Leveraging Correlations in Utility Learning

Ioannis C. Konstantakopoulos*, Lillian J. Ratliff*, Ming Jin, and Costas J. Spanos

***Abstract*— We present two approaches for leveraging correlations in learning the utilities of non-cooperative agents' competing in a game: correlation and coalition utility learning. In the former, we estimate the correlations between agents using constrained Feasible Generalized Least Squares with noise estimation and then use the estimated correlations to generate a correlation utility function for each agent which is a weighted sum of its own estimated utility function and all the agents' estimated utilities that are highly correlated with them. We then optimize the weights to boost the performance of the estimators. In the latter, we use a small amount of training data to estimate the correlations between players and form coalitions between agents that are positively correlated. We then estimate the parameters of the utility functions for each coalition where agents in a coalition jointly optimize their utilities. The correlation utility learning method outperforms existing schemes while the coalition utility learning method is simple enough to be adapted to an online framework after an initial training phase, yet it matches the performance of much more complex schemes. To demonstrate the efficacy of the estimation schemes, we apply them to data collected from a social game framework for incentivizing more efficient shared resource consumption in smart buildings.**

I. Introduction

Due to the increased use of Internet of Things technologies that enable integration of human decision-makers into the management and operations of everything from large scale infrastructure such as the smart grid or intelligent transportation systems to smaller components of smart cities such as smart buildings (see, e.g., [1]), we are in need of better learning and control algorithms that account for *humans in the loop* [2].

In this paper, we propose a new method of estimating utility functions for human decision-makers. Estimating decision-making models for human agents is typically aimed at either *description* or *prediction*. It is our aim to derive a *predictive model* that has the potential to be integrated with an incentive design scheme. Building on existing game theoretic concepts such as *coalition games* [3], we are able to extend our existing robust utility learning framework [4], [5] to a utility learning framework that has the potential to learn the interconnections between players' decision-making processes and leverage them to in improving forecasting algorithms. The coalition mechanism is simply used to support better prediction. In particular, we leverage classical estimation techniques to learn correlations between players and propose two utility learning methods that utilize the correlations to improve forecasting performance. The first method is the *correlation utility learning framework* in which we use estimated correlations to define a *correlation game* in which each player's utility function is converted into a *correlation utility*. This method is described in two steps: in the first step, we apply constrained Feasible Generalized Least Squares (cFGLS) with noise estimation (which is the core of our robust utility learning method [4], [5]) to estimate the correlations between players. In the second step, we construct correlation utilities by taking a weighted sum of each player's constrained Ordinary Least Squares (cOLS) estimated utility function and all other players' estimated utilities that are highly correlated with it. We optimize over the weights to improve the forecast of players' decisions.

The second method is the *coalition utility learning framework* in which subsets of players are modeled at colluding to improve their outcome. This method can also be described in two steps: in the first step, a small subset of data is used to estimate correlations between players—again using cFGLS with noise estimation—that are then used to define coalitions amongst the most positively correlated players. In the second step, we employ a simple cOLS estimation procedure to estimate the parameters of the coalition utilities. The second step of the proposed method, being based on cOLS, can be executed online and thus, can be integrated into an adaptive incentive design framework [6].

We apply both estimation schemes to data collected from a social game experiment to induce energy efficient shared resource consumption in smart buildings which are a fundamental component of *smart cities*. Their efficient design and operation enables flexibility—e.g., by automatically shifting or curtailing demand during peak hours—for sustainability. We show that the correlation–based utility learning methods outperform existing utility estimation schemes including ones based on cOLS and cFGLS. The correlation utility learning method, however, is no less computationally expensive than the cFGLS scheme with noise estimation. On the other hand, the coalition utility learning method outperforms cOLS and matches the performance of the more expensive cFGLS. The major benefit of the coalition framework is that we get approximately the same performance with a less computationally intensive framework and one that can be integrated with an adaptive incentive design algorithm due

*Authors contributed equally

I. Konstantakopoulos, M. Jin, and C. Spanos are with the Electrical Engineering and Computer Sciences Department, University of California, Berkeley, Berkeley, CA 94720. email: {ioanniskon, jinming, spanos, sastry}@eecs.berkeley.edu

L. Ratliff is with the Electrical Engineering Department, University of Washington, Seattle, WA 98195. email: ratliffl@uw.edu

This work is supported by the Republic of Singapores National Research Foundation through a grant to the Berkeley Education Alliance for Research in Singapore (BEARS) for the Singapore-Berkeley Building Efficiency and Sustainability in the Tropics (SinBerBEST) Program. Ioannis C. Konstantakopoulos is a Scholar of the Alexander S. Onassis Public Benefit Foundation.

to the fact that the second step in the method is simply cOLS.

The rest of the paper is organized as follows. In Section II, we describe the agent decision–making model in a game theoretic framework. We describe the cFGLS with noise estimation scheme along with both the correlation and coalition utility learning methods in Section III. We briefly describe the social game for smart building energy efficiency and discuss the results of applying the proposed estimation schemes in Section IV-B. We conclude with some discussion and remarks on future work in Section V.

II. Game–Theoretic Framework

A p-player game is described in terms of the strategy spaces and utility functions for each player. We denote by $\mathcal{I} = \{1, \ldots, p\}$ the index set for players. Let $E_i^{m_i}$ denote the Euclidean strategy space of dimension m_i for player i and $x_i \in E_i^{m_i}$ denote its strategy vector. Define $m = \sum_i m_i$ and denote by $E^m = E_1^{m_1} \times \cdots \times E_p^{m_p}$ the joint strategy space and $x = (x_1, \ldots, x_p)$ the joint strategy. Each player's strategy vector x_i is constrained to a convex set $S_i \subset E_i$. Let ℓ_i be the number of constraints on player i's problem and let $\ell = \sum_{i=1}^p \ell_i$. Denote by $S = S_1 \times \cdots \times S_p$ the constraint set which we can explicitly characterize in terms of mappings $h_i : E^{m_i} \to E^{\ell_i}$ where each component $h_{ij}(x)$, $j = 1, \ldots, \ell_i$ is a concave function of x_i: $S_i = \{x_i | h_i(x_i) \geq 0\}$. It is assumed that S_i is non-empty and bounded.

We model agents as *utility maximizers*—that is, the i–th player faces the optimization problem given by

$$\max_{x_i \in S_i} f_i(x_i, x_{-i}) \tag{1}$$

where $x_{-i} = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_p)$ is the marginal strategy vector for all players excluding player i.

In this framework, the agents are non-cooperative players in a continuous game with convex constraints.

When incentivized to do so—e.g., because they stand to increase their payoff—players form coalitions in which members of the coalition jointly optimize their utilities. That is, the set of players $\mathcal{I}$ is partitioned into subsets such that players in each subset collude.

Suppose the set of players $\mathcal{I}$ is partitioned into p_c coalitions. We will use the notation $\mathcal{C}_i$ as the index set for coalition i for each $i \in \{1, \ldots, p_c\}$. Then, players in coalition $\mathcal{C}_i$ seek to solve the optimization problem given by

$$\max_{x_{\mathcal{C}_i} \in S_{\mathcal{C}_i} \in} f_{\mathcal{C}_i}(x_{\mathcal{C}_i}, x_{-\mathcal{C}_i}) \tag{2}$$

where $f_{\mathcal{C}_i}(x_{\mathcal{C}_i}, x_{-\mathcal{C}_i}) = \sum_{j \in \mathcal{C}_i} f_j(x_{\mathcal{C}_i}, x_{-\mathcal{C}_i})$, $S_{\mathcal{C}_i} = \times_{j \in \mathcal{C}_i} S_j$, $x_{\mathcal{C}_i} = (x_j)_{j \in \mathcal{C}_i}$ and $x_{-\mathcal{C}_i} = (x_j)_{j \in \mathcal{I}/\mathcal{C}_i}$.

In our framework, we will assume that utilities are transferrable—that is players can losslessly transfer part of its utility to another player. As an example, players in a coalition may agree to divide the payoff equally or may agree to some alternative distribution of the payoff in a side contract which we leave unmodeled. In general, players are incentivized to participate in a coalition if the utility of participating is greater than if they played the game as a selfish individual.

We model their interaction using the Nash equilibrium concept:

Definition 1 (Nash Equilibrium): A point $x \in S$ is a *Nash equilibrium* for the coalition game $(f_1, \ldots, f_n)$ on S if for each $i \in \{1, \ldots, p_c\}$ $f_{\mathcal{C}_i}(x_{\mathcal{C}_i}, x_{-\mathcal{C}_i}) \geq f_{\mathcal{C}_i}(x'_{\mathcal{C}_i}, x_{-\mathcal{C}_i})$, $\forall\, x'_{\mathcal{C}_i} \in S_{\mathcal{C}_i}$.

If $p_c = p$ and each player is in its own coalition by itself, then the above definition reduces to the definition of a Nash equilibrium for the p–player non-cooperative game $(f_1, \ldots, f_n)$. It is well known that Nash equilibria exist for concave games [7, Theorem 1].

The definition can be relaxed as follows:

Definition 2 (ε–Approximate Nash Equilibrium): Given $\varepsilon > 0$, a point $x \in S$ is a ε–approximate *Nash equilibrium* for the coalition game $(f_1, \ldots, f_n)$ on S if for each $i \in \{1, \ldots, p_c\}$ $f_{\mathcal{C}_i}(x_{\mathcal{C}_i}, x_{-\mathcal{C}_i}) \geq f_{\mathcal{C}_i}(x'_{\mathcal{C}_i}, x_{-\mathcal{C}_i}) - \varepsilon$, $\forall\, x'_{\mathcal{C}_i} \in S_{\mathcal{C}_i}$.

III. Utility Learning Framework

In this section, we describe the robust, correlation, and coalition utility learning frameworks. The robust utility learning framework describes the first step of both the correlation and coalition utility learning methods where correlations between players are approximated.

Let i–th player's utility function be parameterized as follows:

$$f_i(x_i, x_{-i}; \theta_i) = \varphi_{i,0}(x_i, x_{-i}) + \textstyle\sum_{j=1}^{N_i} \varphi_{i,j}(x_i, x_{-i})\theta_{ij} \tag{3}$$

where $\{\varphi_{i,j}\}_{j=1}^{N_i}$ is a set of non-constant, concave basis functions, $\theta_i = [\theta_{i1} \ \cdots \theta_{iN_i}]^\top$, and we use the notation $f_i(x_i, x_{-i}; \theta_i)$ to indicate the parameterization by θ_i. The parameters θ_i are assumed unknown and these are the parameters we seek to learn.

A. Utility Estimation Under Non-Spherical Noise

Let n_i denote the number of data points for player i and define $n_d = \sum_{i=1}^p n_i$ be the total number of data points. 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. We define residual functions capturing the amount of suboptimality of the observations $x_i^{(k)}$ [8], [9]. Indeed, let the residual of the stationarity and complementary conditions for player i's optimization problem be given by

$$r_{\mathrm{s},i}^{(k)}(\theta_i, \mu_i) = D_i f_i(x^{(k)}) + \textstyle\sum_{j=1}^{\ell_i} \mu_{ij} D_i h_{i,j}(x_i^{(k)}) \tag{4}$$

and $r_{\mathrm{c},i}^{j,(k)}(\mu) = \mu_{i,j} h_{i,j}(x_i^{(k)})$, $j \in \{1, \ldots, \ell_i\}$, respectively where $\mu_i = (\mu_{ij})_{j=1}^{\ell_i}$ are Lagrange multipliers. Define $r_{\mathrm{s}}^{(k)}(\theta) = [r_{\mathrm{s},1}^{(k)}(\theta_1, \mu_1) \ \cdots \ r_{\mathrm{s},p}^{(k)}(\theta_p, \mu_p)]^\top$ and $r_{\mathrm{c}}^{(k)} = [r_{\mathrm{c},1}^{(k)}(\mu_1) \ \cdots \ r_{\mathrm{c},p}^{(k)}(\mu_p)]^\top$ where $r_{\mathrm{c},i}^{(k)}(\mu_i) = [r_{\mathrm{c},i}^{1,(k)}(\mu_i) \ \ldots \ r_{\mathrm{c},i}^{\ell_i,(k)}(\mu_i)]$.

Given the observations of the agents' decisions, we solve the following convex optimization problem:

$$\begin{aligned} &\min_{\mu,\theta} \textstyle\sum_{k=1}^{n_d} \chi(r_s^{(k)}(\theta, \mu), r_c^{(k)}(\mu)) \\ &\text{s.t.} \ \ \theta_i \in \Theta_i, \mu_i \geq 0 \ \ \forall\, i \in \mathcal{I} \end{aligned} \tag{P–1}$$

where $\chi : \mathbb{R}^p \times \mathbb{R}^{p\cdot\ell} \to \mathbb{R}_+$ 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 $\mathbb{R}^p \times \mathbb{R}^{p\cdot\ell}$), the inequality $\mu_i \geq 0$ is element-wise and the Θ_i's are constraint sets for the parameters θ_i that collect prior information about the utility functions f_i. As an example, if $\{\varphi_{i,j}\}_{j=1}^N$ are all concave, then $\Theta_i = \mathbb{R}_+^N$ ensures that f_i is concave. For learning utilities in a game theoretic context, we would like to ensure that the observations are ε–approximate Nash equilibria for the estimated game and to do that we select Θ_i such that each player's parameterized utility function is concave. As indicated in [8], it is important to select each Θ_i such that it encodes enough prior information about each f_i so as to prevent trivial solutions; we ensure this by selecting the set of basis functions $\{\varphi_{i,j}\}_{j=1}^N$ for each player to be non-constant, concave functions and assuming $\varphi_{i,0} \not\equiv 0$ in our parameterization. We describe how we construct the Θ_i's for the social game for smart buildings in detail in Section IV.

Now, we convert (P–1) to a standard estimation framework. Let first us define the regressor design matrix $X = \text{diag}(X_1 \cdots X_n)$ by letting $X_i = [(X_i^{(1)})^\top \ \cdots \ (X_i^{(n_i)})^\top]^\top$,

$$X_i^{(k)} = \begin{bmatrix} Dh_i(x_i^{(k)}) & D\varphi_i(x^{(k)}) \\ \text{diag}(h_i(x_i^{(k)})) & 0_{\ell_i \times N_i} \end{bmatrix} \tag{5}$$

where $Dh_i(x_i^{(k)}) = [D_i h_{i,1}(x_i^{(k)}) \ \cdots \ D_i h_{i,\ell_i}(x_i^{(k)})] \in \mathbb{R}^{1\times \ell_i}$, $D\varphi_i(x^{(k)}) = [D_i\varphi_{i,1}(x^{(k)}) \ \cdots \ D_i\varphi_{i,N_i}(x^{(k)})] \in \mathbb{R}^{1\times N}$, and $\text{diag}(h_i(x_i^{(k)})) \in \mathbb{R}^{\ell_i \times \ell_i}$ is a diagonal matrix with entries $h_{i,j}(x_i^{(k)})$ for $j \in \{1, \ldots, \ell_i\}$ along the diagonal. We also define the observation-dependent vector $Y = [Y_1 \cdots Y_p]^\top$ where

$$Y_i = [-D_i\varphi_{i,0}(x^{(1)}) \ 0_{1\times\ell_i} \ \cdots \ -D_i\varphi_{i,0}(x^{(n_i)}) \ 0_{1\times\ell_i}]^\top$$

and the regression coefficient $\beta = [\mu_1 \ \theta_1 \ \cdots \ \mu_p \ \theta_p]^\top$.

Using the ℓ_2–norm on $\mathbb{R}^p \times \mathbb{R}^{p\cdot\ell}$ for χ in (P–1) leads to a cOLS problem:

$$\min_\beta \left\{ \|Y - X\beta\|_2 \middle| \ \beta \in \mathcal{B} \right\} \tag{P–2}$$

where $\mathcal{B}$ is a set that aggregates the constraints on the θ_i's and $\mu_{i,j}$'s—that is,

$$\mathcal{B} = \{\beta | \ \mu_{i,j} \geq 0, j \in \{1, \ldots, \ell_i\}, \theta_i \in \Theta_i, i \in \mathcal{I}\}.$$

Suppose that the prior information encoded in the Θ_i's can be expressed by inequality constraints on the entries of θ_i (which will be the case for our social game example), then we can express the constraint set as $\mathcal{B} = \{\beta | \beta > \bar{\beta}\}$ for some $\bar{\beta}$. Our data generation process, as described by problem (P–2), is a classical multiple linear regression as follows

$$Y = X\beta + \epsilon, \ \beta \in \mathcal{B} \tag{6}$$

where $\epsilon = (\epsilon_1, \ldots, \epsilon_p)$ is a spherical error term following: $E(\epsilon|X) = 0^{n_o \times 1}$ and $\text{cov}(\epsilon|X) = \sigma^2 I^{n_o \times n_o}$ where $n_o = \sum_{i=1}^p (\ell_i + 1)n_i$ is the total number of observations such that $Y \in \mathbb{R}^{n_o}$.

The data generation process (6) lacks robustness in the presence of non-spherical noise, results in biased parameter estimates—ones that do not satisfy the Gauss–Markov theorem for Best Linear Unbiased Estimator (BLUE) [10]—and performs poorly in forecasting. Robustness can be ensured by assuming heteroskedasticity [10, Chapter 5] which also allows for inference of correlated errors in the resulting regression model. These correlated errors can then be used to determine the relationship between players' decision-making processes. Thus, in our data generation model, we adopt a non-spherical standard error ϵ which is modeled by $\text{cov}(\epsilon|X) = G \succ 0, \ G \in \mathbb{R}^{n_o \times n_o}$. The model's standard error ϵ is drawn from multivariable normal probability distribution with zero mean and different variances and ϵ models autocorrelated events.

Moreover, by multiplying (6) on the left with $G^{-\frac{1}{2}}$, we can derive an unbiased estimator which satisfies the BLUE property. The resulting constrained Generalized Least Squares (cGLS) statistical model is given by

$$(G^{-\frac{1}{2}}Y) = (G^{-\frac{1}{2}}X)\beta + (G^{-\frac{1}{2}}\epsilon), \ \beta \in \mathcal{B}. \tag{7}$$

In many applications, the explicit form of $\text{cov}(\epsilon|X) = G$ is unknown. However, we can infer the noise by imposing structural constraints on G. We consider a block diagonal noise structure [10, Chapter 5]. In previous works [4], [5], we have explored other noise structures. For the present work, we choose the block diagonal noise structure just to streamline and simplify the presentation.

We impose that $G = \text{blkdiag}(K_1, \cdots, K_p) \in \mathbb{R}^{n_o \times n_o}$ where $K_i = \text{blkdiag}(B_{i,1}, \ldots, B_{i,n_i}) \in \mathbb{R}^{(\ell_i+1)n_i \times (\ell_i+1)n_i}$ with each $B_{i,k} \in \mathbb{R}^{(\ell_i+1)\times(\ell_i+1)}$. The estimates $\hat{B}_{i,k}$ are constructed from the residuals of the estimator for β. This is a standard procedure which is outlined in [10, Chapter 5]; hence, we leave the details to the Appendix A.

We substitute the inferred noise, $\hat{G}$, into the cGLS statistical model (7) to get the one–step constrained Feasible GLS (cFGLS) estimators. We iterate between the estimation of $\hat{G}$ and β^{cFGLS} either until convergence or for a fixed number of iterations to prevent overfitting. To resolve this trade-off and find the optimal iteration size we adopt a simple cross-validation method.

B. Player Correlation Estimation

For both the methods we propose for utility learning, we need an estimate of the correlation between players. We use a Wild Bootstrapping [10], [11] process which is consistent with the selected heteroskedastic noise structure—it is also commonly used to artificially increase the size of the data set when the number of unknown parameters is large compared to the number of observations—which is the case for the social game application we present in Section IV as well as many other applications where human decision–makers are involved.

The bootstrapping process can be described in two steps: First, fit a cFGLS model, $\hat{\beta}^{\text{cFGLS}}$. Then, generate N_{boot} replicates of *pseudo–data* using the data generation process $\tilde{Y} = X\hat{\beta}^{\text{cFGLS}} + \Phi(e)\varepsilon$, where $\tilde{Y} \in \mathbb{R}^{n_o \times 1}$ is the new observation vector (pseudo-observations), $\hat{\beta}^{\text{cFGLS}} \in \mathbb{R}^{n_o \times 1}$ is the cFGLS estimator, $\varepsilon \sim \mathcal{N}(0, I^{n_o \times n_o})$, $e \in \mathbb{R}^{n_o \times 1}$ is the residual vector given by $e = \tilde{Y} - X\hat{\beta}^{\text{cFGLS}}$ and $\Phi : \mathbb{R}^{n_o \times 1} \to \mathbb{R}^{n_o \times 1}$ is a nonlinear transformation such that $\Phi(e) = \hat{G}^{\frac{1}{2}} \in \mathbb{R}^{n_o \times n_o}$. Since $E(\Phi(e)\varepsilon|X) = \Phi(e)E(\varepsilon|X) = \Phi(e)E(\varepsilon) = 0_{n_o \times n_o}$, using the data generation process in (8), we resample from i.i.d variables. Using Wild Bootstrapping, the empirical covariance matrix of $\hat{\beta}_j^{\text{cFGLS}}$ is an asymptotic approximation of the covariance matrix and is given by

$$\hat{C}_\beta = \frac{1}{N_{\text{boot}}} \sum_{j=1}^{N_{\text{boot}}} \left(\hat{\beta}_j^{\text{cFGLS}} - \hat{\beta}^{\text{ave}}\right)\left(\hat{\beta}_j^{\text{cFGLS}} - \hat{\beta}^{\text{ave}}\right)^\top \quad (8)$$

where $\hat{\beta}_s^{\text{cFGLS}}$ is the estimator using the j-th pseudo-data sample and $\hat{\beta}^{\text{ave}} = \frac{1}{N_{\text{boot}}} \sum_{s=1}^{N_{\text{boot}}} \hat{\beta}_j^{\text{cFGLS}}$. Asymptotic estimation of the empirical covariance matrix reveals hidden structures between players and is what we leverage both in the correlation and coalition utility learning procedures.

Now that we have described cFGLS with noise estimation, in the following two sub-sections we will descibe how we use the approximations of correlations to boost the performance of the forecast based on the estimated utilities.

C. Correlated Utility Learning

The empirically learned correlations are used to reduce the forecasting error by crafting a new correlated game in which we construct a *correlation utility* for each player by composing a weighted sum of the player's estimated utility and the estimated utilities of all the players that are highly correlated with it. We then optimize over the weights in order to further reduce the forecasting error.

When the correlations between players are positive, we create what we refer to as *psuedo-coalitions* since players are not *explicitly* agreeing to collude in the game but rather are doing so *implicitly*. The degree of *psuedo-coalition* is discovered by the robust utility learning process through estimating the empirical covariance of $\hat{\beta}^{\text{cFGLS}}$. On the other hand, when the correlations between players are negative, we find that these negative correlations can be used to take advantage of active players' richer data sets in predicting the behavior of players that less active or ones that have little variation in their data.

We refer to the learned utility—$\hat{f}_i$ for player i—from the robust utility learning framework as the *nominal utility* whose estimate is given by

$$\hat{f}_i(x_i, x_{-i}) = \varphi_{i,0}(x_i, x_{-i}) + \sum_{j=1}^{N_i} \hat{\theta}_{i,j}^{\text{cOLS}} \varphi_{i,j}(x_i, x_{-i}) \quad (9)$$

where $\hat{\theta}_i^{\text{cOLS}}$ is extracted from the cOLS estimated $\hat{\beta}_i^{\text{cOLS}} = (\hat{\mu}_i^{\text{cOLS}}, \hat{\theta}_i^{\text{cOLS}})$.

Using the correlations we learn when we estimate $\hat{\beta}^{\text{cFGLS}}$ as described above, we construct a new utility $\hat{g}_i$ by combining scaled versions of a subset (potentially all) of the other players' utilities that are correlated with player i. Let $\mathcal{Q}_i \subset \mathcal{I}$ denote the subset of players correlated with player i and let $\mathcal{K}_i \subset \mathcal{Q}_i$ be the set of players used in constructing $\hat{g}_i$. The correlated utility $\hat{g}_i$ for player i is given by

$$\hat{g}_i(x) = \sum_{l \in \mathcal{Q}_i} \left(\alpha_{i,l} c_{i,l} \varphi_{i,0}(x) + \sum_{j=1}^{N_j} \alpha_{i,l} \hat{\theta}_{i,j}^{\text{cOLS}} \varphi_{i,j}(x)\right) \quad (10)$$

where as usual $x = (x_i, x_{-i})$, $\alpha_{i,i}$ is the estimated variance of player i determined by the empirical covariance matrix, $\alpha_{i,l}$ is the covariance between the parameter estimates for player i and l also determined by the empirical covariance matrix, and c_{il} are scaling constants over which we optimize. We refer to the resulting game as an *approximated correlation game*[1].

Given the form of $\hat{g}_i$, our goal is to optimize the scaling constants c_{il} in order to reduce the forecasting error. We formulate a convex optimization problem using the first– and second–order conditions on each player's individual optimization problem where we assume that player i is now solving the problem given by

$$\max_{x_i \in S_i} \hat{g}_i(x_i, x_{-i}). \quad (11)$$

The convex optimization problem we solve is formulated in a similar fashion to the base utility learning problem of Section III-A. Let $c_i \in \mathbb{R}^{|\mathcal{K}_i|}$ be defined as $c_i = (c_{i,j})_{j \in \mathcal{K}_i}$ and let $c = (c_i)_{i \in \mathcal{I}}$. Let the residual of the stationarity condition of (11) be given by

$$r_{\text{s},i}^{(k)}(z_i, \mu_i; \hat{\theta}_i^{\text{cOLS}}) = D_i \hat{g}_i(x^{(k)}) + \mu_i^\top D_i h_i(x_i^{(k)}) \quad (12)$$

and the residual of the complementary slackness conditions be given by

$$r_{\text{c},i}^{j,(k)}(\mu_i) = \mu_{i,j} h_{i,j}(x_i^{(k)}), \ j \in \{1, \ldots, \ell_i\}. \quad (13)$$

As before, let $r_{\text{c},i}^{(k)}(\mu_i) = [r_{\text{c},i}^{1,(k)}(\mu_i) \ \cdots \ r_{\text{c},i}^{\ell_i,(k)}(\mu_i)]$.

Define $Q_i \in \mathbb{R}^{n_i \times |\mathcal{K}_i|}$ by

$$Q_i = \left[\alpha_{i,j} D_{i,i}^2 \varphi_{i,0}(x^{(k)})\right]_{k=1, j \in \mathcal{K}_i}^{n_i}. \quad (14)$$

and $q_i \in \mathbb{R}^{n_i}$ by

$$q_i = \left[\sum_{j \in \mathcal{K}_j} \alpha_{i,j} \left(\sum_{l=1}^{N_i} \hat{\theta}_{il}^{\text{cOLS}} D_{i,i}^2 \varphi_{i,l}(x^{(k)})\right)\right]_{k=1}^{n_i}. \quad (15)$$

Then, we have the following convex optimization problem:

$$\begin{aligned} \min_{c,\mu} \ & \sum_{i=1}^{p} \sum_{k=1}^{n_i} \chi(r_{\text{s},i}^{(k)}(z_i, \mu_i; \hat{\theta}^{\text{cOLS}}), r_{\text{c},i}^{(k)}(\mu_i)) \\ \text{s.t.} \ & Q_i z_i + q_i \le 0, \ \mu_i \ge 0 \ \ \forall \ i \in \mathcal{I} \end{aligned} \quad \text{(P–2)}$$

Solving (P–2) gives us estimated correlated utilities $\hat{g}_i$ for each $i \in \mathcal{I}$ that we then use to forecast the players' decisions.

D. Coalition Utility Learning

As an alternative approach, when players are highly correlated we can re-estimate their parameters by returning to the utility learning procedure but now with players who are highly correlated treated as if they are participating in a coalition. Using the empirically learned correlations, we partition the set of players $\mathcal{I}$ into p_c coalitions. Analogous to the correlation utility learning method, our aim is to define a *coalition utility* $\tilde{g}_{\mathcal{C}_i}$ and estimate the parameters of the

[1]We remark that there exists an equilibrium concept called *correlated equilibrium* [12] which generalizes a Nash equilibrium by characterizing correlations between randomized strategies; we mention this only to alleviate any potential confusion. The equilibrium concept we utilize for the approximated correlation game is still a pure Nash equilibrium.

coalition utilities assuming coalitions play a game against each other, where those in a coalition jointly optimize their utilities.

Let $-\mathcal{C}_i = \mathcal{I}/\mathcal{C}_i$ be the set of players not in coalition $\mathcal{C}_i$. The coalition utility $\tilde{g}_{\mathcal{C}_i}$ for player i is given by

$$\tilde{g}_{\mathcal{C}_i}(x_{\mathcal{C}_i}, x_{-\mathcal{C}_i}) = \textstyle\sum_{j\in\mathcal{C}_i} f_j(x_{\mathcal{C}_i}, x_{-\mathcal{C}_i}) \tag{16}$$

where the nominal utility function used for the coalition game for player i is given by

$$f_i(x) = \textstyle\sum_{j\in\mathcal{C}_i}\varphi_{j,0}(x) + \sum_{l=1}^{N_j}\varphi_{j,l}(x)\theta_{j,l} \tag{17}$$

with $x = (x_{\mathcal{C}_i}, x_{-\mathcal{C}_i})$. Then players in $\mathcal{C}_i$ are jointly solving

$$\max_{x_{\mathcal{C}_i}\in S_{\mathcal{C}_i}} \tilde{g}_{\mathcal{C}_i}(x_{\mathcal{C}_i}, x_{-\mathcal{C}_i}). \tag{18}$$

Given $\tilde{g}_{\mathcal{C}_i}$, we develop a convex optimization problem to estimate parameters θ_i in order to reduce the forecasting error. Again, the problem is formulated in a similar fashion to the base utility learning problem of Section III-A.

Define the vector $\theta_{\mathcal{C}_i} \in \mathbb{R}^{M_i}$ where $M_i = \sum_{j\in\mathcal{C}_i} N_j$ by $\theta_{\mathcal{C}_i} = \left(\theta_1, \theta_2, \ldots, \theta_{|\mathcal{C}_i|}\right)$. For optimization problem (18), let the residual of the stationarity condition be given by

$$r_{\mathrm{s},\mathcal{C}_i}^{(k)}(\theta_{\mathcal{C}_i}, \mu_i) = D_i\tilde{g}_i(x_{\mathcal{C}_i}^{(k)}, x_{-\mathcal{C}_i}^{(k)}) + \textstyle\sum_{j=1}^{|\mathcal{C}_i|}\mu_j^\top D_j h_j(x_j^{(k)}) \tag{19}$$

and the residual of the complementary slackness conditions be given by

$$r_{\mathrm{c},l}^{j,(k)}(\mu_l) = \mu_{l,j}h_{l,j}(x_l^{(k)}),\ j\in\{1,\cdots,\ell_l\},\ l\in\mathcal{C}_i. \tag{20}$$

with $r_{\mathrm{c},l}^{(k)}(\mu_l) = [r_{\mathrm{c},l}^{1,(k)}(\mu_l)\ \cdots\ r_{\mathrm{c},l}^{\ell_l,(k)}(\mu_l)]^\top$, $r_{\mathrm{c},\mathcal{C}_i}^{(k)}(\mu_{\mathcal{C}_i}) = [r_{\mathrm{c},l}^{(k)}(\mu_l)]_{l\in\mathcal{C}_i}^\top$ and $\mu_{\mathcal{C}_i} = (\mu_l)_{l\in\mathcal{C}_i}$. Let $n_{\mathcal{C}_i} = \sum_{j\in\mathcal{C}_i} n_j$.

To estimate the coalition utilities, we solve

$$\begin{aligned}&\min_{\theta,\mu} \textstyle\sum_{i=1}^{N_c}\sum_{k=1}^{n_{\mathcal{C}_i}}\chi(r_{\mathrm{s},\mathcal{C}_i}^{(k)}(\theta_{\mathcal{C}_i}, \mu_{\mathcal{C}_i}), r_{\mathrm{c},\mathcal{C}_i}^{(k)}(\mu_{\mathcal{C}_i}))\\ &\text{s.t.}\ \ \theta_j\in\Theta_j,\ \mu_j\geq 0\ \ \forall\ j\in\mathcal{I}\end{aligned} \tag{P–3}$$

Solving (P–3) using the ℓ_2–norm for the convex penalty function χ gives us an ordinary least squares framework for estimating coalition utilities $\tilde{g}_{\mathcal{C}_i}$ for each $i\in\{1,\ldots,p_c\}$.

IV. Results for Smart Building Social Game

In this section, we briefly describe a social game experiment to encourage energy efficient shared resource consumption in smart buildings[2]. We also present the results of the two correlation–based utility learning methods to the data collected from this experiment.

A. Description of the Social Game Experiment and Model

Let us briefly describe the experimental setup, the participant decision–making model using the notation of the previous sections, and our previous work using the data from this experiment.

[2]We have several additional papers which flesh out the details of the experiment [4], [9], [13], and we refer the interested reader to those manuscripts.

1) Experimental Setup: 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[3]), which enables its users to adjust the lighting through a web portal. Using this web portal, the social game consists of participants (or users) who select a lighting setting by balancing their preferences on comfort, productivity, and desire to win a prize. The portal also allows users to visualize the social game—in particular, the dim level of the lights—as well as view the point levels and historical votes of all users.

The game is designed to leverage interactions amongst users, who win points based on how energy efficient their vote is compared to others. The users select their desired lighting dim level in the continuous interval $[0, 100]$ (0 being off, and 100 being the maximum level of lighting). There is a default lighting setting and when the users log in, they can choose a different setting or leave it at this default lighting level. Experiments with four default lighting levels, i.e. $\{10, 20, 60, 90\}$ where the numbers are the percentage of maximum lighting, were conducted, covering a spectrum of lighting conditions. The users can vote as frequently as they like and the average of all the users' current votes sets the implemented lighting dim level in the collaboratory. To enforce the rule that those who are not present cannot participate, we executed a simple presence detection algorithm based on their power usage [14].

2) Participant Decision–Making Model: The participants in the social game are the players whose utility functions we seek to estimate. There are 20 participants in the social game.

We model the participants as having utility functions composed of two terms that capture the tradeoff between lighting satisfaction and desire to win. An participant's lighting satisfaction level is modeled using a Taguchi loss function [15] which is interpreted as modeling satisfaction in such a way that it is decreasing as variation decreases around their selected lighting setting:

$$\varphi_{i,1}(x_i, x_{-i}) = -\left(\bar{x} - x_i\right)^2 \tag{21}$$

where $x = (x_1, \ldots, x_n)$ is the collection of lighting votes and $\bar{x} = \frac{1}{n}\sum_{i=1}^n x_i$ is its average and is the implemented lighting setting in the space. On the other hand, we model the participant's desire to win as

$$\varphi_{i,0}(x_i, x_{-i}) = -\rho z\left(x_i\right)^2 \tag{22}$$

where ρ is the total number of points distributed by the building manager and z is a scaling factor that is used primarily to scale the two terms of the utility function given that we inflate the points offered to make the points appeal

[3]`http://www.lutron.com`

to players and thus induce greater participation[4,5].

The points are distributed by the building manager using the relationship

$$\rho \frac{x_b - x_i}{n x_b - \sum_{j=1}^{n} x_j} \tag{23}$$

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.

Using the notation of the previous sections, each player's utility function is given by

$$f_i(x_i, x_{-i}) = \varphi_{i,0}(x_i, x_{-i}) + \theta_i \varphi_{i,1}(x_i, x_{-i}) \tag{24}$$

where θ_i is an unknown parameter that captures the tradeoff between winning and lighting satisfaction.

Each player's decision x_i is constrained to lie in the interval $S_i = [0, 100]$ where 0 represents the lights in the off setting and 100 represents the lights in the maximum on setting. Let $h_{i,j}(x_i, x_{-i})$ for $j \in \{1, 2\}$ denote the constraints on player i's optimization problem. For player i, the constraints are described by $h_{i,1}(x_i) = 100 - x_i$ and $h_{i,2}(x_i) = x_i$. Let $S_i = \{x_i \in \mathbb{R} | \; h_{i,j}(x_i) \geq 0, \; j \in \{1, 2\}\}$ and $S = S_1 \times \cdots \times S_n$. Let $\Theta_i = \{\theta_i \in \mathbb{R} | \; \theta_i \geq \theta_{\text{LB}}\}$ where θ_{LB} is determined by finding the least value of θ_i for which the game is concave (and thus, a Nash equilibrium exists [7, Theorem 1]) and the Nash equilibrium is isolated[6].

3) Previous Work: We have a number of previous works on the development of utility learning algorithms using this data set. In [9], we modeled the interaction between the building manager and the participants as a leader–follower(s) game where the building manager is the leader and the participants are the followers. We designed incentives by estimating the parameters using a cOLS framework and optimized the points and default lighting value using the estimated utilities.

In a follow-up work [4], focusing on the estimation aspect of the above, we developed a robust utility learning scheme using cFGLS and ensemble methods such as bagging. In other work [13], we developed techniques for learning a mixture of utilities model for players which allows us to capture the fact that over time players may make decisions that are consistent with different utility maximization models due to endogenous (time-varying preferences, mood, etc.) and exogenous (weather, schedule, etc.) factors.

In our observations of players, we noticed very significant correlations between players decisions which perhaps indicates that they are potentially colluding or that at least by treating them as colluding through *psuedo-coalitions* in the estimation procedure, our predictive model will potentially be more accurate.

B. Results

The social game data set is composed of lighting votes participants made throughout the duration of the experiment. The time from one vote to the next may be several minutes to hours depending on the activity level of the participants.

For both methods, we select combinations of players in support of improving the estimators' performance by utilizing information learned from players with the most variation in their votes in order to improve the estimates of players who consistently vote the same value or have a limited participation record. In this way, we *boost* the performance of our utility learning scheme by transferring information providing by the voting record of the more active players to other players.

For the correlation utility learning method, we apply a 10–fold cross validation [11] proceedure with an 80%–20% training/testing data split in order to limit overfitting. Using the each of the training data sets, we estimate the correlations between players using the robust utility learning method described in Section III-A which gives us $\hat{C}_\beta$. In Table I, we show a subset of the estimated covariance matrix $\hat{C}_\beta$. Using these values, we construct a correlated game as described in Section III-C for which we estimate the parameters using the correlated utility learning method.

We construct a correlation game with the following *pseudo-coalitions*:

(i) $\mathcal{K}_2 = \{2, 6, 20\}$: player 2's utility function is modified by player 6's and player 20's where each of these players are what we call *passive players* (i.e. their votes tend to be strongly related to their satisfaction as opposed to increasing their chances of winning—see the **red** cells in Table I);

(ii) $\mathcal{K}_8 = \{8, 14\}$: player 8's utility function is modified by player 14's where player 8 and 14 are what we call *aggressive players* (i.e. their votes tend to be much lower indicating a greater desire to win points—see the **green** cells in Table I);

(iii) $\mathcal{K}_{14} = \{2, 8, 14\}$.: player 14's utility function is modified by player 8's and player 2's where player 14 is positively correlated with player 8 and negatively correlated with player 2—see the **blue** cells in Table I.

All other players' utilities in the correlated game remain unchanged; that is, they are taken to be $\hat{g}_i \equiv \hat{f}_i, \; i \in \mathcal{I}/\{2, 8, 14\}$. We use the cOLS estimated parameters $\hat{\theta}_i^{\text{cOLS}}$ to create the correlation game. Then we apply the correlation utility learning method to optimize the $c_{i,j}$'s.

On the other hand, for the coalition utility learning method, we again divide the data into training and testing data sets with an 80%–20% split (and use cross validation). However, we use a small subset of the training data (approximately 3–5% of the data which is roughly 2 days of the experiment) to approximate the correlations between players. With these correlations we select coalitions. Then we use cOLS to estimate the parameters of the utilities for the coalition game.

[4] Inflating the points is a process of *framing* [16]—that is, dependent on how the reward system is presented to players greatly impacts their participation. Framing is routinely used in rewards programs for credit cards among many other point-based programs. The scaling factor z in the winning function is an attempt to remove the framing effect from the estimation procedure.

[5] We also remark that on other work, we have explored different basis functions [9]; however, the above simple quadratic functions tend to have the best performance and can be easily interpreted.

[6] In our previous works [5], [13], we describe in detail how we find this lower bound and for the sake of space we refer the reader to that work.

TABLE I

ESTIMATED COVARIANCE MATRIX FOR THE MOST ACTIVE PLAYERS. THE COLORED COLUMN-ROW PAIRS INDICATE THE AGENTS USED TO CREATE THE CORRELATION GAME—I.E. THE COLUMN INDICATES THE AGENT WHOSE ESTIMATED PARAMETER IS USED TO MODIFY THE ROW AGENT'S UTILITY.

Id	**2**	**6**	**8**	**14**	**20**
2	0.04	**0.06**	-2.80	-5.19	**0.03**
6	0.06	7.84	-16.8	0.84	-0.02
8	-2.80	-16.8	6.4×10^4	$\mathbf{4.28\times10^4}$	-7.60
14	**-5.19**	0.84	$\mathbf{4.28\times10^4}$	8.84×10^4	-12.6
20	0.03	-0.02	-7.60	-12.6	0.07

The reason we use only a small subset of the data is that the cFGLS and noise estimation scheme described in Section III-A is computationally expensive, especially with a larger bootstrapped data set. Our ultimate goal is to have a utility estimation method with low forecasting error that is simple enough to be converted to an online estimation scheme so that it can be integrated into an adaptive incentive design algorithm [6]. By employing cOLS in the estimation step, we are able to partially meet this goal.

Using the social game data, we select the most correlated players which happen to be players 8 and 14—the estimated correlation between these players is several orders of magnitude greater than the correlation between any of the other players. Hence, we create the coalition game $\mathcal{G}_{\text{coal}} = \{\tilde{g}_{\mathcal{C}_1}, \tilde{g}_{\mathcal{C}_2}\}$ where $\mathcal{C}_1 = \{8, 14\}$ and $\mathcal{C}_i = \{i\}$ for $i \in \mathcal{I}/\mathcal{C}_1$. As we noted, players 8 and 14 are both aggressive players. However, player 8 has little variation in its voting record—the voting record contains mostly zero votes. One interpretation is that the correlation method is in some sense serving to reduce dimensionality since players are clustered together and *shared information* between them is leveraged to improve the forecast performance.

In Table II, we present the root mean square error (RMSE), mean average error (MAE), and mean absolute scaled error (MASE) for the cOLS and cFGLS estimators and the estimated correlated utilities $\{\hat{g}_i(\cdot;\{\hat{\theta}_j^{\text{cOLS}}\}_{j\in\mathcal{K}_i})\}_{i\in\mathcal{I}}$ and coalition utilities $\{\tilde{g}_{\mathcal{C}_i}(\cdot;\{\tilde{\theta}_j^{\text{coal}}\}_{j\in\mathcal{C}_i})\}_{i\in\mathcal{I}}$ (where the $\tilde{\theta}_j^{\text{coal}}$'s are re-estimated using cOLS as in (P–3)). In the lower plot of Figure 1, we show the forecast produced by the cOLS, cFGLS, correlated, and coalition utility learning methods. We see that the correlated and coalition estimation schemes reduce the estimation error when comparing to cOLS. Their performance is on par with cFGLS and the correlated estimation scheme even outperforms cFGLS.

In general, cOLS performs poorly when players are treated as selfish individuals (see the lower plot in Figure 1 and Table II)—this is in part due to the size and lack of variation of the votes. Yet, using the correlation estimation scheme the cOLS estimators performance is improved by optimizing the weights of the correlation utilities. The coalition estimation scheme also reduces the estimation as compared to cOLS. This is, again, due to the fact that the method allows for *information sharing* since data from one player is used in estimating the utility of another player via learned correlations.

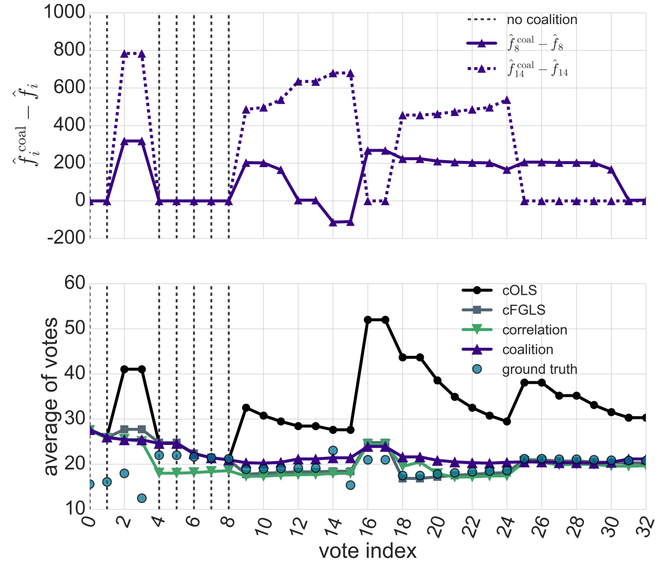

Fig. 1. Forecasting results for default lighting setting 20 (lower plot) and happiness metric comparing estimated utilities using the coalition $\hat{f}_i^{\text{coal}}$ and cOLS $\hat{f}_i$ utility learning methods (upper). The x–axis values indicate the index of when a choice was made by one or more 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 of the participants. The **dark gray** dashed lines indicate when no coalition was used in the coalition estimate (instead all players played selfishly—this occurs when player's 8 and 14 are not both present in the office and thus, cannot collude).

Moreover, in the upper plot of Figure 1, we show the difference between the coalition estimated utility and the cOLS estimated utility for players 8 and 14. In particular, using each observed approximate Nash $x^{(k)}$, we compute the cOLS estimated utility value

$$\hat{f}_i(x^{(k)}; \hat{\theta}_i^{\text{cOLS}}) \tag{25}$$

and the coalition estimated utility value which we take to be an equitable distribution of the payoff amongst the players in a coalition, i.e.

$$\hat{f}_i^{\text{coal}}(x^{(k)}; \hat{\theta}_i^{\text{coal}}) = \tfrac{1}{|\mathcal{C}_1|}\tilde{g}_{\mathcal{C}_1}(x^{(k)}; \hat{\theta}_{\mathcal{C}_1}^{\text{coal}}) \tag{26}$$

where $\hat{\theta}_{\mathcal{C}_1}^{\text{coal}} = (\hat{\theta}_i^{\text{coal}})_{i\in\mathcal{C}_1}$ and each $\hat{\theta}_i^{\text{coal}}$ are the cOLS estimated parameters for the coalition game. We then compute the *happiness metric* which we define to be the difference between these two estimated utilities:

$$\hat{f}_i^{\text{coal}}(x^{(k)}; \hat{\theta}_i^{\text{coal}}) - \hat{f}_i(x^{(k)}; \hat{\theta}_i^{\text{cOLS}}). \tag{27}$$

We remark that it is difficult to estimate the true structure of coalition side payments/utility transfers were they actually taking place. What is interesting, however, is that not only do we see improved estimator performance but also the happiness metric—which assumes a uniform distribution of wealth amongst coalitions—indicates that in fact the players have greater utility when treated as colluding. Specifically, looking at Figure 1, when there is no coalition (dashed grey lines), the utilities are equal which should be the case;

TABLE II

ROOT MEAN SQUARE ERROR (RMSE), MEAN ABSOLUTE ERROR (MAE) AND MEAN ABSOLUTE SCALED ERROR (MASE) FOR THE FORECAST USING THE cOLS, cFGLS, AND CORRELATION, AND COALITION UTILITY LEARNING METHODS IN THE DEFAULT LIGHTING SETTING 20.

Error	cOLS	cFGLS	correlated	coalition
RMSE	22.53	11.36	11.3	12.79
MAE	18.35	6.81	6.49	7.45
MASE	7.34	2.72	2.63	3.02

however, when we use a coalition for players 8 and 14, they are *better off* under the coalition utility with the exception of player 8 on two voting instances (votes indexed by 14 and 15). The fact that the players are generally happier under the coalition estimated utility than the cOLS estimated utility may indicate that there is some (explicit or implicit) collusion happening in practice. We are developing algorithms to approximate the Shapley value [17] which is a measure of a how important each player is to the coalition. Our approach will approximate the distribution of total surplus among players generated by collusion.

V. DISCUSSION

We presented two novel utility learning schemes that leverage estimated correlations between players in order to boost the performance of the estimated utilities in forecasting player decisions. Both methods outperform existing techniques based on classical estimation methods. Moreover, the coalition utility learning method is significantly less computationally intensive than cFGLS. After an initial batch training phase to compute correlations, it is amenable to online implementation and thus, has the potential to be integrated into an online algorithm for utility learning and incentive design [6].

We remark that the incentive mechanisms will ultimately modify players' utilities and thus, whether or not they are incentivized to collude, a central planner (such as a building manager or, more broadly, a service provider). This exposes an interesting avenue for future research in investigating the persistence of equilibria or coalitions after introduction of incentives. We are currently conducting experiments with an online version of the coalition utility learning method integrated with an adaptive incentive design scheme to collect data to support this work.

Moreover, as we mention at the end of the previous section, in order to aid in better incentive design, we seek algorithms for estimating the *power* a player has in a game via approximation of their Shapley value. Learning the players with the most bargaining power will help shape and target incentive mechanisms.

APPENDIX

A. Formation of the Noise Structure

Let $\hat{\beta}^{\text{cOLS}}$ be the cOLS estimate of β with residual vector $e = Y - X\hat{\beta}^{\text{cOLS}} \in \mathbb{R}^{n_o}$. The residual vector e can be decomposed into residuals for each player by writing $e = [e_1^\top \ \cdots \ e_p^\top]^\top$. We use e_i to compute an estimate $\hat{K}_i$ of K_i which is, in turn, used to compute $\hat{G}$. The residuals come in ℓ_i pairs since at each observation k, $Y_i^{(k)} \in \mathbb{R}^{\ell_i+1}$. There are n_i instances at which we have ℓ_i+1 observations. Let $(e_i)_{k,j} = (e_i)_{(\ell_i+1)(k-1)+j}$ where $k \in \{1,\ldots,n_i\}$ and $j \in \{1,\ldots,\ell_i+1\}$. Then, with the residuals, we form estimates $\hat{B}_{i,k} \in \mathbb{R}^{(\ell_i+1)\times(\ell_i+1)}$ of $B_{i,k}$ using $(\hat{B}_{i,k})_{jj} = n_i^{-1}\sum_{t=1}^{n_i} e_{t,j}^2$ and $(\hat{B}_{i,k})_{lj} = n_i^{-1}\sum_{t=1}^{n_i} e_{t,j}e_{t,\ell}$ for $j \neq \ell$ in

$$\hat{B}_{i,k} = \begin{bmatrix} (\hat{B}_{i,k})_{11} & \cdots & (\hat{B}_{i,k})_{1(\ell_i+1)} \\ \vdots & \ddots & \vdots \\ (\hat{B}_{i,k})_{(\ell_i+1)1} & \cdots & (\hat{B}_{i,k})_{(\ell_i+1)(\ell_i+1)} \end{bmatrix} \tag{28}$$

REFERENCES

[1] B. Morvaj, L. Lugaric, and S. Krajcar, "Demonstrating smart buildings and smart grid features in a smart energy city," in *Proc. 2011 3rd Intern. Youth Conf. on Energetics*, 2011, pp. 1–8.

[2] 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.

[3] M. J. Osborne and A. Rubinstein, *A Course in Game Theory*. MIT Press, 1994.

[4] 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 *Proc. 1st Intern. Workshop on Science of Smart City Operations and Platforms Engineering*, 2016.

[5] ——, "A robust utility learning framework," *Submitted to the IEEE Trans. Control Systems Technology*, 2016.

[6] L. J. Ratliff, "Incentivizing Efficiency in Societal-Scale Cyber-Physical Systems," Ph.D. dissertation, University of California, Berkeley, 2015.

[7] J. B. Rosen, "Existence and Uniqueness of Equilibrium Points for Concave N-Person Games," *Econometrica*, vol. 33, no. 3, pp. 520–534, 1965.

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

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

[10] D. A. Freedman, *Statistical models: theory and practice*. Cambridge University Press, 2009.

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

[12] R. J. Aumann, "Subjectivity and correlation in randomized strategies," *J. Mathematical Economics*, vol. 1, no. 1, pp. 67–96, 1974.

[13] I. C. Konstantakopoulos, L. Ratliff, M. Jin, C. Spanos, and S. S. Sastry, "Inverse modeling of non-cooperative agents via mixture of utilities," Proc. 55th IEEE Conf. on Decision and Control, Tech. Rep., 2016.

[14] 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.

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

[16] A. Tversky and D. Kahneman, "The framing of decisions and the psychology of choice," *Science*, vol. 211, no. 4481, pp. 453–458, 1981.

[17] L. S. Shapley, "A Value for n–person Games," in *Contributions to the Theory of Games*, ser. Annals of Mathematical Studies. Princeton University Press, 1953, pp. 307–317.