the parameters using an OLS framework and optimized the

points and default lighting value using the estimated utilities of the occupants. This work is based on the theoretical utility learning and incentive design framework presented in [17]. In a follow-up work [15], we extended the utility learning framework to a more robust learning scheme using cFGLS and ensemble methods such as bagging. In the present work, we take a step back and re-examine the utility learning step using statistical methods that provide greater accuracy in the estimation and prediction of player decision-making. In future work, we will fold the new estimation scheme into the overall incentive design framework.

Let the number of occupants participating in the game be denoted by n. We model the occupants as utility maximizers having utility functions that are a mixture of utilities where the base utility function is composed of two terms that capture the tradeoff between comfort and desire to win. Let us first describe this base utility function before we dive into the mixture of utilities model.

We model an occupant's comfort level using a Taguchi loss function [18] which is interpreted as modeling dissatisfaction in such a way that it is increasing as variation increases from their selected lighting setting. In particular, each occupant has the following *comfort* function:

$$\psi_i(x_i, x_{-i}) = -(\bar{x} - x_i)^2 \tag{1}$$

where $x_i \in \mathbb{R}$ is occupant i's lighting vote, $x_{-i} = \{x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n\}$, and

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i \tag{2}$$

is the average of all the occupant votes and is the lighting setting which is implemented. Hence, this term measures the discomfort an occupant feels given that its vote is x_i and the state of the environment is actually $\bar{x}$.

In addition, each occupant has the following *winning* function:

$$\phi_i(x_i, x_{-i}) = -\rho c (x_i)^2 \tag{3}$$

where ρ is the total number of points distributed by the building manager and c is a scaling factor which we set to 10^{-4}. The points are distributed by the leader using the relationship

$$\rho \frac{x_b - x_i}{n x_b - \sum_{j=1}^{n} x_j} \tag{4}$$

where $x_b = 90$ is the baseline setting for the lights, i.e. the lighting setting that occurred before the implementation of the social game in the office.

Let each agent's decision x_i be constrained to lie in the set $S_i = [0, 100]$. Then the constraint set can be described as follows. Let $h_{i,j}(x_i, x_{-i})$ for $j \in \{1, 2\}$ denote the constraints on occupant i's optimization problem. For occupant i, the constraints are described by

$$h_{i,1}(x_i) = 100 - x_i \tag{5a}$$
$$h_{i,2}(x_i) = x_i \tag{5b}$$

Let $\mathcal{C}_i = \{x_i \in \mathbb{R} | \ h_{i,j}(x_i) \geq 0, \ j \in \{1, 2\}\}$ and $\mathcal{C} = \mathcal{C}_1 \times \cdots \times \mathcal{C}_n$.

In our previous works, each occupant's utility function was

given by

$$f_i(x_i, x_{-i}) = \psi_i(x_i, x_{-i}) + \theta_i \phi_i(x_i, x_{-i}) \qquad (6)$$

where θ_i is parameter unknown to the leader, the source of asymmetric information in the leader–follower(s) game. We propose that for each player there will be M decision making profiles each with a different parameter θ_s^i, $s \in \{1, \ldots, M\}$ so that each player's utility function is described by a mixture of these profiles. In particular, each player's utility is given by

$$f_i(x_i, x_{-i}) = \psi_i(x_i, x_{-i}) + \theta_i^M(x_i, x_{-i})\phi_i(x_i, x_{-i}) \quad (7)$$

where $\theta_i^M(x_i, x_{-i})$ is the mixture model unknown to the leader (e.g. building manager) and has the following form:

$$\theta_i^M(x_i, x_{-i}) = \sum_{s=1}^{M} \pi_s^i(x^p, \xi_s^i)\theta_s^i \qquad (8)$$

where $\pi_j^i(x^p, \xi_j^i)$ is a softmax function defined by

$$\pi_j^i(x^p, \xi_j^i) = \frac{\exp\left(-(\xi_j^i)^\top x^p\right)}{\sum_{s=1}^{M} \exp\left(-(\xi_s^i)^\top x^p\right)} \qquad (9)$$

where ξ_s^i's are the different vectors that govern the softmax function and characterize distribution probability of each θ_s. The ξ_s^i's are unknown to leader and must be learned along with the parameters θ_s^i. The softmax function assigns a probability to each θ_s^i based indirectly on past actions through $x^p = \begin{bmatrix} D_i h_{i,1}(x_i^p) & D_i h_{i,2}(x_i^p) & D_i \phi_i(x^p) \end{bmatrix}$ and it is a continuous probability density and sums to one— i.e. $\sum_{l=1}^{M} \pi_l^i(x^p, \xi_l^i) = 1$. More specifically, the unknown parameters θ_i^M vary using a learned softmax function driven by agents' past actions.

The reason we adopt the mixture of utilities approach where parameters can depend on past actions is that characterizing agents' behavior with a static parameter is a difficult task for accurate, long-run predictions since agents' preferences may change over time or be affected by random external signals.

The i-th occupant faces the following optimization problem:

$$\max_{x_i \in S_i} f_i(x_i, x_{-i}) \qquad (10)$$

where $S_i = [0, 100] \subset \mathbb{R}$ is the constraint set for x_i. In this framework, the occupants are non-cooperative agents in a continuous game with convex constraints. We model their interaction using the Nash equilibrium concept.

Definition 1: A point $x \in \mathcal{C}$ is a **Nash equilibrium** for the game $(f_1, \ldots, f_n)$ on $\mathcal{C}$ if

$$f_i(x_i, x_{-i}) \geq f_i(x_i', x_{-i}) \quad \forall \, x_i' \in \mathcal{C}_i \qquad (11)$$

for each $i \in \{1, \ldots, n\}$.
The interpretation of the definition of Nash is as follows: no player can unilaterally deviate and increase their utility. Additional constraints on the parameters $\{\theta_i^M\}_{i=1}^n$ ensure that the game is a concave n-person game on a convex set.

Theorem 1 (Rosen [19]): A Nash equilibrium exists for every concave n-person game.

Define the Lagrangian of each player's optimization problem as follows:

$$L_i(x_i, x_{-i}, \mu_i) = f_i(x_i, x_{-i}) + \sum_{j \in A_i(x_i)} \mu_{i,j} h_{i,j}(x_i) \ (12)$$

where $A_i(x_i)$ is the active constraint set at x_i. The differential game form [20], [17] is given by

$$\omega(x, \mu) = [D_1 L_1(x, \mu_1)^T \ \cdots \ D_n L_n(x, \mu_n)^T]^T \qquad (13)$$

where $D_i L_i$ denotes the derivative of L_i with respect to x_i.

Definition 2 (Ratliff [17]): A point $x^* \in \mathcal{C}$ is a differential Nash equilibrium for the game $(f_1, \ldots, f_n)$ on $\mathcal{C}$ if $\omega(x^*, \mu^*) = 0$, $z^T D_{ii} L_i(x^*, \mu_i^*)z < 0$ for all $z \neq 0$ such that $D_i h_{i,j}(x_i^*)^T z = 0$, and $\mu_{i,j} > 0$ for $j \in A_i(x_i^*)$. These conditions are sufficient for defining a local Nash equilibrium.

Proposition 1 (Ratliff [13]): A differential Nash equilibrium of the n-person concave game $(f_1, \ldots, f_n)$ on $\mathcal{C}$ is a Nash equilibrium.
A sufficient condition guaranteeing that a Nash equilibrium x is isolated is that the Jacobian of $\omega(x, \mu)$, denoted $D\omega(x, \mu)$, is invertible [17]. We refer to such points as being *non-degenerate*. The above characterization of Nash are useful in formulating the utility learning problem as they provide, first, a set of stationarity conditions that can be exploited and, second, a mechanism for computing Nash—projected gradient descent, which converges for stable Nash equilibria.

III. UTILITY LEARNING FRAMEWORK

In this section, we present the utility learning framework under a non–spherical noise assumption and the theoretical formulation of the utility learning framework using the mixture of utilities model of decision-making for players. Note that we will use the words *learner* and *estimator* to mean the same thing.

A. Utility Estimation Under Non-Spherical Noise

The utility estimation problem can be formulated as a convex optimization problem by using first– and second–order conditions for Nash equilibria [17], [20]. Strengthen by the constraints that guarantee unique Nash equilibria our utility learning method can be viewed as a constrained regression model.

Let K_i denote the number of data points for player i. We assume that each observation $x^{(k)}$ corresponds to an ε– approximate Nash equilibrium where the superscript notation $(\cdot)^{(k)}$ indicates the k-th observation. Thus, we can consider first-order optimality conditions for each occupants optimization problem and define a residual function capturing the amount of suboptimality of $x_i^{(k)}$ [21], [22]. Indeed, let the residual of the stationarity and complementary conditions for occupant i's optimization problem be given by

$$r_{\text{s},i}^{(k)}(\theta_i, \mu_i) = D_i f_i(x_i^{(k)}, x_{-i}^{(k)}) + \sum_{j=1}^{2} \mu_i^j D_i h_{i,j}(x_i^{(k)}) \tag{14}$$

and $r_{\text{c},i}^{j,(k)}(\mu) = \mu_i^j h_{i,j}(x_i^{(k)})$, $j \in \{1, 2\}$, respectively. Define $r_{\text{s}}^{(k)}(\theta) = [r_{\text{s},1}^{(k)}(\theta_1, \mu_1) \ \cdots \ r_{\text{s},n}^{(k)}(\theta_n, \mu_n)]^T$ and

$r_c^{(k)} = [r_{c,1}^{(k)}(\mu_1) \cdots r_{c,n}^{(k)}(\mu_n)]^T$ where $r_{c,i}^{(k)}(\mu_i) = [r_{c,i}^{1,(k)}(\mu_i) \ r_{c,i}^{2,(k)}(\mu_i)]$ and $\mu_i = (\mu_i^1, \mu_i^2)$. Then, given the data from the occupants' actions, we solve the following convex optimization problem:

$$\min_{\mu,\theta} \sum_{k=1}^K \chi(r_s^{(k)}(\theta,\mu), r_c^{(k)}(\mu)) \tag{P1}$$
$$\text{s.t. } \theta_i \geq \theta_{LB}, \mu_i \geq 0 \ \ \forall \, i \in \{1,\ldots,n\}$$

where θ_{LB} is a lower bound for the unknown parameters $\{\theta_i\}_{i=1}^n$ that ensures the inferred game is concave and $\chi : \mathbb{R}^n \times \mathbb{R}^{2n} \to \mathbb{R}_+$ is a nonnegative, convex penalty function satisfying $\chi(z_1, z_2) = 0$ if and only if $z_1 = 0$ and $z_2 = 0$, i.e. any norm on $\mathbb{R}^n \times \mathbb{R}^{2n}$, the inequality $\mu_i \geq 0$ is element-wise.

To determine θ_{LB} we utilize the second derivative condition on players' utility functions; in particular, if $D_{ii}^2 f_i(x) = -2(1 - 1/n)^2 - 2\theta_i c\rho < 0$ for each i, then the game is concave. Hence, $\theta_i > -c^{-1}\rho^{-1}(1 - n^{-1})^2$ where the right-hand side is a a negative non-increasing function of n. Using a relaxation, concavity is ensured regardless of the number of players by setting $n = 2$, the minimum number of users in a non-cooperative game. Then, given fixed ρ and $0 < \zeta << 1$, the lower bound $\bar{\theta}_{LB} = -0.3571 + \zeta$ will guarantee the estimated game is concave. The subgradient projection method applied to the gradient dynamics $\dot{x} = [D_1 f_1(x)^T \cdots D_n f_n(x)^T]^T$ and the constraint set defined by (5b) are known to converge to a differential Nash equilibrium of the constrained n-person concave game [23] and we know the differential Nash equilibrium is unique if the game Hessian, given by

$$H = \begin{bmatrix} D_{11}f_1 & \cdots & D_{1n}f_1 \\ \vdots & \ddots & \vdots \\ D_{n1}f_n & \cdots & D_{nn}f_n \end{bmatrix}, \tag{15}$$

is positive definite [20, Theorem 2]. This is automatically guaranteed for $n \geq 4$ provided the constraint defined by $\bar{\theta}_{LB}$ using $\zeta = 10^{-2}$; this is straightforward to verify by determining the eigenvalues of H as n varies via the method described in [24]. Hence, we use a lower bound on the θ_i's in (P1) that guarantees our the game is not only concave but has a unique differential Nash equilibrium. Indeed, we set $\theta_{LB} = -c^{-1}\rho^{-1}(1 - 4^{-1})^2 + \zeta = -0.8035 + \zeta$ for a given ρ and $0 < \zeta << 1$.

Now, we convert (P1) to a standard estimation framework. Define the regressor-design matrix $X = \text{diag}(X_1 \cdots X_n)$ where $X_i = [(X_i^{(1)})^T \cdots (X_i^{(K_i)})^T]^T$,

$$X_i^{(k)} = \begin{bmatrix} D_i h_{i,1}(x_i^{(k)}) & D_i h_{i,2}(x_i^{(k)}) & D_i \phi_i(x^{(k)})) \\ h_{i,1}(x_i^{(k)}) & 0 & 0 \\ 0 & h_{i,2}(x_i^{(k)}) & 0 \end{bmatrix}, \tag{16}$$

the observation-dependent vector $Y = [Y_1 \cdots Y_n]^T$ where $Y_i = [-D_i \psi_i(x^{(1)}) \ 0 \ 0 \ \cdots \ -D_i \psi_i(x^{(K_i)}) \ 0 \ 0]^T$ and the regression coefficient-learner $\beta = [\mu_1^1 \ \mu_1^2 \ \theta_1 \ \cdots \ \mu_n^1 \ \mu_n^2 \ \theta_n]^T$.

Using an Euclidean norm on $\mathbb{R}^n \times \mathbb{R}^{2n}$ for χ in (P1) leads to a constrained OLS (cOLS) problem:

$$\min_{\beta} \{\|Y - X\beta\|_2 | \ \beta > \beta_{LB}\} \tag{P2}$$

where $\beta_{LB} = [0 \ 0 \ \theta_{LB} \cdots 0 \ 0 \ \theta_{LB}]^T$. Our data generation process, as described by problem (P2), is a classical multiple linear regression with inequality constraints as follows

$$Y = X\beta + \epsilon, \ \beta > \beta_{LB} \tag{17}$$

where $\epsilon = (\epsilon_1, \ldots, \epsilon_n)$ is a spherical error term following: $E(\epsilon|X) = 0^{n_d \times 1}$ and $\text{cov}(\epsilon|X) = \sigma^2 I^{n_d \times n_d}$ where n_d is the total data points.

The data generation process (17) lacks robustness in presence of non-spherical noise, results in biased utility learners (parameter estimates), and performs poorly in forecasting players decision making. Robustness can be ensured by assuming heteroskedasticity [25, Chapter 5]. Moreover, heteroskedasticity allows for inference of correlated errors in the resulting regression model which can be used to determine the relationship between agents decision-making processes. In our data generation model, we adopt a non-spherical standard error ϵ and it is robust for data generation processes in which the error terms do not follow constant variance or are correlated. Mathematically the non-spherical error terms are modeled by

$$\text{cov}(\epsilon|X) = G \succ 0, \ G \in \mathbb{R}^{n_d \times n_d}. \tag{18}$$

The model's standard error ϵ is drawn from multivariable normal probability distribution with zero mean and different variances and ϵ models autocorrelated events. Hence, we have a complex statistical model that captures possible interactions between users through correlations between their decision-making models.

On the other hand, under a non-spherical standard error the cOLS estimator is biased and does not satisfy the Gauss–Markov theorem for Best Linear Unbiased Estimator (BLUE). By multiplying (17) on the left with $G^{-\frac{1}{2}}$, we can derive an unbiased estimator which satisfies the BLUE property [25]. The resulting constrained Generalized Least Squares (cGLS) statistical model is given by

$$(G^{-\frac{1}{2}}Y) = (G^{-\frac{1}{2}}X)\beta + (G^{-\frac{1}{2}}\epsilon), \ \beta > \beta_{LB}. \tag{19}$$

In real applications, like the utility learning problem, the explicit form of $\text{cov}(\epsilon|X) = G$ is unknown. Due to the fact that we have many unknown parameters for our problem's given data points, using the residuals from the resulting cOLS regression learner (17), we apply noise inference by imposing structural constraints on G matrix.

We impose two structures for the non-spherical error term. The first is known as the *Freedman Noise Structure* [25, Chapter 5], [26] and is given by $\hat{G} = \text{diag}(\tilde{K}, \cdots, \tilde{K}) \in \mathbb{R}^{n_d \times n_d}$ where $\tilde{K} \in \mathbb{R}^{K_i \times K_i}$. In the utility learning problem for the social game described in the previous section, $K_i = 3$ and

$$\tilde{K} = \begin{bmatrix} \tilde{K}_{11} & \tilde{K}_{12} & \tilde{K}_{13} \\ \tilde{K}_{21} & \tilde{K}_{22} & \tilde{K}_{23} \\ \tilde{K}_{31} & \tilde{K}_{32} & \tilde{K}_{33} \end{bmatrix} \in \mathbb{R}^{3 \times 3} \tag{20}$$

where $\tilde{K}_{11} = \frac{3}{n_d}\sum_{i=1}^{\frac{n_d}{3}} e_{3i-1}^2$, $\tilde{K}_{22} = \frac{3}{n_d}\sum_{i=1}^{\frac{n_d}{3}} e_{3i-2}^2$, $\tilde{K}_{33} = \frac{3}{n_d}\sum_{i=1}^{\frac{n_d}{3}} e_{3i}^2$, $\tilde{K}_{1,3} = \tilde{K}_{3,1} = \frac{3}{n_d}\sum_{i=1}^{\frac{n_d}{3}} e_{3i-1}e_{3i}$, $\tilde{K}_{2,3} = \tilde{K}_{3,2} = \frac{3}{n_d}\sum_{i=1}^{\frac{n_d}{3}} e_{3i-2}e_{3i}$, $\tilde{K}_{1,2} = \tilde{K}_{2,1} = \frac{3}{n_d}\sum_{i=1}^{\frac{n_d}{3}} e_{3i-1}e_{3i-2}$ and the e_i's are the residuals.

The second is known as the *HC4 Noise Structure* [27] and is given by

$$\hat{G} = \text{diag}(\frac{e_1^2}{(1-h_1)^{\delta_1}}, \frac{e_2^2}{(1-h_2)^{\delta_2}}, \cdots, \frac{e_{n_d}^2}{(1-h_{n_d})^{\delta_{n_d}}}) \quad (21)$$

where $\delta_i = \min\{4, n_d h_i/\sum_{i=1}^{n_d} h_i\}$ and the h_i's are the diagonal elements of $H = X(X^\top X)^{-1}X^\top$. With this structure, the penalty for each residual increases with $h_i/\sum_{i=1}^{n_d} h_i$.

To infer the noise structure $\hat{G}$ and estimate the parameters of the utility functions β, we start with the fitted cOLS learner and use the cOLS residuals to get an initial $\hat{G}$. We substitute the inferred noise, $\hat{G}$, to the cGLS statistical model (19) to get the one-step constrained Feasible GLS (cFGLS) learners. We iterate between the estimation of $\hat{G}$ and β_{cFGLS} either until convergence or for a fixed number of iterations to prevent overfitting.

B. Utility Estimation—Mixture of cFGLS

We extend the above learning framework to a probabilistic setting that allows us to learn a mixture of utilities. A mixture of regression models is a powerful statistical model that splits data points into several subareas while fitting regression learners in each of area's splitted data sets. Unfortunately, in real applications we do not have *a priori* knowledge of each subarea's data structure and density. Thus, the Mix-cFGLS procedure has to learn, in parallel, both the subareas' data point density plus the best regression learners that fit data points in these subareas. This is achievable by the Expectation-Maximization (EM) algorithm over the cost function resulting from the complete likelihood of our statistical model.

The Mix-cFGLS model includes a softmax function as a gate for calculating the distribution probability of each component for each player's utility function. The probabilistic model for player σ is given by

$$p(Y^\sigma|X^\sigma, \beta^\sigma, \xi^\sigma) = \sum_{j=1}^M (\pi_j(x^\sigma, \xi^{\sigma_j}) \cdot N(Y^\sigma|(\beta^{\sigma_j})^\top x^\sigma, G^\sigma \succ 0, \beta^{\sigma_j} > \beta_{LB}^\sigma)) \quad (22)$$

where $\beta^\sigma = [\beta^{\sigma_1} \cdots \beta^{\sigma_M}] \in \mathbb{R}^{3\times M}$, $\xi^\sigma = [\xi^{\sigma_1} \cdots \xi^{\sigma_M}] \in \mathbb{R}^{3\times M}$, $\beta_{LB}^\sigma = [0\ 0\ \theta_{LB}]^T \in \mathbb{R}^{3\times 1}$, K_σ denotes the number of data points for player σ, M is the number of mixture components,

$$Y^\sigma = [-D_\sigma \psi_\sigma(x^{(1)})\ 0\ 0\ \cdots\ -D_\sigma \psi_\sigma(x^{(K_\sigma)})\ 0\ 0]^T$$

is an observation–dependent vector, x^σ is a covariate vector and is a transposed row of the regressor–design matrix $X^\sigma = [(X_\sigma^{(1)})^T \cdots (X_\sigma^{(K_\sigma)})^T]^T$ for player σ, $\beta^{\sigma_j} \in \mathbb{R}^{3\times 1} = [\mu_\sigma^1\ \mu_\sigma^2\ \theta_\sigma]^T$ is the regression coefficient–learner for each mixture component of player σ, $\xi^{\sigma_j} \in \mathbb{R}^{3\times 1}$ is the coefficient of softmax for each mixture component of player

σ, $\pi_j(x, \xi^{\sigma_j})$ for $j = 1, 2, \cdots, M$ is the mixture coefficient probability distribution governed by a softmax function (see (9)), and $G^\sigma \succ 0$ is the non-spherical structure that models the noise term of the statistical process of each player's utility learning problem.

Given a data set for player σ and assuming i.i.d. observations $D^\sigma = \{(x_i^\sigma, y_i^\sigma) : i = 1, \cdots, K_\sigma\}$, where x_i^σ, y_i^σ are the i-th rows of X^σ, Y^σ respectively, the log-likelihood function of the model is given by

$$L(\beta^\sigma, \xi^\sigma|D^\sigma) = \sum_{i=1}^{K_\sigma} \log\Big[\sum_{j=1}^M \pi_j(x_i^\sigma, \xi^{\sigma_j}) \cdot N(Y^\sigma|(\beta^{\sigma_j})^\top x_i^\sigma, G^\sigma \succ 0, \beta^{\sigma_j} > \beta_{LB}^\sigma)\Big] \quad (23)$$

We optimize the cost function (23) using the EM algorithm. In support of this, we introduce a set of binary latent variables $Z^\sigma = \{z_i^\sigma\}$ such that $z_{i,j}^\sigma \in \{0, 1\}$; in particular, for each data point indexed by i, we have a latent variable that indicates which mixture j it belongs to. Given $Z^\sigma = \{z_i^\sigma\}$ we have the complete data set, called $D_C^\sigma = \{(x_i^\sigma, y_i^\sigma, z_i^\sigma) : i = 1, \cdots, K_\sigma\}$. The *complete* log-likelihood for the (22) problem is given by

$$L_C(\beta^\sigma, \xi^\sigma|D_C^\sigma) = \sum_{i=1}^{K_\sigma}\sum_{j=1}^M z_{i,j}^\sigma \log\Big[\pi_j(x_i^\sigma, \xi^{\sigma_j}) \cdot N(y_i^\sigma|(\beta^{\sigma_j})^\top x_i^\sigma, G^\sigma \succ 0, \beta^{\sigma_j} > \beta_{LB}^\sigma)\Big] \quad (24)$$

However, since we do not observe the latent variables Z^σ we compute their posterior probability in the expectation step—*E–step*—of the EM algorithm. The posterior probability of latent variables Z^σ is given by

$$\tau^\sigma(i, j) = E[z^\sigma(i, j)] = p(Z_i^\sigma = 1|X^\sigma, Y^\sigma) = \frac{\pi_j(x_i^\sigma, \xi^{\sigma_j})N(y_i^\sigma|(\beta^{\sigma_j})^\top x_i^\sigma, G^\sigma \succ 0, \beta^{\sigma_j} > \beta_{LB}^\sigma)}{\sum_{s=1}^M \pi_s(x_i^\sigma, \xi^{\sigma_s})N(y_i^\sigma|(\beta_s^\sigma)^\top x_i^\sigma, G^\sigma \succ 0, \beta^{\sigma_j} > \beta_{LB}^\sigma)}. \quad (25)$$

Next in the maximization step—*M–step*—of the EM algorithm, we maximize the following objective function:

$$L_C(\beta^\sigma, \xi^\sigma|D_C^\sigma) = \sum_{i=1}^{K_\sigma}\sum_{j=1}^M \tau_{i,j}^\sigma \log\Big[\pi_j(x_i^\sigma, \xi^{\sigma_j}) \cdot N(y_i^\sigma|(\beta^{\sigma_j})^\top x_i^\sigma, G^\sigma \succ 0, \beta^{\sigma_j} > \beta_{LB}^\sigma)\Big]$$
$$= \sum_{i=1}^{K_\sigma}\sum_{j=1}^M \tau_{i,j}^\sigma \log\Big[N(y_i^\sigma|(\beta^{\sigma_j})^\top x_i^\sigma, G^\sigma \succ \beta_{LB}^\sigma)\Big] + \sum_{i=1}^{K_\sigma}\sum_{j=1}^M \tau_{i,j}^\sigma \log[\pi_j(x_i^\sigma, \xi^{\sigma_j})] = Q_{\beta^\sigma, \xi^\sigma}. \quad (26)$$

The solution of the optimization problem at the *M–step* is given by solving the following completely decoupled optimization problems. First, we minimize the cost function

$$\hat{Q}_{\beta^\sigma} = \sum_{i=1}^{K_\sigma}\sum_{j=1}^M \tau_{i,j}^\sigma \log\Big[N(y_i^\sigma|(\beta^{\sigma_j})^\top x_i^\sigma, G^\sigma \succ 0, \beta^{\sigma_j} > \beta_{LB}^\sigma)\Big] \quad (27)$$

with respect to β^σ. The above cost function is solved using a weighted constrained Iteratively Reweighted Least Squares (IRLS) algorithm using data pairs $\{x_i^\sigma, y_i^\sigma\}$ with weights $\tau_{i,j}^\sigma$.

Then, we minimize the cost function

$$\hat{Q}_{\xi^\sigma} = \sum_{i=1}^{K_\sigma} \sum_{j=1}^{M} \tau_{i,j}^\sigma \log\left[\pi_j(x_i^\sigma, \xi^{\sigma_j})\right] \quad (28)$$

with respect to ξ^σ. The above cost function is optimized using an IRLS algorithm using data pairs $\{x_i^\sigma, \tau_{i,j}^\sigma\}$.

In summary, we propose Algorithm 1 for solving EM for the utility learning problem cast as a Mix-cFGLS problem. A key aspect to the solution is in selecting an appropriate noise structure for each player's data structure. We initially fit the data using cFGLS, initialize our algorithm and use the estimated $\hat{G}$ in the EM update steps. Since EM is a co-ordinate descent algorithm for the non-convex optimization problem (23), we run it several times and select the learners resulting from the highest log-likelihood L_N

Algorithm 1 EM-algorithm for Mix-cFGLS utility learning of player σ

1: **function** EM-MIX-CFGLS(X,Y,M,C_{limit})
2:     **Initialization:** Fit data with cFGLS learner using
3:     an appropriate Heteroskedasticity-noise structure G
4:     from (20) or (21)
5:     $\hat{G}_{EM} \leftarrow \hat{G}_{cFGLS}$ ▷ assignment of noise matrix
6:     $\theta_s^\sigma \leftarrow \theta_{cFGLS}$ for $s = 1, \cdots, M$ ▷ θ_s^σ initialization
7:     $\xi_s^\sigma \leftarrow 0$ for $s = 1, \cdots, M$ ▷ ξ_s^σ initialization
8:     $C \leftarrow C_{limit}$ ▷ convergence tolerance
9:     $k \leftarrow 1$ ▷ iteration number
10:    $M_{max} \leftarrow N$ ▷ upper iterations bound
11:    Compute initial log-likelihood value, L_I, using (24)
12:    **Main Program:**
13:    **while** $k < M_{max}$ **do**
14:       Update latent variables τ^σ using (25) ▷ *E–step*
15:       Update θ^σ solving (27)   ▷ *M–step*
16:       Update ξ^σ solving (28)   ▷ *M–step*
17:       Update log-likelihood, L_N, using (24)
18:       **if** $L_N - L_I < C$ **then**
19:          break
20:       **else**$L_I \leftarrow L_N$
21:       $k \leftarrow k + 1$
22:    **Outputs:** θ^σ, ξ^σ and L_N

C. Parametric Bootstrap: Balancing Bias vs. Variance

As in our past work [15], we again employ bootstrapping techniques to improve the results. This helps increase the size of our social game data set.

The technique we consider is *wild bootstrapping* since it is a technique of parametric bootstrapping that is consistent with heteroskedasticity inference and the cFGLS framework. Wild bootstrap in regression models is a powerful tool for reducing the overall variance [28]. Using wild bootstrapping we estimate the asymptotic approximation of bias and standard error of the cFGLS estimators [25], [28]. The wild bootstrapping data generation model assumes $E(Y|X) = X\beta$ but allows for heteroskedasticity—a noise structure using transformations of residuals resulting from the cFGLS fitting (see (20), (21)).

The data generation process under wild bootstraping is given by

$$Y^* = X\beta_{cFGLS} + \Phi(e)\epsilon^* \quad (29)$$

where $Y^* \in \mathbb{R}^{n_d \times 1}$ is the new observation-dependent vector (pseudo-vector), $\beta_{cFGLS} \in \mathbb{R}^{n_d \times 1}$ is the learner estimated using a cFGLS framework, $\epsilon^* \sim N(0, I^{n_d \times n_d})$, $e \in \mathbb{R}^{n_d \times 1}$ is the residuals vector—namely, the difference between observed and fitted cFGLS values and mathematically is given by $e = Y - X\beta_{cFGLS}$. In addition, $\Phi(e) = \hat{G}^{\frac{1}{2}} \in \mathbb{R}^{n_d \times n_d}$ is a non-linear transformation that maps from $\mathbb{R}^{n_d \times 1}$ to $\mathbb{R}^{n_d \times 1}$ using the estimated noise structure from (20) or (21).

The bootstrapping process can be described in two steps: First, we fit our cFGLS model and then we perturb our model by adding Gaussian noise to the predicted values of the cFGLS statistical model. The data generation process (29) creates N replicates of *pseudo–data* which gives us N fitted bootstrap cFGLS estimators. Using ensemble methods such as bagging, we combine the resulting N *weak* bootstrapped cFGLS estimators. Bagging works efficient with high variance models and does not hurt the overall performance of the statistical model. The bagged estimator is given by

$$\beta_{Bagged} = \frac{1}{N} \sum_{s=1}^{N} \beta_{cFGLS}^s \quad (30)$$

where β_{cFGLS}^s is the estimator using the s–th pseudo–data sample. We refer to the bagged estimates as *bagged mega-learners* since they combine a number of weak learners.

Bagging serves to reduce the estimator bias. In the mixture of utilities model, we considered two noise structures: (20) and (21). For both cases, bagging results in an estimator that has higher forecasting accuracy since it reduces the first term of the so called *Bias-Variance Tradeoff* which, for a process $Y = X\theta + \epsilon$, is seen in the Mean Square Error (MSE):

$$\text{MSE}(x) = E[(Y - \theta_{est}^\top x)^2]$$
$$= \underbrace{(E[\theta_{est}^\top x] - Y)^2}_{Bias} + \underbrace{E[(\theta_{est}^\top x - E[\theta_{est}^\top x])^2]}_{Variance} \quad (31)$$

In machine learning, you can enhance forecasting accuracy by allowing for a small amount of bias if it results in a large reduction in variance. This is widely used in Ridge regression and in Lasso [28] in a form of a *prior* knowledge. In the Mix-cFGLS framework, we are able to explore the tradeoff between minimizing bias and variance. Indeed, having captured the noise structure using heteroskedasticity inference we are able to reduced estimation bias. However, since the learners in Mix-cFGLS framework are not static—they depend on historical data—we allow an amount of *bias* in order to gain a substantial decrease the *variance*.

IV. UTILITY LEARNING RESULTS

In this section, we present the results of Mix-cFGLS utility learning applied to the data collected from the social game experiment we conducted. We show that the resulting forecast using the proposed method is more accurate than ensemble utility methods such as bagging, and classical methods such as OLS. For the Mix-cFGLS we use two mixture components, one aggressive θ_A and one defensive

ROOT MEAN SQUARE ERROR (RMSE), MEAN ABSOLUTE ERROR (MAE) AND
MEAN ABSOLUTE SCALED ERROR (MASE) [29] OF FORECASTING USING
MIX-cFGLS, BAGGED, AND cOLS UTILITY LEARNERS. FORECASTING PREDICTS
OCCUPANTS' BEHAVIOR FOR DEFAULT LIGHTING SETTINGS OF 20 AND 10.

Default 20	Mix-cFGLS	Bagged	cOLS
RMSE	**7.45**	8.31	22.53
MAE	**4.11**	5.20	18.35
MASE	**1.65**	2.08	7.34
Default 10	Mix-cFGLS	Bagged	cOLS
RMSE	**7.75**	8.17	17.63
MAE	**6.42**	7.01	14.24
MASE	**4.42**	4.82	9.13

θ_D, to represent each occupant. In particular, θ_A represents a player's profile that cares more about rewards, i.e. sacrifices comfort level for winning more points, while θ_D represents the opposite, i.e. covets comfort over winning.
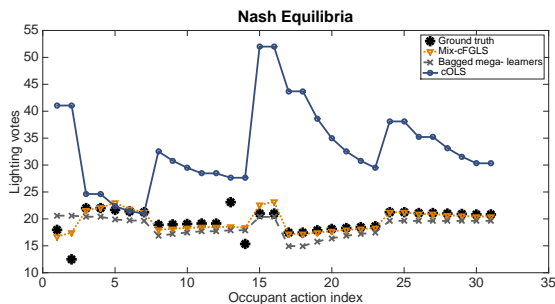

Fig. 1. The ground truth mean of the observed lighting votes for default lighting setting of 20 is depicted by the black dots. The forecasting results via simulation of the occupant game using the cOLS, bagged mega-learners, and Mix-cFGLS learners are indicated in blue, grey, and orange respectively. On the x–axis we indicate the index of when a choice was made by one or more of the occupants (i.e. when the implemented lighting setting is changed); the time from one index to the next may be several minutes to hours depending on the activity level of the occupants. Notice that the mean of the Nash equilibria of the simulated game using the bagged mega-learners and Mix-cFGLS learners is approximately near the true mean where the cOLS learners produce Nash equilibria with a large error. Mix-cFGLS learners have a nearly perfect forecast.

Using occupant voting data we simulate the game defined by the learned Mix-cFGLS utility functions and show that the estimated model significantly reduces prediction error as compared to classical OLS and bagging (see Table I). Our training data consisted of 80% of the users' actions in each default area. We test—i.e. compare the simulated forcast from our learned utility functions to the ground truth—on the remaining data for each default region[2].

In Figure 1, we see that the mixture of utilites model nearly approximates the ground truth and outperforms the other methods. In addition, using wild bootstrapping, we approximate the bias of the learners for the bagging and

[2]We remark that the prediction error as reported in Table I is different than in our earlier publication [15] due to the fact that we utilized a different proportion of the data for testing and training
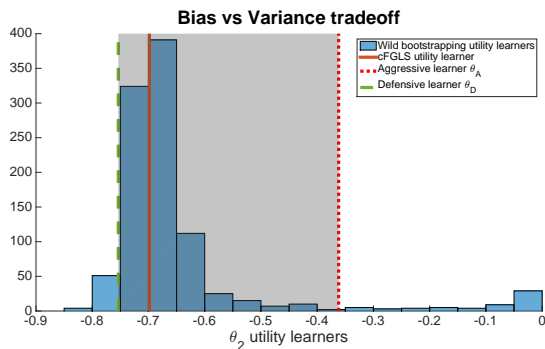

Fig. 2. The histogram depicts the learners using the wild bootstrapping technique. We have approximately a Gaussian distribution around the initial cFGLS estimator which depicts an unbiased cFGLS utility learner. The vertical orange line represents the cFGLS estimator. The grey shaded region represents the area between the aggressive (red dotted vertical line) and defensive (green dashed vertical line) utility learners—i.e. θ_A and θ_D, respectively—estimated using Mix-cFGLS framework.

TABLE II

cFGLS ESTIMATES, BAGGED MEGA-LEARNER ESTIMATES AND BIAS
APPROXIMATION USING WILD BOOTSTRAPPING. IN BOLD, WE DENOTE
THE OCCUPANTS WITH NEARLY UNBIASED ESTIMATORS. THESE ARE
THE OCCUPANTS WITH THE LARGEST VARIATION IN THEIR VOTES.

User id	cFGLS	Bagged	bias
2	-0.69	-0.58	**0.11**
6	0.5	1.62	1.12
8	298.06	121.2	-176.86
14	337.52	151.26	-186.26
20	-0.8	-0.73	**0.07**

cFGLS methods. We remark that the Mix-cFGLS method aims to increase the bias in exchange for a reduction in variance. In Figure 2, we show the histogram of the cFGLS learners for player with user–id 2 obtained by replicates of data using wild bootstrapping. This particular occupant represents a player that prefers *comfort* to winning the majority of the time. Notice his bagged cFGLS estimator is almost completely unbiased. The resulting utility learner Mix-cFGLS varies inside the grey region due to the fact that the softmax function gives a weighted sum of θ_A (red dotted vertical line) and θ_D (green dashed vertical line) learners (borders of the shaded region). This shows how the Mix-cFGLS introduces bias in the player's parameters estimates and, in exchange, we get more accurate forecasting of the users' actions.

Lastly, in Table II, we present the cFGLS estimates, the bagged mega-learner estimates, and the bias for the most active users. For some users, we see that the bias is not significantly reduced; this is likely due to the fact that some users are not nearly as active as others and hence, there was little variation in their data. In future work, we can design incentive mechanisms targeted at these individuals so that they provide *more informative* responses that will help us learn and more accurate model for their decision-making.

V. Discussion and Future Work

We presented a new framework of parametric utility learning using a probabilistic interpretation for combining multiple utility functions via Mix-cFGLS using a non-spherical noise model. Our framework allows for the estimated parameters of the learned utility functions to depend on the historical actions of the players which, in turn, captures the fact that players' utility functions are not static. Moreover, Mix-cFGLS enhances forecasting accuracy by allowing for a slight amount of bias in the utility learners in exchange for a reduction in variance of the MSE.

We applied the mixture of utilities method to learn the utility functions of participants in the social game that we conducted and showed that the forecasting error of the occupants' actions is significantly less than other methods such cOLS, cFGLS, and bagged mega-learners. The mixture of utilities framework can be applied broadly to many different applications.

There are several directions for future research. We are actively working on employing prior knowledge—e.g. survey results—into the estimation of players utility functions. This results in a penalized Mix-cFGLS model and has the potential to lead to better forecasting. In addition, we are exploring a graphical representation model for utility learning using Hidden Markov Models (HMM) or more advanced Contextual Hidden Markov Model (CHMM) with regression emissions. As for the social game experiment, we are in the process of implementing new building energy management social game experiments, both small- and large-scale in Singapore and on the Berkeley campus.

References

[1] J. McQuade, "A system approach to high performance buildings," *United Technologies Corporation, Tech. Rep*, 2009.

[2] A. Aswani, N. Master, J. Taneja, V. Smith, A. Krioukov, D. Culler, and C. Tomlin, "Identifying models of HVAC systems using semi-parametric regression," in *Proc. of the American Control Conf.*, 2012, pp. 3675 – 3680.

[3] M. Boman, P. Davidsson, N. Skarmeas, K. Clark, and R. Gustavsson, "Energy saving and added customer value in intelligent buildings," in *Third International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology*, 1998, pp. 505–517.

[4] D. Bourgeois, C. Reinhart, and I. Macdonald, "Adding advanced behavioural models in whole building energy simulation: A study on the total energy impact of manual and automated lighting control," *Energy and Buildings*, vol. 38, no. 7, pp. 814 – 823, 2006, special Issue on Daylighting Buildings.

[5] Y. Ma, G. Anderson, and F. Borrelli, "A distributed predictive control approach to building temperature regulation," in *Proc. of the American Control Conf.*, 2011, pp. 2089–2094.

[6] F. Oldewurtel, A. Parisio, C. Jones, M. Morari, D. Gyalistras, M. Gwerder, V. Stauch, B. Lehmann, and K. Wirth, "Energy efficient building climate control using stochastic model predictive control and weather predictions," in *Proc. of the American Control Conf.*, 2010, pp. 5100–5105.

[7] T. Lovett, E. Gabe-Thomas, S. Natarajan, E. O'Neill, and J. Padget, "'just enough' sensing to enliten: A preliminary demonstration of sensing strategy for the 'energy literacy through an intelligent home energy advisor' (enliten) project," in *Proc. of the Fourth Inter. Conf. on Future Energy Systems*. New York, NY, USA: ACM, 2013, pp. 279–280.

[8] J. Mathieu, M. Dyson, D. Callaway, and A. Rosenfeld, "Using residential electric loads for fast demand response: The potential resource and revenues, the costs, and policy recommendations," in *ACEEE Summer Study on Energy Efficiency in Buildings*, 2012.

[9] M. Roozbehani, M. Dahleh, and S. Mitter, "Dynamic pricing and stabilization of supply and demand in modern electric power grids," in *First IEEE International Conference on Smart Grid Communications*, oct. 2010, pp. 543 –548.

[10] M. Albadi and E. El-Saadany, "A summary of demand response in electricity markets," *Electric Power Systems Research*, vol. 78, no. 11, pp. 1989 – 1996, 2008.

[11] J. L. Mathieu, P. N. Price, S. Kiliccote, and M. A. Piette, "Quantifying changes in building electricity use, with application to demand response," *IEEE Transactions on Smart Grid,*, vol. 2, no. 3, pp. 507–518, 2011.

[12] M. Lee, O.Aslam, B. Foster, D. Kathan, J. Kwok, L. Medearis, R. Palmer, P. Sporborg, and M. Tita, "Assessment of demand response and advanced metering," Federal Energy Regulatory Commission, Tech. Rep., 2013.

[13] L. Ratliff, M. Jin, I. C. Konstantakopoulos, C. Spanos, and S. S. Sastry, "Social Game for Building Energy Efficiency: Incentive Design," in *Proc. of the 52nd Allerton Conf. on Communication, Control, and Computing*, 2014.

[14] I. C. Konstantakopoulos, L. J. Ratliff, M. Jin, S. S. Sastry, , and C. Spanos, "Social game for building energy efficiency: Utility learning, simulation, and analysis," arXiv, Tech. Rep., 2014.

[15] I. C. Konstantakopoulos, L. Ratliff, M. Jin, C. Spanos, and S. S. Sastry, "Smart building energy efficiency via social game: A robust utility learning framework for closing–the–loop," in *In the Proceedings of the 1st International Workshop on Science of Smart City Operations and Platforms Engineering under Cyber Physical Systems Week*, 2016.

[16] M. Jin, R. Jia, Z. Kang, I. C. Konstantakopoulos, and C. Spanos, "Presencesense: Zero-training algorithm for individual presence detection based on power monitoring," in *BuildSys'14, November 5–6, 2014, Memphis, TN, USA*. ACM, 2014, pp. 1–10.

[17] L. J. Ratliff, "Incentivizing efficiency in societal-scale cyber-physical systems," Ph.D. dissertation, University of California, Berkeley, 2015.

[18] G. Taguchi, E. A. Elsayed, and T. C. Hsiang, *Quality engineering in production systems*. McGraw-Hill College, 1989.

[19] J. B. Rosen, "Existence and uniqueness of equilibrium points for concave n-person games," *Econometrica*, vol. 33, no. 3, p. 520, 1965.

[20] L. J. Ratliff, S. A. Burden, and S. S. Sastry, "On the characterization of local nash equilibria in continuous games," *IEEE Transactions on Automatic Control*, 2016 (to appear).

[21] A. Keshavarz, Y. Wang, and S. Boyd, "Imputing a convex objective function," in *IEEE International Symposium on Intelligent Control*. IEEE, 2011, pp. 613–619.

[22] L. J. Ratliff, R. Dong, H. Ohlsson, and S. S. Sastry, "Incentive design and utility learning via energy disaggregation," in *Proc. of the 19th World Congress of the Inter. Federation of Automatic Control*, 2014.

[23] S. D. Flåm, "Solving non-cooperative games by continuous subgradient projection methods," in *System Modelling and Optimization*. Springer, 1990, pp. 115–123.

[24] M. Gendreau, "On the location of eigenvalues of off-diagonal constant matrices," *Linear Algebra and its Applications*, vol. 79, pp. 99 – 102, 1986.

[25] D. A. Freedman, *Statistical models: theory and practice*. cambridge university press, 2009.

[26] D. A. Freedman and S. C. Peters, "Bootstrapping a regression equation: Some empirical results," *Journal of the American Statistical Association*, vol. 79, no. 385, pp. 97–106, 1984.

[27] F. Cribari-Neto, "Asymptotic inference under heteroskedasticity of unknown form," *Computational Statistics & Data Analysis*, vol. 45, no. 2, pp. 215–233, 2004.

[28] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning, Data Mining, Interference, and Prediction*. Springer, 2009.

[29] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *International journal of forecasting*, vol. 22, no. 4, pp. 679–688, 2006.