Scalable Tools for Learning Models of Strategic Decision-Making

Lillian J. Ratliff

joint with Ming Jin, Ioannis Konstantakopoulos, Costas Spanos, and S. Shankar Sastry

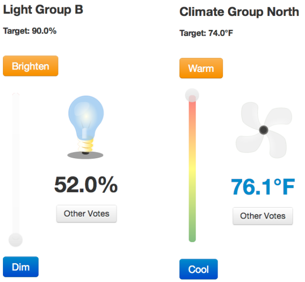
UC Berkeley

2015.05.28

FORCES
FOUNDATIONS OF RESILIENT CYBER-PHYSICAL SYSTEMS

Inducing Energy Efficient Usage of Shared Resources

- Social game for changing usage of shared resources.
- e.g. Lights, HVAC, etc.



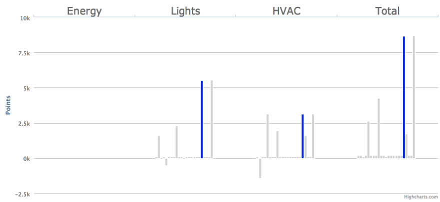
Social Game How will you save energy?

Comfort | Points | Energy Use | Energy Commitment | Game Rule | Winners

Summer 2014 - Week 12

		Quota	Commitment	Actual	Points	Bonus	Running Point Total
Energy	Aug. 15, 2014	130 Wh	Wh	1.24 Wh	0	5	5
Lights	Aug. 15, 2014	90.0%		0.5%	5,427	100	5,527
HVAC	Aug. 15, 2014	74.0°F		78.0°F	3,031	100	3,131
Grand Point Total							8,663

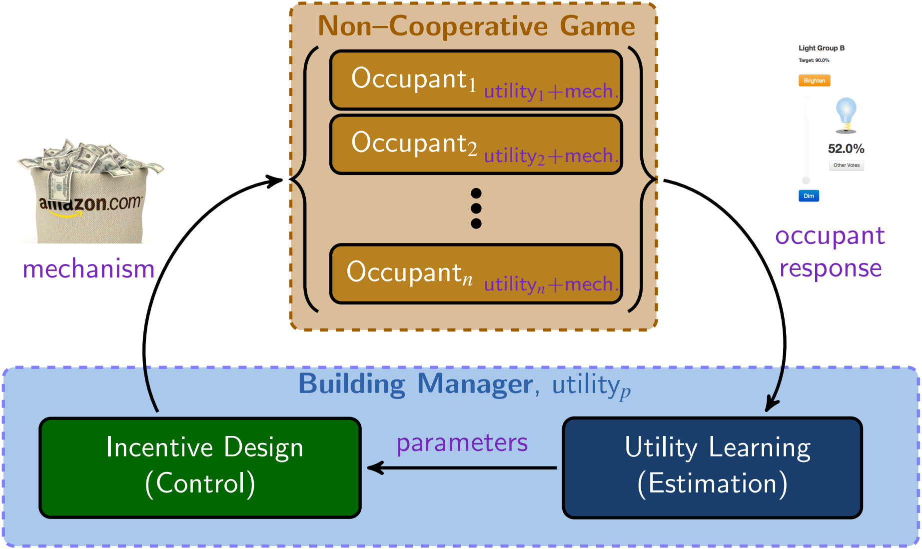
Update



Points are used to determine probability of winning in lottery

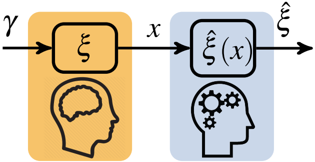
[1] L. Ratliff, *et al.*, Allerton, 2014.

Social Game Abstraction

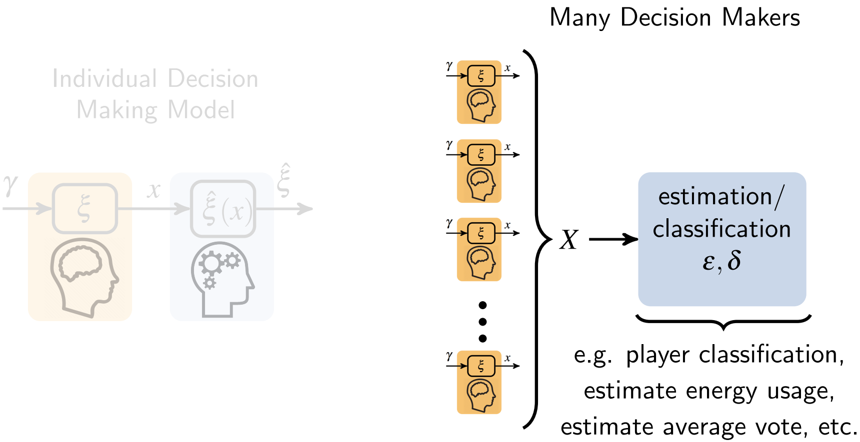


Scaling Up to Games with Many Players

Individual Decision Making Model

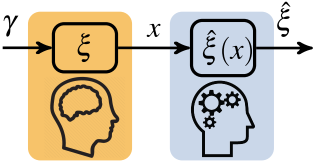


Scaling Up to Games with Many Players

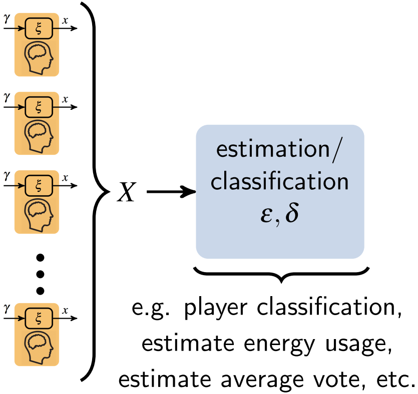


Scaling Up to Games with Many Players

Individual Decision Making Model



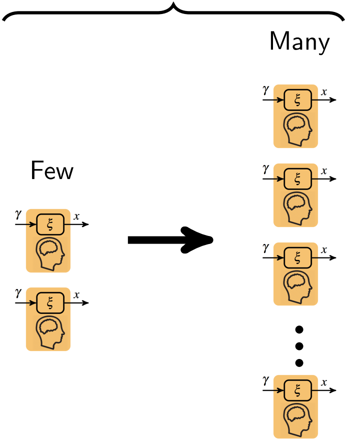
Many Decision Makers



Can we leverage the individual-level game-theoretic model of decision making in estimation/classification task?

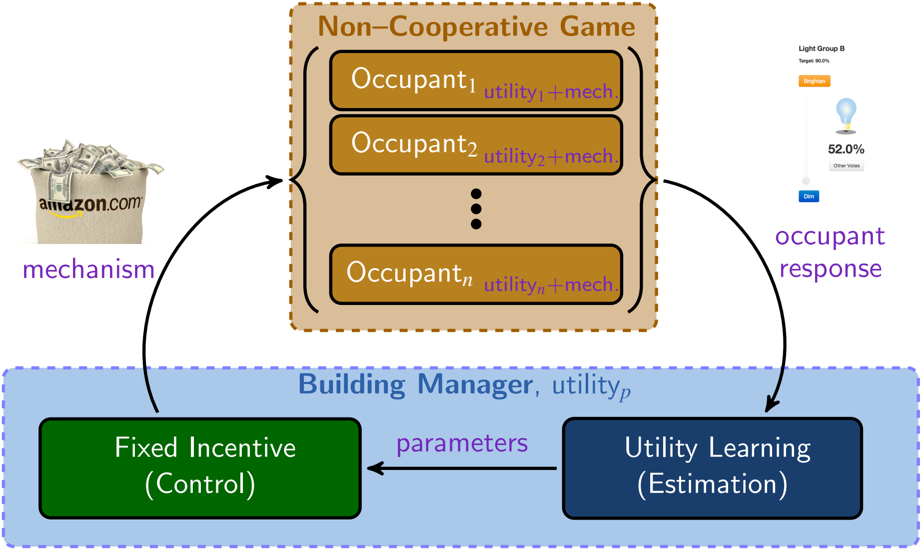
Today's Talk

1. Strategic Decision-Making



2. Bounds on Inference Error
3. Algorithm for Computing Stopping Time
4. Examples:
   - Behavior Parameter Inference
   - Energy Consumption Prediction
   - Classification of Players

Social Game for Inducing Efficient Usage of Shared Resources



Game–Theoretic Generative Behavioral Model

Each occupant selects

$$\underbrace{\text{vote}}_{x_i} \in [\underbrace{\text{min light setting}}_{0}, \underbrace{\text{max light setting}}_{100}]$$

in order to maximize

$$\underbrace{\text{occupant utility}}_{\Psi_i(x_i, x_{-i}, \gamma; \xi)} = \underbrace{\text{comfort}}_{\psi_0(x_i, x_{-i}, \gamma(x_i, x_{-i}))} + \xi_i \underbrace{\text{desire to win}}_{\psi_1(x_i, x_{-i}, \gamma(x_i, x_{-i}))}$$

ξ_i: tradeoff between comfort and desire to win, γ is the incentive

Definition (Nash Equilibrium)

A collection of lighting settings $x = (x_1, \ldots, x_n)$ is a **Nash equilibrium if no occupant can increase her utility by selecting a different lighting setting** x_i', i.e. for each $i \in \{1, \ldots, n\}$

$$\Psi_i(x_i, x_{-i}, \gamma(x_i, x_{-i})) \geq \Psi_i(x_i', x_{-i}, \gamma(x_i, x_{-i})) \quad \forall \quad x_i' \in [0, 100]$$

Game–Theoretic Generative Behavioral Model

Each occupant selects

$$\underbrace{\text{vote}}_{x_i} \in [\underbrace{\text{min light setting}}_{0}, \underbrace{\text{max light setting}}_{100}]$$

in order to maximize

$$\underbrace{\text{occupant utility}}_{\Psi_i(x_i,x_{-i},\gamma;\xi)} = \underbrace{\text{comfort}}_{\psi_0(x_i,x_{-i},\gamma(x_i,x_{-i}))} + \xi_i \underbrace{\text{desire to win}}_{\psi_1(x_i,x_{-i},\gamma(x_i,x_{-i}))}$$

ξ_i: tradeoff between comfort and desire to win, γ is the incentive

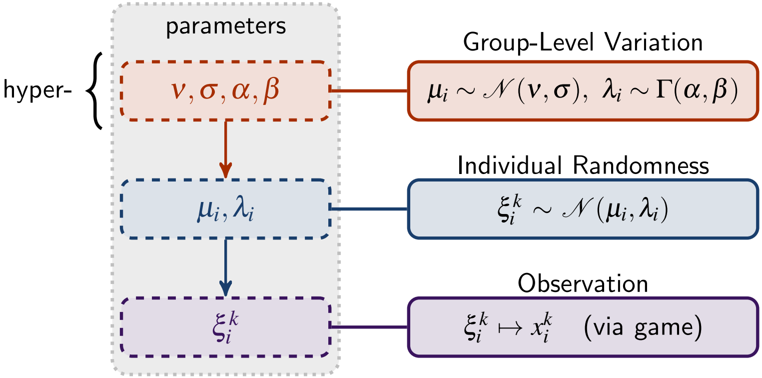
Definition (Nash Equilibrium)

A collection of lighting settings $x = (x_1, \ldots, x_n)$ is a **Nash equilibrium if no occupant can increase her utility by selecting a different lighting setting** x_i', i.e. for each $i \in \{1, \ldots, n\}$

$$\Psi_i(x_i, x_{-i}, \gamma(x_i, x_{-i})) \geq \Psi_i(x_i', x_{-i}, \gamma(x_i, x_{-i})) \quad \forall \quad x_i' \in [0, 100]$$

Learning to Learn Framework

Learning to learn: can improve estimation by simultaneously learning multiple similar tasks



$$\underbrace{\text{occupant utility}}_{\Psi_i(x_i, x_{-i}, \gamma; \xi)} = \underbrace{\text{comfort}}_{\psi_0(x_i, x_{-i}, \gamma(x_i, x_{-i}))} + \xi_i^k \underbrace{\text{desire to win}}_{\psi_1(x_i, x_{-i}, \gamma(x_i, x_{-i}))}$$

Bounds on Parameter Inference Error

Learning to Learn: we can provide bounds on parameter inference error.

Cramér–Rao bound for hyper–parameters $\theta = (\alpha, \beta, \nu, \sigma)$: Let X^t be all observations up to time t. For estimator $\hat{\theta}(X^t)$,

$$\underbrace{\mathbb{E}_{X^t}\left[(\theta_i - \hat{\theta}_i)^2\right]}_{\text{MSE of } \theta_i} \geq \frac{1}{n\zeta_i}, \text{ where } \zeta_i = \underbrace{-\mathbb{E}_{X^t}\left[\frac{\partial^2 \ln p(X^t|\theta)}{\partial \theta_i^2}\right]}_{\text{curvature}}$$

Take Away: lower bound decreases by order $1/n$ (number of users)

Bayesian Cramér–Rao Bound for $\theta_r = (\mu_i, \lambda_i)$, for any estimator $\hat{\theta}_r$,

$$\mathbb{E}_{\xi^t,\theta}\left[\left(\hat{\theta}(\xi^t) - \theta\right)\left(\hat{\theta}(\xi^t) - \theta\right)^T\right] \geq \begin{bmatrix} \frac{(\alpha-1)\beta\sigma}{T\sigma-(\alpha-1)\beta} & 0 \\ 0 & \frac{2(\alpha-1)(\alpha-2)\beta^2}{T+2\alpha-2} \end{bmatrix}$$

Take Away: decreases by order $1/T$ where T is number of samples

Hybrid Cramér–Rao bound applicable to the joint estimation of random and non–random parameters.

Bounds on Parameter Inference Error

Learning to Learn: we can provide bounds on parameter inference error.

Cramér–Rao bound for hyper–parameters $\theta = (\alpha, \beta, \nu, \sigma)$: Let $\boldsymbol{X}^t$ be all observations up to time t. For estimator $\hat{\theta}(\boldsymbol{X}^t)$,

$$\underbrace{\mathbb{E}_{\boldsymbol{X}^t}\left[(\theta_i - \hat{\theta}_i)^2\right]}_{\text{MSE of } \theta_i} \geq \frac{1}{n\zeta_i}, \text{ where } \zeta_i = \underbrace{-\mathbb{E}_{\boldsymbol{X}^t}\left[\frac{\partial^2 \ln p(\boldsymbol{X}^t|\theta)}{\partial \theta_i^2}\right]}_{\text{curvature}}$$

Take Away: lower bound decreases by order $1/n$ (number of users)

Bayesian Cramér–Rao Bound for $\theta_r = (\mu_i, \lambda_i)$, for any estimator $\hat{\theta}_r$,

$$\mathbb{E}_{\xi^t,\theta}\left[\left(\hat{\theta}(\xi^t) - \theta\right)\left(\hat{\theta}(\xi^t) - \theta\right)^T\right] \geq \begin{bmatrix} \frac{(\alpha-1)\beta\sigma}{T\sigma-(\alpha-1)\beta} & 0 \\ 0 & \frac{2(\alpha-1)(\alpha-2)\beta^2}{T+2\alpha-2} \end{bmatrix}$$

Take Away: decreases by order $1/T$ where T is number of samples

Hybrid Cramér–Rao bound applicable to the joint estimation of random and non–random parameters.

Bounds on Parameter Inference Error

Learning to Learn: we can provide bounds on parameter inference error.

Cramér–Rao bound for hyper–parameters $\theta = (\alpha, \beta, \nu, \sigma)$: Let $\boldsymbol{X}^t$ be all observations up to time t. For estimator $\hat{\theta}(\boldsymbol{X}^t)$,

$$\underbrace{\mathbb{E}_{\boldsymbol{X}^t}\left[(\theta_i - \hat{\theta}_i)^2\right]}_{\text{MSE of } \theta_i} \geq \frac{1}{n\zeta_i}, \text{ where } \zeta_i = \underbrace{-\mathbb{E}_{\boldsymbol{X}^t}\left[\frac{\partial^2 \ln p(\boldsymbol{X}^t|\theta)}{\partial \theta_i^2}\right]}_{\text{curvature}}$$

Take Away: lower bound decreases by order $1/n$ (number of users)

Bayesian Cramér–Rao Bound for $\theta_r = (\mu_i, \lambda_i)$, for any estimator $\hat{\theta}_r$,

$$\mathbb{E}_{\xi^t,\theta}\left[\left(\hat{\theta}(\xi^t) - \theta\right)\left(\hat{\theta}(\xi^t) - \theta\right)^T\right] \geq \begin{bmatrix} \frac{(\alpha-1)\beta\sigma}{T\sigma-(\alpha-1)\beta} & 0 \\ 0 & \frac{2(\alpha-1)(\alpha-2)\beta^2}{T+2\alpha-2} \end{bmatrix}$$

Take Away: decreases by order $1/T$ where T is number of samples

Hybrid Cramér–Rao bound applicable to the joint estimation of random and non–random parameters.

Reliable Estimation of Stopping Time Algorithm

REST: data-driven method based on concentration inequalities.

Consider $n >> 1$ occupants and a planner with an objective $f : (\boldsymbol{x}^1, \ldots, \boldsymbol{x}^t) \mapsto f(\boldsymbol{x}^1, \ldots, \boldsymbol{x}^t) \in \mathbb{R}$ e.g.

- Estimate average lighting: $\frac{1}{t}\sum_{j=1}^{t} \underbrace{\left(\frac{1}{n}\sum_{i=1}^{n}[\boldsymbol{x}^j]_i\right)}_{\text{Avg. vote at time } j}$.
- Lighting energy: $\frac{1}{t}\sum_{j=1}^{t} \underbrace{g\left(\frac{1}{|S \backslash S^j_{\text{absent}}|}\sum_{i \notin S^j_{\text{absent}}}[\boldsymbol{x}^j]_i\right)}_{\text{Avg. energy at time } j}$
- Occupant classification error: $\frac{1}{n}\sum_{i=1}^{n} 1(h(\boldsymbol{x}^1_i, \ldots, \boldsymbol{x}^t_i) \neq \boldsymbol{y}_i)$

McDiarmid's Inequality

For $\forall j, \forall \boldsymbol{x}^1, \ldots, \boldsymbol{x}^t, \hat{\boldsymbol{x}}^j \in \mathscr{X}$, if $|f(\boldsymbol{x}^1, \ldots, \boldsymbol{x}^j, \ldots, \boldsymbol{x}^t) - f(\boldsymbol{x}^1, \ldots, \hat{\boldsymbol{x}}^j, \ldots, \boldsymbol{x}^t)| \leq c_j$ for $c_j > 0$ (bounded difference property), then for all $\varepsilon > 0$,

$$\Pr(f - \mathbb{E}[f] \geq \varepsilon) \leq \exp\left(\frac{-2\varepsilon^2}{\sum_{j=1}^{t} c_j^2}\right)$$

REST Algorithm — McDiarmid's method

Stopping time is determined empirically:

1. At time t, learn the parameters of the model, $\hat{\theta} \leftarrow M_{est}(\boldsymbol{x}^t)$
2. Simulate the data by the model, $\boldsymbol{x}_{sim}^{k-t} \leftarrow M_{sim}(\hat{\theta}, k-t)$
3. Determine the bounded difference property bound c_j

Theorem (McDiarmid's stopping time)

The estimated stopping time, τ_{MD}^{EST}, by the McDiarmid's method satisfies

$$\Pr\left(\left| f(\boldsymbol{X}^t) - \mathbb{E}\left[f(\boldsymbol{X}^t) \right] \right| \geq \varepsilon \right) \leq 1 - \delta, \quad \forall t \geq \tau_{MD}^{\text{EST}}$$

where $\boldsymbol{X}^t = (\boldsymbol{x}^1, \ldots, \boldsymbol{x}^t)$ is the data matrix up to time t, ε is the **desired precision**, and $1 - \delta$ is the **specified probability bound**.

REST Algorithm — Delta method

Delta method: approximate probability distribution for a function of an asymptotically normal statistical estimator

1. At time t, estimate the standard deviation of the sample $\hat{\sigma} \leftarrow std(y_1^t)$
2. Determine the stopping condition:

$$\Phi\left(\varepsilon; 0, \frac{f'(x^t)\hat{\sigma}}{\sqrt{k}}\right) - \frac{1}{2} \geq \frac{\delta}{2},$$

$\Phi(z; \mu, \theta)$ is the CDF of $\mathcal{N}(\mu, \theta)$ at point $z \in \mathbb{R}$.

Theorem (Delta method stopping time)

The stopping time estimate, $\tau_{Delta}^{\mathsf{EST}}$, by the Delta method satisfies:

$$\Pr\left(|g(\mathbf{X}^m) - \mathbb{E}[g(\mathbf{X}^m)]| \geq \varepsilon\right) \leq 1 - \delta, \quad \forall m \geq \tau_{Delta}^{\mathsf{EST}}$$

where $\mathbf{X}^t = (\boldsymbol{x}^1, \ldots, \boldsymbol{x}^t)$ is the data matrix up to time t, ε is the **desired precision**, and $1 - \delta$ is the **specified probability bound**.

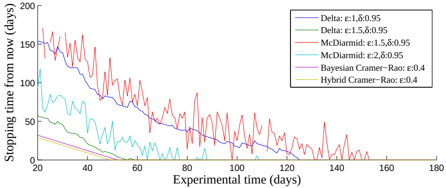
Learning Game–Theoretic Behavior Parameter ξ

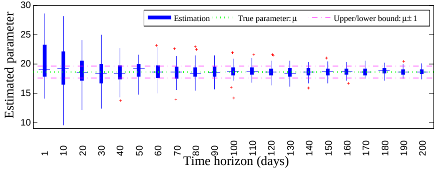
Task: estimate μ_i, mean of trade-off ξ_i of winning and comfort

- Trade-off factor ξ_i drawn from $\mathcal{N}(\mu_i, \lambda_i)$
- Multinomial actions: p_i^{def}default, p_i^{abs} absent, p_i^{act} active

$$\underbrace{\text{occupant utility}}_{\Psi_i(x_i,x_{-i},\gamma;\xi)} = \underbrace{\text{comfort}}_{\psi_0(x_i,x_{-i},\gamma(x_i,x_{-i}))} + \xi_i \underbrace{\text{desire to win}}_{\psi_1(x_i,x_{-i},\gamma(x_i,x_{-i}))}$$

Unbiased MLE: $\hat{\mu}_1 = \frac{1}{t}\sum_{k=1}^{t} \xi_1^k$ (sample mean)





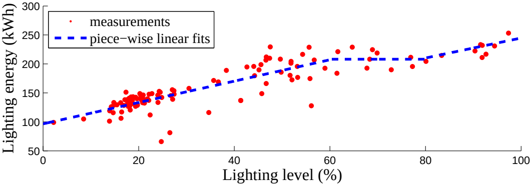
- Estimated Stopping Time via Cramér–Rao, McDiarmid, and Delta
- Remark: Estimates Concentrate within bounds around ~ 100 days

Energy Consumption Prediction

Task: estimate the effects of game dynamics on energy savings of the system: $E^{\text{light}} = g\left(\frac{1}{|S \backslash S_{\text{absent}}|} \sum_{i \notin S_{\text{absent}}} x_i\right)$

- Mean energy consumption, $\mu_E = \mathbb{E}\left[E^{\text{light}}\right]$
- Percentage of energy consumption below a certain threshold, $p_{E,\lambda} = \Pr\left(E^{\text{light}} < \lambda\right)$ for demand-response programs

- Piece-wise linear, only those present can control the lighting

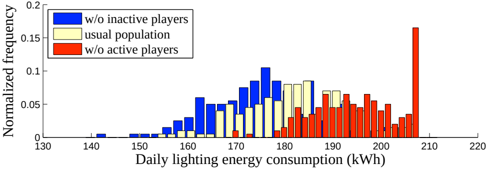


Energy consumption in social games is random and depends on the competitive environment

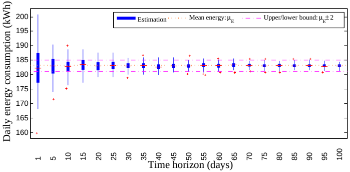
Energy Consumption Prediction

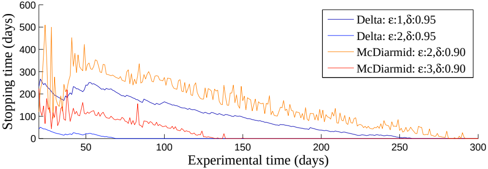
Remark: Active players influence more since they influence the dynamics

- Energy consumption with the top 10% active/inactive players removed



Result: Energy estimation starts to concentrate within the tolerance bounds around day 50





Classification of Player Category — Scaling to Many Players

Task: classify agents into different categories based on behavior.

- Incentive design: data-driven method to understand preferences
- Customer segmentation for energy management

User profiles:

	Lighting Comfort	Incentive Award	Game Participation
Comforter	★★★	★	★★★★
Gamer	★	★★★	★★★★
Balancer	★★	★★	★★★★
Nonchalancer	★	★	★★

** each $\star$ indicates abstractly the amount the user types care about each of the categories

Classification of Player Category — Metrics

Method: compare the empirical distribution P_i of ξ_i^t to the distribution Q_j of ξ_j for category j using JS Divergence:

$$JSD(P||Q) = \frac{1}{2}D(P||M) + \frac{1}{2}D(Q||M)$$

where $M = \frac{1}{2}(P+Q)$, and $D(P||M) = \sum_k P(k)\ln\frac{P(k)}{Q(k)}$.

- Classification rule:

$$h(x_i^t) = \arg\min_{j\in\{1,2,3,4\}} JSD(P_i||Q_j).$$

- Cost function (proportion of misclassified users):

$$L(h;\boldsymbol{x}^t,\boldsymbol{y}) = \frac{1}{n}\sum_{i=1}^{n} 1(h(\boldsymbol{x}_i^t) \neq \boldsymbol{y}_i);$$

Classification of Player Category — Results

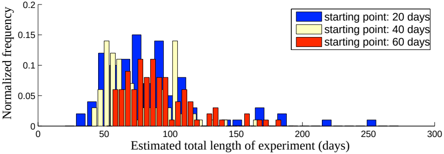
Random function of deviation from the best performance:

$$f(h;\boldsymbol{x}^t,\boldsymbol{y}) = L(h;\boldsymbol{x}^t,\boldsymbol{y}) - \inf_{h\in\mathscr{H}} \mathbb{E}\left[L(h;\boldsymbol{x},\boldsymbol{y})\right];$$

Bound on loss: $\Pr\left(L_t \geq \varepsilon + \kappa_t + \rho\right) \leq \Pr\left(f_t - \mathbb{E}\left[f_t\right] \geq \varepsilon\right) \leq 1-\delta$

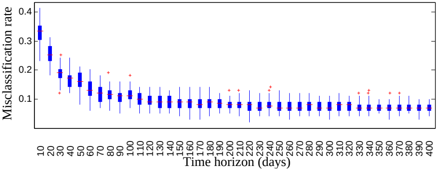
Simulation: 33 replicates of each type using occupant models generated out of data from real experiments (distribution of ξ_i).

Estimated stopping time



mean= 89 (blue), 78 (yellow), 94 (red)

Misclassification error plateaus ≈ day 80



REST captures the complexity of the problem

Conclusion and Future Work

Algorithm for computing sampling time via concentration inequalities

- Allows for scaling from individual inference tasks to higher-order learning tasks with many players.
- **e.g.** parameter inference, energy consumption estimation, user type classification
- Results can improve automation and control, provide guarantees for demand response programs, and for incentives design
- **generic enough to apply to any scenario where learning behavior of large numbers of competitive agents is desired!**

Conclusion and Future Work

Algorithm for computing sampling time via concentration inequalities

- Allows for scaling from individual inference tasks to higher-order learning tasks with many players.
- **e.g.** parameter inference, energy consumption estimation, user type classification
- Results can improve automation and control, provide guarantees for demand response programs, and for incentives design
- **generic enough to apply to any scenario where learning behavior of large numbers of competitive agents is desired!**

Future work:

- Employ algorithm in practice in Singapore with large number of agents.
- Improve the behavioral model of competitive agents: model-based vs. data-driven techniques for learning utility functions
- Add privacy guarantees, e.g. differential privacy constraints.