Sequential Decision Making: A Tutorial

Yan Chen, Chih-Yu Wang

School of Elect. Eng., University of Electronic Science and Technology of China
Research Center for Information Technology Innovation, Academia Sinica

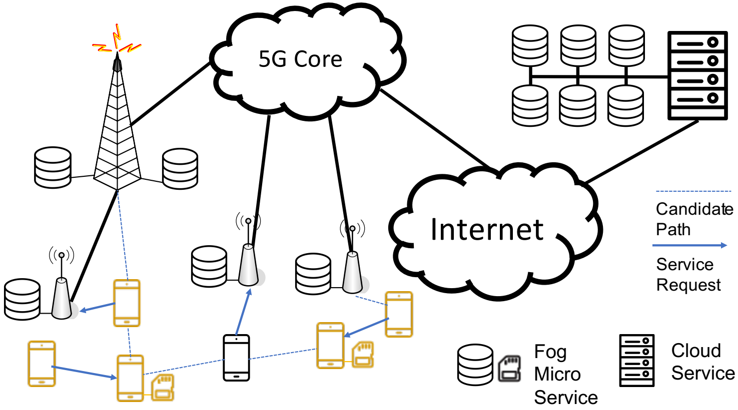
Outline

Choosing a restaurant in a food corner

Some people made decisions before you...

- Which line to join: quality vs. waiting time

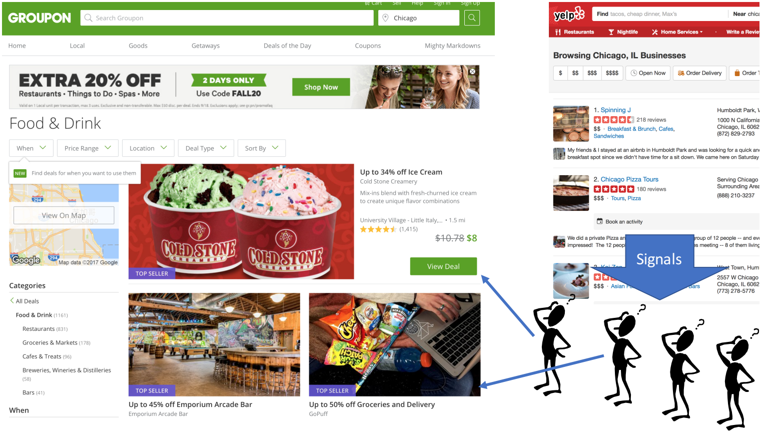
Example 1: Service Access at Fog Computing



Fog Service Selection: which service entity to send request?

- Transmission vs. Computing latency

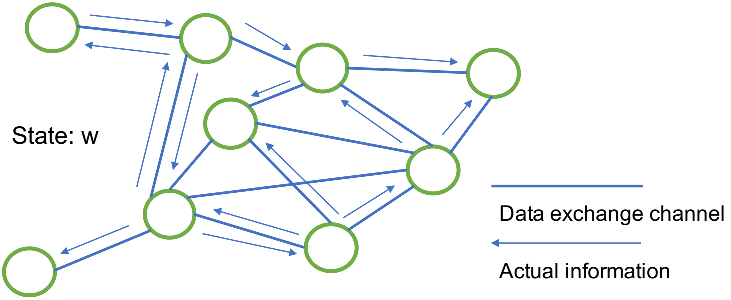
Example 2: Deal Selection on Social Media Website



Deal Selection

- Meal Quality vs. Service Quality

Example 3: Estimation in Distributed Adaptive Filtering



State estimation strategy update

- Relies on neighbor's information
- To trust or not to trust?

Observations

- Decisions are usually made in sequential
  $\rightarrow$ timing difference
- Agents have different amounts of information
  $\rightarrow$ information asymmetry
- The utility of an agent is influenced by the decisions of all agents
  $\rightarrow$ network externality

Rational Decision Making

Collect information (learning) about uncertain states

- Signals, rumors collected by the agents
- Information shared by other agents
- Actions revealed by other agents

Estimate (predicting) the corresponding utility

- Conditioning on the available information
- Predicting the decisions of subsequent agents

Make the optimal decision by maximizing the expected utility

Approach: Social Learning

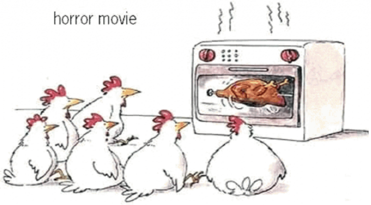
A cognitive process

- Learn information from the observed actions and the corresponding consequence
- Observation → Knowledge Extraction → Decision

Model

- Theory: Bayesian and non-Bayesian learning
- Goal: consensus, learning sequence, information cascade

Limitation: Network externality is ignored



Approach: Multi-Armed Bandit

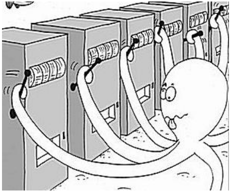
A gambler stands at the row of slot machines, each returns a random reward if the gambler plays

- The optimal strategy is to select the machine (bandit) by maximizing the long term reward

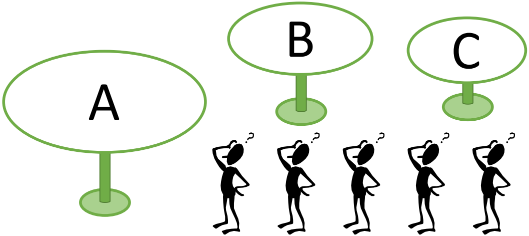
Solution concepts

- Exploration versus exploitation
- Converging to the optimal policy by minimizing the regret

Limitation: single player $\rightarrow$ no competition



What You will Learn



- Nash Game
- Bayesian Game
- Table Selection Problem

Game Theory Basic

- Static System: Influence from Others
- Dynamic System: Predicting the Future

Network Externality

- Acquired: Learning from Signals
- Observed: Learning from Actions

Sequential Learning

- Prediction
- Pricing
- Voting

Management

Basic Game

Game: a set of players make moves to maximize their reward following the given rules.

- Players: $\mathcal{N} = \{1, 2, 3, ..., N\}$
- Actions: $\mathbf{a} = \{a_1, a_2, ..., a_N\} \in \mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2 \times ... \times \mathcal{A}_N$
- Utility Functions: $\mathcal{U} = \{U_1(\mathbf{a}), U_2(\mathbf{a}), ..., U_N(\mathbf{a})\}$

Example: Prison of Dilemma

Two criminals, each chooses to stay silent or betray. The decision influences the number of years in jail

Action	P_2 Stays Silent	P_2 Betrays
P_1 Stays Silent	$(-1, -1)$	$(-10, 0)$
P_1 Betrays	$(0, -10)$	$(-2, -2)$

Solution Concepts

Traditional centralized solutions for multi-objective optimization

Weighted Sum Optimal

$$\max_{\mathbf{a} \in \mathcal{A}} \sum_{i \in \mathcal{N}} w_i U_i(\mathbf{a})$$

when $w_i = 1 \; \forall i \in \mathcal{N}$, the solution is **social optimal**

Pareto Optimal

A solution $\mathbf{a}$ is Pareto optimal if for any $\mathbf{a}' \in \mathcal{A}$,

$$\exists i \in \mathcal{N}, U_i(\mathbf{a}') < U_i(\mathbf{a})$$

Any social optimal solution is a Pareto optimal solution.

Solution Concepts

Nash Equilibrium

A game with players $1, 2, ..., N$. Each player i has an action space $\mathcal{A}_i$ and a utility function $U_i(a_i, \mathbf{a}_{-i})$, where a_i is the player's action and $\mathbf{a}_{-i}$ is the action profile of all players except player i. Nash equilibrium is the action profile $\mathbf{a}^* = \{a_1^*, a_2^*, ..., a_N^*\}$ where

$$U_i(a_i^*, \mathbf{a}_{-i}^*) \geq U_i(a_i, \mathbf{a}_{-i}^*), \forall i \in \mathcal{N}, a_i \in \mathcal{A}_i.$$

- A game reaches NE when no one has the incentive to change their actions even if they could $\rightarrow$ everyone has applied their **best response**
- Rational prediction on the outcome of the game

Prison of Dilemma

Example

Action	P_2 Stay Silent	P_2 Betrays
P_1 Stay Silent	$(-1, -1)$	$(-10, 0)$
P_1 Betrays	$(0, -10)$	$(-2, -2)$

The best response of Player 1

- If Player 2 stays silent, it is better to betray to get free
- If Player 2 betrays, it is better to betray for fewer years in jail
- Betray is **dominant** strategy

Social-Optimal Solution

- $(a'_1, a'_2) = \{Silent, Silent\}$, sum of years: 2 years

Nash Equilibrium

- $(a^*_1, a^*_2) = \{Betray, Betray\}$, sum of years: 4 years

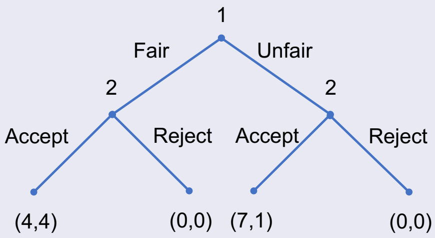
Nash Equilibrium may not be efficient $\rightarrow$ Price of Anarchy

Sequential Game

A game that players make decisions at different time

Ultimatum Game

Two players share a cake. Player 1 determines the shares, and player 2 determines whether to accept or reject the offer.



Nash Equilibrium?

Subgame-Perfect Nash Equilibrium

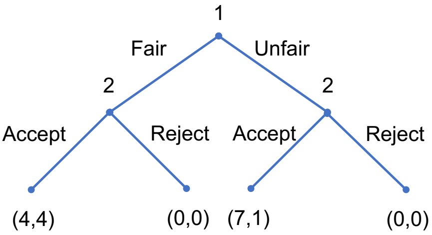
- A subgame is a part of a sequential game, starting from an initial point and including all successors

Subgame-perfect Nash Equilibrium

A Nash equilibrium is a subgame-perfect Nash equilibrium if and only if it is also a Nash equilibrium for any subgame.

Every response in the subgame is rational

Ultimatum Game



(Fair, (Accept, Reject))

- Nash Equilibrium
- Player 2 claims that she will reject unless it is fair

(Unfair, (Accept, Accept))

- Nash equilibrium and Subgame-Perfect Nash equilibrium
- Player 2 takes whatever it receives

Bayesian Game

A game with some unknown information

- Player set $\mathcal{N}$ and Action set $\mathcal{A}$
- Type of players $\mathbf{t} = (t_1, t_2, ..., t_N) \in \mathcal{T} = \mathcal{T}_1 \times \mathcal{T}_2 \times ... \times \mathcal{T}_N$

Unknown System State

- State $\theta \in \Theta$ (unknown to some or all players)
- Probability (belief) $\mathbf{p_i}$ on state θ over state space Θ
- Utility Functions $U_i(\mathbf{a}, \theta)$, $\mathbf{a} \in \mathcal{C}$

The utilities of players depend on **both** state and actions

- The state is unknown $\rightarrow$ expected utility
- The probability may be influenced by observed signals or actions $\rightarrow$ (Bayesian) learning

Bayesian Nash Equilibrium

Players maximize their expected utilities based on belief

$$\mathbf{p_i}(\mathbf{I_i}) = \{p_{i,\theta} | \theta \in \Theta\}, \sum_{\theta \in \Theta} p_{i,\theta}(I_i) = 1$$

where I_i is the information received by player i in the game.

$$p_{i,\theta}(I_i) = \frac{Prob(I_i|\theta)}{\sum_{\theta' \in \Theta} Prob(I_i|\theta')}$$

Bayesian Nash Equilibrium

Bayesian Nash equilibrium is the action profile $\mathbf{a}^*$ where

$$\sum_{\theta \in \Theta} p_{i,\theta}(I_i) U_i(a_i^*, \mathbf{a}_{-i}^*, \theta) \geq \sum_{\theta \in \Theta} p_{i,\theta}(I_i) U_i(a_i, \mathbf{a}_{-i}^*, \theta), \forall i \in \mathcal{N}, a_i \in \mathcal{A}_i.$$

Reputation Game

Firm 1 is in the market and prefers monopoly. Firm 2 is entering the market. Firm 1 has two types: Sane and Crazy (50-50).

	Stay	Exit
Sane / Prey	(2,5)	(X,0)
Sane / Accommodate	(5,5)	(10,0)
Crazy / Prey	(0,-10)	(0,0)

Equilibrium Types

Pooling Equilibrium: When $X = 8$, Both Sane and Crazy Firm 1 will choose to prey, and Firm 2 will exit

Separating Equilibrium: When $X = 2$, Sane Firm 1 will Accommodate, and Firm 2 will Stay when seeing Accommodate and exit when seeing Prey. (Firm 1's action is a signal)

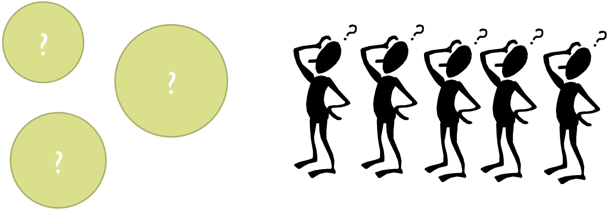
Dining in a Chinese Restaurant

A Chinese restaurant serves some meals

- Finite tables, each with different but **unknown** sizes

Customers arrive sequentially

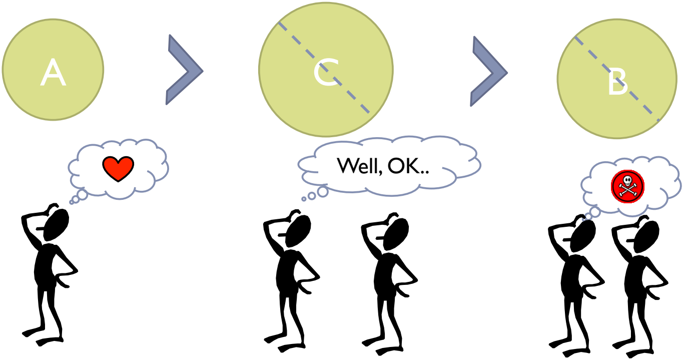
- Choose a table **before** entering the restaurant
- Have some (inaccurate) knowledge on the table size



Dining in a Chinese Restaurant

All customers prefer bigger dinning space

- Tables have different sizes
- ...but some tables get crowded, some do not

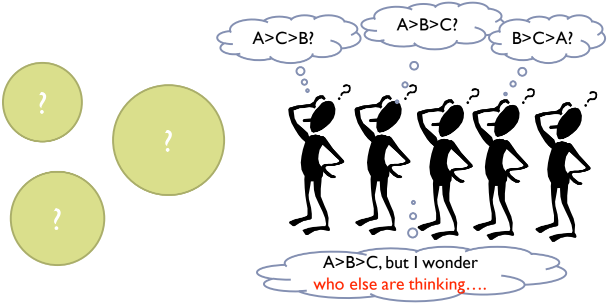


Dining in a Chinese Restaurant

A rational customer chooses the table with biggest dinning space

Challenge: Customers do not know

- The exact table sizes
- and the decisions of subsequent customers!



Problem: optimal table selection strategy for each customer

Outline

Simple Chinese Restaurant Game

Simultaneous Game

All customers choose the table at the same time

- No sequential move
- Only network externality

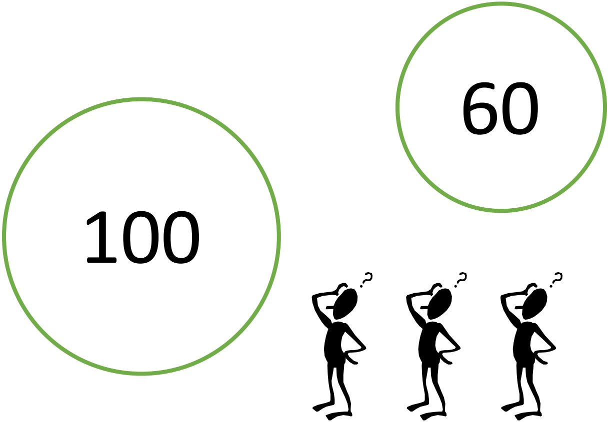
K tables, each with size $R_j(\theta)$

- System state: $\theta \in \Theta$
- Table sizes are determined when θ is given

N Customers

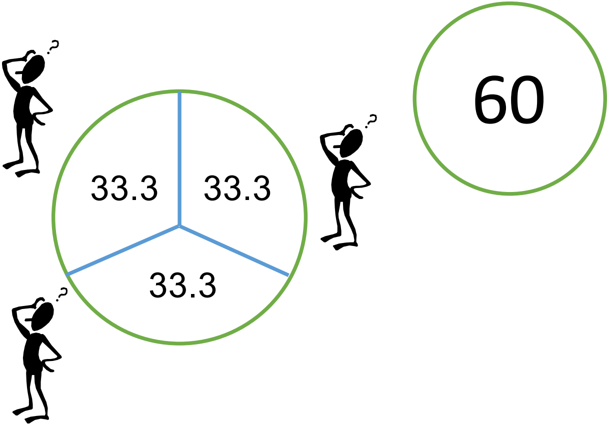
- Actions: $x_i \in \{1, 2, ..., K\}$
- Utility:
  - n_x: number of users choosing x at the end of the game
  - $U(R_{x_i}(\theta), n_{x_i})$: increasing with $R_{x_i}(\theta)$, decreasing with n_{x_i}

Nash Equilibrium in CRG



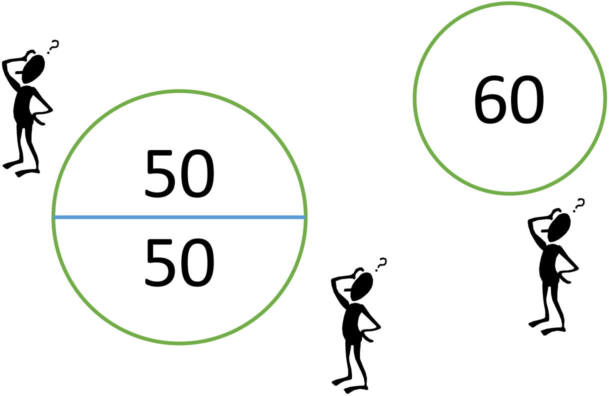
Which table would you choose?

Nash Equilibrium in CRG



Anyone wants to change your mind?

Nash Equilibrium in CRG



Anyone wants to change your mind?

Equilibrium Grouping

Theorem (Equilibrium Grouping)

Given the current system state θ, for any Nash equilibrium of the Chinese restaurant game with perfect signal, its equilibrium grouping $\mathbf{n}^* = \{n_1^*, n_2^*, ..., n_K^*\}$ should satisfy

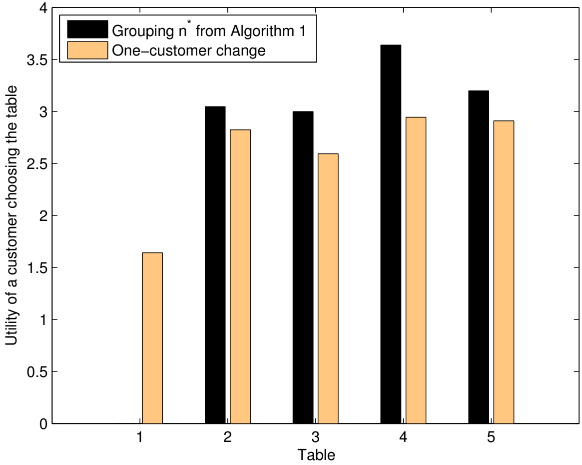
$$U(R_x(\theta), n_x^*) \geq U(R_y(\theta), n_y^* + 1), \ \textit{if } n_x^* > 0, \forall x, y \in \{1, 2, ..., K\}$$

- A customer will have less utility if he chooses another table

Observations

- Equilibrium grouping $\mathbf{n}^*$ determines the expected utility offered by each table
- Customers in different tables may have different utilities

Equilibrium Grouping



Uniqueness of Equilibrium Grouping

Theorem (Uniqueness of Equilibrium Grouping)

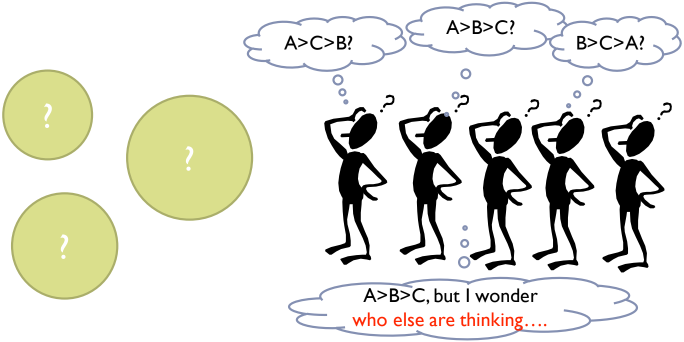
If the inequality in equilibrium grouping conditions strictly holds for all $x, y \in \mathcal{X}$, then the equilibrium grouping $\mathbf{n}^* = (n_1^*, ..., n_J^*)$ is unique.

The outcome is "predictable" in some sense, and can be found through a myopic response algorithm

- Customers sequentially choose the table myopically
- Equilibrium grouping is reached when all customers are seated

Sequential Chinese Restaurant Game

- Customers arrive and choose the table sequentially
- Customer k: I saw the choices of customer $1 \sim k-1$, and I know customer $k+1 \sim N$ will see what I choose (and think)...
- Every customer is facing a different game



Sequential Game: Advantage of Playing First

Perfect signal $\rightarrow$ the state θ is completely revealed

Theorem (Equilibrium Grouping)

Given the current system state θ, a sequential Chinese restaurant game with perfect signal's equilibrium grouping $\mathbf{n}^* = \{n_1^*, n_2^*, ..., n_K^*\}$ should satisfy

$$U(R_x(\theta), n_x^*) \geq U(R_y(\theta), n_y^* + 1), \text{ if } n_x^* > 0, \forall x, y \in \{1, 2, ..., K\}$$

Observations

- Equilibrium grouping $\mathbf{n}^*$ determines the expected utility offered by each table
- Customers in different tables may have different utilities
- Same equilibrium grouping as the one in simultaneous game

Finding Equilibrium Grouping

Theorem (Existence of Subgame-Perfect Nash Equilibrium)

There always exists a subgame perfect Nash equilibrium with the corresponding equilibrium grouping $\mathbf{n}^*$ in a sequential Chinese restaurant game.

Strategy in subgame-perfect Nash equilibrium

- Choose best table among those not "full" yet according to $\mathbf{n}^*$
- If already deviated, find a new equilibrium grouping from observed $\mathbf{n}_i$

Observations

- Customers playing early choose the table with larger expected utility according to predicted $\mathbf{n}^*$
- When you have perfect knowledge (and certain that others do, too), choose earlier

Dynamic System

Dynamic System

Customers enter and leave the system dynamically

- Tables remain the same with sizes known by all customers
- Customers only stay for an (undetermined) period of time

How We Define Utility?

- Immediate utility: the utility a user may receive for a short period of time (slot) $\rightarrow$ may change with time
- Long-term utility: the utility a user receives in total within the duration of her stay

Optimal Table Selection Strategy?

Wireless Access Network Selection

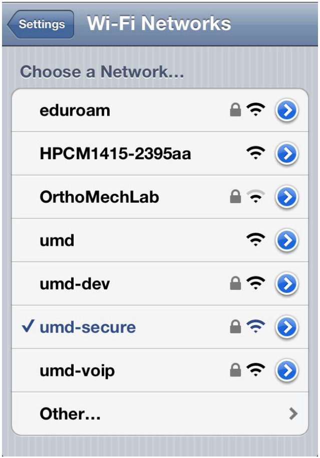
Mobile Internet Access

- Multiple wireless network services available: Wi-Fi, 3G/4G/5G...
- Multiple wireless network accesses available: Wi-Fi APs, BSs

Traditional access strategy

- Centralized admission control: scalability
- Priority-based access policy: ignore network status
- SINR-based access policy: ignore network externality

Rational access strategy?



Wireless Access Network Game

K networks, each can server up to N users

- Any further connection request will be rejected when the network is full

Deterministic users for network k

- Poisson arrival rate $\bar{\lambda}_k$
- The duration of the stay follows exponential duration $\bar{\mu}$
- Can only access network k

Rational users

- Poisson arrival rate $\bar{\lambda}_0$
- The duration of the stay follows exponential duration $\bar{\mu}$
- Can choose any network to access
- Receive utility $R_k(s_k)$ at each time slot if choosing network k
- $R_k(s_k)$ decreases when s_k increases

Wireless Access Network Game

Game Model

- Player: rational users
- Action: network $k \in K$
- Utility: expected long-term utility in the network

Challenges

- Heterogeneous network characteristic
- Stochastic user population

How do we measure expected utility and determine optimal action?

Multi-Dimensional Markov Decision Process

Markov Decision Process

- (Global) reward depending on the system state and action chosen by one coordinator
- State transition is Markovian
- Goal: finding the optimal policy that maximizes the reward

Extending Markov Decision Process to Multi-Dimensional form:

Multi-Dimensional Markov Decision Process

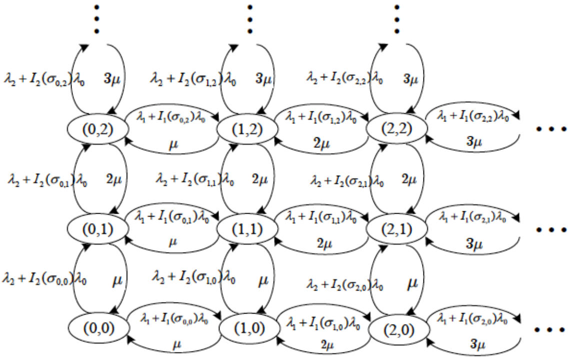
- System state at time t:$\mathbf{s} = (s_1, s_2, ..., s_K)$
- Reward of each network k:$R_k(\mathbf{s})$
- Policy for Network Access: $\pi(\mathbf{s}) \in K$

- Policy is determined by multiple players
- Multiple reward functions instead of single global one

State Transition

System state changes when

- A new user arrives and enters one of the networks
- An existing user leaves a network



Policy $\pi(\mathbf{s})$ determines the state transition probability

State Transition

There are two perspectives

Transition Probability observed by ordinator

$$
Pr(\mathbf{s}'|\mathbf{s},\pi) = \begin{cases} \pi(\mathbf{s})\lambda_0 + \lambda_{\pi(\mathbf{s})}, & \mathbf{s}' = (s_1, ..., s_{\pi}(\mathbf{s}) + 1, ...) \\ \lambda_j, & \mathbf{s}' = (s_1, ..., s_j + 1, ...), j \neq \pi(\mathbf{s}) \\ & \text{and } \pi(\mathbf{s}) > 0; \\ s_{k'}\mu, & \mathbf{s}' = (s_1, ..., s_{k'} - 1, ...); \\ 1 - \sum s_{k'}\mu - \sum_{j=0}^{K} \lambda_j, & \mathbf{s} = \mathbf{s}', \pi(\mathbf{s}) > 0; \\ 1 - \sum s_{k'}\mu - \sum_{j=1}^{K} \lambda_j, & \mathbf{s} = \mathbf{s}', \pi(\mathbf{s}) = 0; \\ 0, & \text{else.} \end{cases}
$$

State Transition

There are two perspectives

Transition Probability observed by user at network k

$$Pr(\mathbf{s}'|\mathbf{s},\pi,k) = \begin{cases} \pi(\mathbf{s})\lambda_0 + \lambda_{\pi(\mathbf{s})}, & \mathbf{s}' = (s_1, ..., s_\pi(\mathbf{s}) + 1, ...) \\ \lambda_j, & \mathbf{s}' = (s_1, ..., s_j + 1, ...), j \neq \pi(\mathbf{s}) \\ & \text{and } \pi(\mathbf{s}) > 0; \\ s_{k'}\mu, & \mathbf{s}' = (s_1, ..., s_{k'} - 1, ...), k' \neq k; \\ (s_k - 1)\mu, & \mathbf{s}' = (s_1, ..., s_{k'} - 1, ...); \\ 1 - (\sum s_{k'} - 1)\mu - \sum_{j=0}^{K} \lambda_j, & \mathbf{s} = \mathbf{s}', \pi(\mathbf{s}) > 0; \\ 1 - (\sum s_{k'} - 1)\mu - \sum_{j=1}^{K} \lambda_j, & \mathbf{s} = \mathbf{s}', \pi(\mathbf{s}) = 0; \\ 0, & \text{else.} \end{cases}$$

This user in network k still stays in this network if she can observe

Expected Utility

Typical approach: sum of immediate utility

$$E\left[U_k(\mathbf{s^{t_e}})\right] = E\left[\sum_{t=t^e}^{\infty}(1-\mu)^{t-t^e} R_k(\mathbf{s_t})|s^{t_e}, \pi\right]$$

We focus on the stationary state $\rightarrow$ Bellman equations

$$W_k(\mathbf{s}) = R_k(\mathbf{s}) + (1-\mu)\sum_{\mathbf{s}'} Pr(\mathbf{s}'|\mathbf{s}, \pi, k) W_k(\mathbf{s}') \qquad (1)$$

The rational responses for a new user observing state $\mathbf{s}$ should be

$$\pi(\mathbf{s}) = \arg\max_{k\in K, s_k<N} W_k(\mathbf{s}) \qquad (2)$$

Expected utility and rational responses couple together through state transition probability

Equilibrium Conditions

Equilibrium Conditions

The policy $\pi^*(\mathbf{s})$ is a Nash equilibrium if and only if

$$W_k^*(\mathbf{s}) = R_k(\mathbf{s}) + (1-\mu)\sum_{\mathbf{s}'} Pr(\mathbf{s}'|\mathbf{s},\pi^*,k)W_k^*(\mathbf{s}')$$

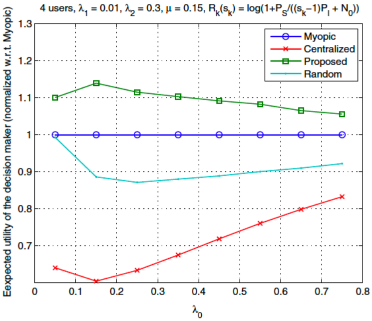
$$\pi^*(\mathbf{s}) = \arg\max_{k\in K, s_k<N} W_k^*(\mathbf{s})$$

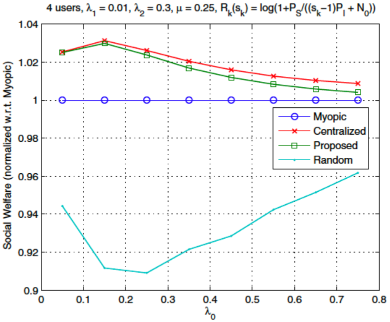
Nash equilibrium can be found through value-iteration algorithm

1. Initialize π, update transition probability $Pr(\mathbf{s}'|\mathbf{s},\pi)$
2. Update expected reward W_k with (1)
3. Update π with (2)
4. Repeat 2 and 3 until π converges

Performance Evaluation

Two networks, three user sets $(\lambda_0, \lambda_1, \lambda_2)$





- Equilibrium strategy provides highest individual utility
- Overall social welfare is closest to optimal one

Outline

Learning for Knowledge

Sequential Decision Making with Incomplete Information

- Information asymmetry
- Collect information $\rightarrow$ Learn knowledge

Information revealed in the network

- Signals: sensing result, ratings, reviews...
- Actions: queues, orders, subscriptions...

Knowledge to learn from collected information

- Unknown state of the system
- Prediction on behaviors of other players

Sequential Chinese Restaurant Game

Assuming that the state and players won't change within the game.

K tables, each with size $R_j(\theta)$

- System state: $\theta \in \Theta$
  - Unknown to players $\rightarrow$ Information to learn

N Customers

- Each with an informative signal $s_i \in \mathbf{S} \sim f(s|\theta)$
  - "hint" for the unknown state $\theta \rightarrow$ table size $R_j(\theta)$
- Actions: $x_i \in \{1, 2, ..., K\}$
- Utility:
  - n_x: number of users choosing x at the end of the game
  - $U(R_{x_i}(\theta), n_{x_i})$: increasing with $R_{x_i}(\theta)$, decreasing with n_{x_i}

Example

Two tables, two possible orders (states)

State	Table 1	Table 2
1	100	50
2	50	100

- System state $\theta \in \{1, 2\}$
- Table size function: $R_x(\theta) = \begin{cases} 100, & \text{if } x = \theta, \\ 50, & \text{otherwise} \end{cases}$

N customers

- Signal: $s \in \{1, 2\}$, $f(s|\theta) = \begin{cases} 0.9, \text{if } s = \theta, \\ 0.1, \text{otherwise} \end{cases}$
- Signal likely (but not exactly) reflects the true system state
- Utility function: $U(R, n) = R/n$

Sequential with Imperfect Signal

Customers make decisions sequentially

- Without perfect knowledge of the state $\rightarrow$ Signal is important
- After making decision, her signal is announced to all others
  - Customers making decisions later have more information

Information observed by customer i

- Choices of previous customers: $\mathbf{n_i} = \{n_{i,j} | j = 1, 2, ..., K\} \rightarrow$ **grouping**
- Signals of previous customers: $\mathbf{h_i} = \{s_j | j = 1, 2, ..., i - 1\}$
- Her received signal: s_i

Imperfect Signal: Belief in State

Information set

- Grouping: $\mathbf{n_i} = \{n_{i,j} | j = 1, 2, ..., K\}$
- Signals of previous customers: $\mathbf{h_i} = \{s_j | j = 1, 2, ..., i-1\}$
- Her received signal: s_i

Belief: Estimation on the current state θ

$$g_{i,j} = Pr(\theta = j | (\text{information observed by customer } i))$$

Bayesian Learning Update

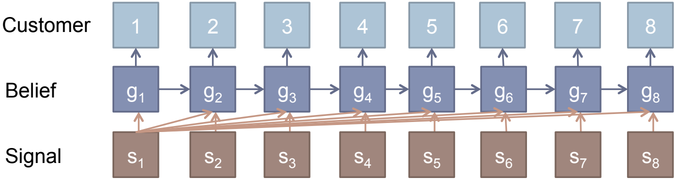
Prior belief (common knowledge, unconditional)

- $g_{0,j} = Pr(\theta = j)$

Signals revealed sequentially

- Belief is updated when new signal is revealed

$$
\begin{aligned}
g_{i,j} &= Pr(\theta = j|\mathbf{h}_i, s_i) = \frac{Pr(s_i|\theta = j)Pr(\theta = j|\mathbf{h}_i)}{\sum_{j' \in \Theta} Pr(s_i|\theta = j')Pr(\theta = j'|\mathbf{h}_i)} \\
&= \frac{g_{i-1,j}Pr(s_i|\theta = j)}{\sum_{j' \in \Theta} g_{i-1,j'}Pr(s_i|\theta = j')}
\end{aligned}
$$



Bayesian Nash Equilibrium

Best response of customer i, given the observed information:

$$BE_i(\mathbf{n}_i, \mathbf{h}_i, s_i) = \arg \max_{x_i \in \{1,2,\ldots,K\}} E[U(x_i)|\mathbf{n}_i, \mathbf{h}_i, s_i]$$

The expected utility is given by

$$\begin{aligned} & E[U(x_i)|\mathbf{n}_i, \mathbf{h}_i, s_i, x_i = j] \\ & = \sum_{w \in \Theta} g_{i,w} E[U(R_j(w), n_j)|\mathbf{n}_i, \mathbf{h}_i, s_i, x_i = j, \theta = w] \end{aligned}$$

A closed-form solution is generally impossible

- A recursive method is proposed

Recursive Best Response

Again, we find the outcome through checking the possible results in all subgames

Backward Induction

- Find the best response of last player under **all subgames**
- Given the response of player $i+1 \sim N$, find player i's best response under all subgames
- Repeat until all players' best responses are derived.

Recursive Best Response

- Take customer $i+1$'s best response $BE_{i+1}(\cdot)$ to derive $BE_i(\cdot)$

Predicting the choice of next customer with $BE_{i+1}(\mathbf{n_{i+1}}, h_{i+1}, s_{i+1})$

Recursive Best Response

The decisions of remaining players (random variable)

$$m_{i,j} = n_j - n_{i,j}$$

Let's assume that the distribution $Pr(m_{i,j} = X|...)$ is known

- Expected utility of customer i can be written as

$$\begin{aligned} & E[U(x_i)|\mathbf{n}_i, \mathbf{h}_i, s_i, x_i = j] \\ = & \sum_{w \in \Theta} \sum_{X=0}^{N-i+1} g_{i,w} Pr(m_{i,j} = X|\mathbf{n}_i, \mathbf{h}_i, s_i, x_i = j, \theta = w) U(R_j(w), n_{i,j} + X) \end{aligned}$$

Recursive: derive $Pr(m_{i,j} = X|...)$ and $BE_i(\cdot)$ with $Pr(m_{i+1,j} = X|...)$ and $BE_{i+1}(\cdot)$

Recursive Best Response

Recursive estimation on $Pr(m_{i,j}|\ldots)$

$$Pr(m_{i,j} = X|\mathbf{n_i}, \mathbf{h_i}, s_i, x_i, \theta = l) = \begin{cases} Pr(m_{i+1,j} = X - 1|\mathbf{n_i}, \mathbf{h_i}, s_i, x_i, \theta = l), & x_i = j, \\ Pr(m_{i+1,j} = X|\mathbf{n_i}, \mathbf{h_i}, s_i, x_i, \theta = l), & x_i \neq j, \end{cases}$$

$$= \begin{cases} \sum_{u \in \{1,\ldots,J\}} \int_{s \in \mathcal{S}_{i+1,u}(\mathbf{n_{i+1}}, \mathbf{h_{i+1}})} Pr(m_{i+1,j} = X - 1|\mathbf{n_{i+1}}, \mathbf{h_{i+1}}, s_{i+1} = s, x_{i+1} = u, \theta = l) f(s|\theta = l) ds, & x_i = j \\ \sum_{u \in \{1,\ldots,J\}} \int_{s \in \mathcal{S}_{i+1,u}(\mathbf{n_{i+1}}, \mathbf{h_{i+1}})} Pr(m_{i+1,j} = X|\mathbf{n_{i+1}}, \mathbf{h_{i+1}}, s_{i+1} = s, x_{i+1} = u, \theta = l) f(s|\theta = l) ds, & x_i \neq j \end{cases}$$

Best Response of customer i

$$BE_i(\mathbf{n_i}, \mathbf{h_i}, s_i) = \arg\max_j \sum_{l \in \Theta} \sum_{x=0}^{N-i+1} g_{i,l} Pr(m_{i,j} = x|\mathbf{n_i}, \mathbf{h_i}, s_i, x_i = j, \theta = l) U(R_j(l), n_{i,j}+x)$$

Recursive Best Response

The last customer's best response

$$BE_N(\mathbf{n_N}, h_N, s_N) = \arg\max_j \sum_{l \in \Theta} g_{N,l} u_N(R_j(l), n_{N,j} + 1).$$

$$Pr(m_{N,j} = 1 | \mathbf{n_N}, \mathbf{h_N}, s_N, x_N, \theta) = \begin{cases} 1, & \text{if } x_N = j, \\ 0, & \text{otherwise.} \end{cases}$$

The best responses of all customers can be derived recursively from customer N to customer 1

Simulation Settings

A Chinese restaurant game with 2 tables

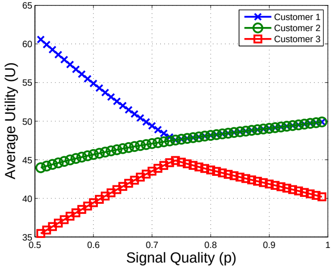
- System state $\theta \in \{1, 2\}$
- Table size function: $R_x(\theta) = \begin{cases} 100, & \text{if } x = \theta, \\ 100r, & \text{otherwise} \end{cases}$
- $0 \le r \le 1$: table size ratio

N customers

- Signal: $s \in \{1, 2\}$, $f(s|\theta) = \begin{cases} p, \text{if } s = \theta, \\ 1 - p, \text{otherwise} \end{cases}$
- $0 \le p \le 1$: signal quality
- Utility function: $U(R, n) = R/n$

Simulation Results

Scenario 1: Resource Pool ($r = 0.4$)



(a) 3 Customers

Signals	Best Response	
s_1, s_2, s_3 p	0.9	0.6
2,2,2	2,2,1	1,2,2
1,2,2	1,2,2	2,1,2
2,1,2	2,1,2	1,2,2
1,1,2	1,1,2	2,1,1
2,2,1	2,2,1	1,2,2
1,2,1	1,2,1	2,1,1
2,1,1	2,1,1	1,2,1
1,1,1	1,1,2	2,1,1

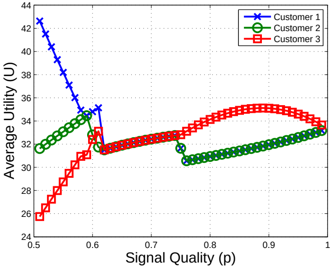
(b) Best Response when N=3

Playing first have significant advantages

- Network externality dominates

Simulation Results

Scenario 2: Available/Unavailable ($r = 0$)



(a) 3 Customers

Signals	Best Response		
s_1, s_2, s_3 p	0.9	0.7	0.55
2,2,2	2,2,2	2,2,2	1,2,2
1,2,2	1,2,2	1,2,2	2,1,2
2,1,2	2,1,2	2,1,2	1,2,2
1,1,2	1,1,1	1,1,2	2,1,1
2,2,1	2,2,2	2,2,1	1,2,2
1,2,1	1,2,1	1,2,1	2,1,1
2,1,1	2,1,1	2,1,1	1,2,1
1,1,1	1,1,1	1,1,1	2,1,1

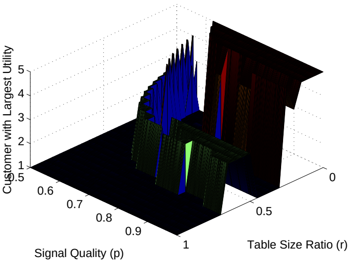
(b) Best Response when $N = 3$

Playing latter have the advantage to identify the better table

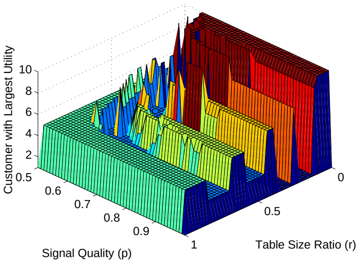
- With good enough signals, otherwise network externality still dominates

Simulation Results

Playing Positions vs. Signal Quality



(a) 5 Customers



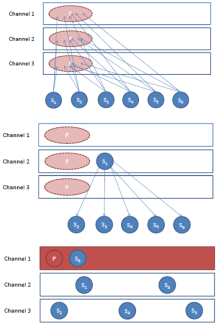
(b) 10 Customers

Playing in the middle may have advantages

- Enough signals to identify the better table
- Not too late to choose it before "full"

App: Cooperative Sensing in Cognitive Radio

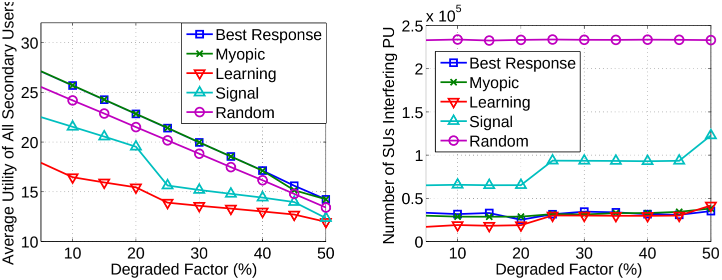
- Secondary users sense the activity of primary user
  - Poor detection accuracy
- Cooperative sensing
  - Users share their sensing results
  - Make smarter decisions
- Negative externality
  - More users accessing the same channel → less access time



Objective

Optimal cooperative sensing + channel selection strategy

Simulation Results



(a) Average Utility (b) SUs Interfering PU

- Signal - No cooperative sensing
- Learning - Traditional Cooperative Sensing
  - Best performance in interference avoidance
  - Worst for secondary users (Negative externality)
- Best Response - Chinese Restaurant Game
  - Improved utility with similar interference level

Learning in Stochastic System

Stochastic System

- Unknown state may change with time
- Users may arrive and depart stochastically

Learning in Stochastic System

- State tracking
- User behavior and population prediction

Dynamic Chinese Restaurant Game

- Infinite customers and finite tables with same size but different reservation states
- Customers arrive and leave by Poisson process

Dynamic Chinese Restaurant Game

Learning the restaurant state

- Tables may be reserved in advance.
- The reservation may be cancelled anytime.
- How to learn the reservation state according to collected information?

Table selection strategy

- Customers sequentially arrive and leave.
- During one customer's meal time in one table
  - New customers may join this table.
  - Old customers in this table may leave.
- How to choose one table from an expected long-term view?

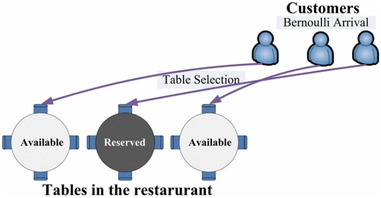
System Model

K tables, each with maximum of N users

- State: $\theta = \{\theta_1, ..., \theta_K\}, \theta_i \in \{0, 1\}$
- $\theta_i = 0 \rightarrow$ the table is unavailable
- State duration: $f_x(t) = \frac{1}{r_x} e^{-t/r_x}$

Customers

- Arrive by Poisson process with rate λ
- Duration of stay follows exponential process with rate μ
- Each receives binary signals conditioning on the current state
  - $\mathbf{s} = \{s_i\} \sim f(s_i|\theta_i)$



Bayesian Learning

Belief: customer's estimation on the current system state

- Each customer reveals her belief to others when entering the restaurant
- New customer learns from the previous shared belief and new signal she receives from the system

$$\mathbf{b}^j = \{b_i^j | b_i^j = Pr(\theta_i = 0 | b_i^{j-1}, b_i^0, s^j, i, f)\}$$

Quantized belief

- Belief is continuous $\rightarrow$ infeasible to store and transmit
- Quanitze b_i^j into M belief levels $\{\mathcal{B}_1, ..., \mathcal{B}_M\}$
  - if $b_i^j \in [\frac{k-1}{M}, \frac{k}{M}]$, $B_i^j = \mathcal{B}_k$
  - Each customer reveals the quantized belief instead
- $B_i^{j-1} \rightarrow b_i^{j-1} \rightarrow$ Bayesian update $\rightarrow b_i^j \rightarrow B_i^j$

Table Selection

Goal: choose the best table

- Maximize the expected utility during whole serving time
- Factors: system state, current and expected grouping

Revisiting Multi-Dimensional Markov Decision Process

- System state $\mathbf{S} = \{\mathbf{B}, \mathbf{G}\}$
  - Belief state $\mathbf{B} = (B_1, B_2, ..., B_K)$
  - Grouping state $\mathbf{G} = (g_1, g_2, ..., g_K)$
- Policy $\pi(\mathbf{S}) \in \mathbf{A} = \{1, 2, .., K\}$
- Immediate reward of table k: $U_k(\mathbf{S})$

How do we define immediate and expected reward with belief state?

Expected Utility of Table k

Immediate utility at table k: $U_k = b_k R_k(g_k)$

- Determined by both belief and grouping state

State transition

- Signal is revealed $\rightarrow$ belief and grouping can be decoupled
- $Pr(\mathbf{S}'|\mathbf{S}) = Pr(\mathbf{B}'|\mathbf{B})Pr(\mathbf{G}'|\mathbf{S})$
- $Pr(\mathbf{B}'|\mathbf{B})$ is calculated by Bayesian learning rule
- $Pr(\mathbf{G}'|\mathbf{S})$ is determined by the policy π

$$Pr(\mathbf{G}'|\mathbf{S}, k, \pi) =$$

$$\begin{cases} \lambda, & \text{if } \pi(\mathbf{S}) = k', \mathbf{g}' = (g_1, .., g_{k'} + 1, ...), \\ g_{k'}\mu, & \text{if } \mathbf{g}' = (g_1, .., g_{k'} - 1, ...), k' \neq k, \\ g_k\mu, & \text{if } \mathbf{g}' = (g_1, ..., g_k - 1, ...), \\ 1 - \lambda - (\sum_{k=1}^{K} g_k - 1), & \text{if } \pi(\mathbf{S}) > 0, \mathbf{G}' = \mathbf{G}, \\ 1 - (\sum_{k=1}^{K} g_k - 1), & \text{if } \pi(\mathbf{S}) = 0, \mathbf{G}' = \mathbf{G}, \\ 0, & \text{otherwise} \end{cases}$$

Finding Equilibrium in Dynamic Chinese Restaurant Game

Equilibrium Condition

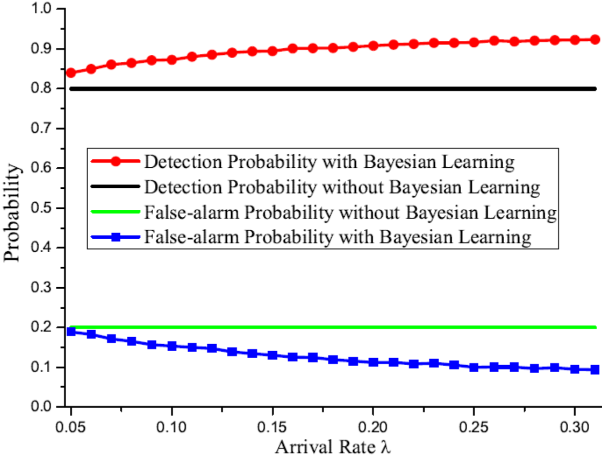
$$W_k^*(\mathbf{S}) = U_k(\mathbf{S}) + (1-\mu)\sum Pr(\mathbf{S}'|\mathbf{S},k)W_k^*(\mathbf{S}') \qquad (3)$$

$$\pi^*(\mathbf{S}) = \arg\max_k W_k^*(\mathbf{S}) \qquad (4)$$

Nash equilibrium can be found through value-iteration algorithm

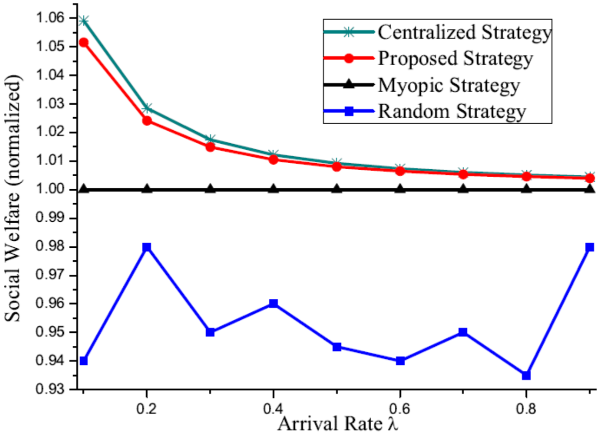
1. Initialize π, update transition probability $Pr(\mathbf{S}'|\mathbf{S},\pi)$
2. Update expected reward W_k with (3)
3. Update π with (4)
4. Repeat 2 and 3 until π converges

App: Cooperative Sensing in Stochastic Network



Higher detection accuracy with Bayesian learning

App: Cooperative Sensing in Stochastic Network



Higher average utility and social welfare with proposed Dynamic CRG strategy

Cons of Signals

The information revealed by agents could be user-generated contents or passively-revealed information.

User-generated contents → Signals

Public reviews, comments and ratings on certain restaurants

- Signals generated by the system and reported by the agent
- Could be untrustworthy when agents have selfish interests
    - A local customer may know the best restaurants in town, but he/she may choose to promote other restaurants in fear that the restaurants may become too popular.
    - Some restaurants will invite popular bloggers or critics to provide positive reviews or rating on the website.

Revealed Actions

Passively-Revealed Actions

Number of subscribers, sold amount, customers waiting in line...

- (Explicitly) related to not only the systematic parameters, but also the intentions of these agents

- A high number of visits may suggest
  - A high-quality service
  - A bad service with a short-term promotion
  - The shutdown of all other restaurants.
- Usually costs more to cheat for the agent she must select a sub-optimal action to reveal a forged information
  - Potentially more trustworthy if we can correctly understand the logic behind their actions.

Learning from Observed Actions

(Dynamic) Chinese Restaurant Game

- Learning from user-generated contents (signals) or previous belief by other agents
- Truth-telling issue
- Infinite observation space

Propose: Hidden Chinese Restaurant Game

- Utilize observed actions instead of signals as information source
- Allow customers to observe the actions of other customers within a limited observation space
- Extract the hidden information from the observed actions

Hidden Chinese Restaurant Game

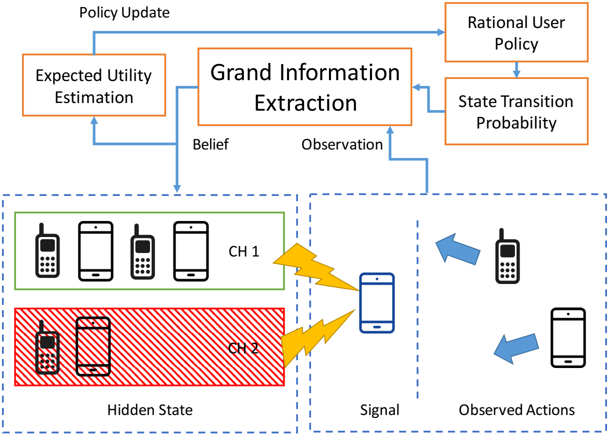


Figure: Hidden Chinese Restaurant Game Framework

System Model

A restaurant with M tables in a state $\theta \in \Theta$, allows at most N customers

- Customers arrival process: Poisson with rate λ
- Customers departure process: Exponential with rate μ
- At each arrival, the customer requests for a seat $x[t] \in \{0, 1, ..., M\}$
- She may not know the number of customers at each table

No new customer enters the restaurant ($x[t] = 0$) when

- The restaurant is full
- No customer arrives
- Customers arrive but choose not to enter
- Existing customer cannot distinguish these events

Customers

Naive Customers

- Actions are predetermined, not necessary related to utility
- The legacy agents or devices whose actions are fixed without the strategic decision making capability

Rational Customers

- Select the tables that maximize their expected utility
- Immediate utility:$u(R_x(\theta), n_x[t])$,
    - $R_x(\theta)$ is the size of table x
    - $n_x[t]$ is the grouping at time t

Observable Information

Private signal

- Each customer receives exactly one signal $s \in \mathcal{S} \sim f(s|\theta)$

Grouping Information

The current grouping $\mathbf{n}[t] = \{n_1[t], ..., n_M[t]\}$ of customers .

- The collective actions of all the previous customers
  - The number of customers waiting at each restaurant
  - The number of customers subscribing to each cellular service

History Information

The history of actions revealed at time $t - H, t - H + 1, ..., t - 1$

- The influences of the former actions to the later customers
- H reflects the limited observation capability

Hidden Chinese Restaurant Game

A Stochastic game with undeterministic number of players

- Player: customers
- Action: $x[t] \in \{0, 1, ..., M\}$
- Utility: $\sum_t u(R_x(\theta), n_x[t])$

Restaurant state θ and externality $n_x[t]$ are the keys

State: The current situation of the system

$$\mathbf{I}[t] = \{\mathbf{n}[t], \mathbf{h}[t], s[t], \theta\}.$$

The information in the state $\mathbf{I}$ is differentiated into two types: observed state $\mathbf{I}^o$ and hidden state $\mathbf{I}^h$.

- The players make decisions based on observed state $\mathbf{I}^o$, while the utility is determined by the whole state $\mathbf{I}$.

Policy

A policy describes the table selection strategy a customer applies in H-CRG given the information she observed.

$$\pi(\mathbf{I}^o) \in \mathbf{A} = \{0, 1, ..., M\}, \forall \mathbf{I}^o.$$

Rational customers: seek to maximize their expected utility

$$\pi^r(\mathbf{I}^o) = \arg \max_{x \in \{0,1,...,M\}} E[U(x)|\mathbf{I}^o], \forall \mathbf{I}^o.$$

where

$$E[U(x)|\mathbf{I}^o[t_a]] =$$
$$\sum_{t=t_a}^{\infty} (1-\mu)^{(t-t_a)} \sum_{\theta \in \Theta} Pr(\theta|\mathbf{I}^o[t_a]) E[u(R_x(\theta), n_x[t])|\mathbf{I}^o[t_a], \theta].$$

Need to estimate the hidden state $\mathbf{I}^h$ from the observed state $\mathbf{I}^o$

State Transition

Possible events to trigger state transition:

- New customer arrives
- Existing customer leaves
- No changes (or zero observable actions)

$$Pr(\mathbf{I}[t+1]|\mathbf{I}[t],\pi^n,\pi^r) =$$

$$\begin{cases} \rho\lambda f(s[t+1]|\theta), & \mathbf{I}[t+1] \in \mathcal{I}^a_{\mathbf{I}[t],\pi^r}; \\ (1-\rho)\lambda f(s[t+1]|\theta), & \mathbf{I}[t+1] \in \mathcal{I}^a_{\mathbf{I}[t],\pi^n}; \\ (n_j[t])\mu f(s[t+1]|\theta), & \mathbf{I}[t+1] \in \mathcal{I}^d_{\mathbf{I}[t]}, n_j[t+1] = n_j[t]-1; \\ (1-\mu\sum_{j=1}^M n_j - \lambda)f(s[t+1]|\theta), & \mathbf{I}[t+1] \in \mathcal{I}^u_{\mathbf{I}[t]}, \mathcal{I}^a_{\mathbf{I}[t],\pi^r} \neq \emptyset, \mathcal{I}^a_{\mathbf{I}[t],\pi^n} \neq \emptyset; \\ (1-\mu\sum_{j=1}^M n_j - \rho\lambda)f(s[t+1]|\theta), & \mathbf{I}[t+1] \in \mathcal{I}^u_{\mathbf{I}[t]}, \mathcal{I}^a_{\mathbf{I}[t],\pi^r} \neq \emptyset, \mathcal{I}^a_{\mathbf{I}[t],\pi^n} = \emptyset; \\ |-|(1-\mu\sum_{j=1}^M n_j - (1-\rho)\lambda)f(s[t+1]|\theta), & \mathbf{I}[t+1] \in \mathcal{I}^u_{\mathbf{I}[t]}, \mathcal{I}^a_{\mathbf{I}[t],\pi^r} = \emptyset, \mathcal{I}^a_{\mathbf{I}[t],\pi^n} \neq \emptyset; \\ (1-\mu\sum_{j=1}^M n_j)f(s[t+1]|\theta), & \mathbf{I}[t+1] \in \mathcal{I}^u_{\mathbf{I}[t]}, \mathcal{I}^a_{\mathbf{I}[t],\pi^r} = \mathcal{I}^a_{\mathbf{I}[t],\pi^n} = \emptyset; \\ 0, & \text{else.} \end{cases}$$

State transitions may not be observable

Belief in Hidden Chinese Restaurant Game

Belief: the probability distribution of the state $\mathbf{I}$ based on the observed state $\mathbf{I}^o$

$$g_{\mathbf{I}|\mathbf{I}^o} = Pr(\mathbf{I}|\mathbf{I}^o).$$

Previous signal and belief are not revealed publicly $\rightarrow$ need to extract belief from observed actions

Grand Information Extraction

$$g_{\mathbf{I}|\mathbf{I}^o} = \sum_{k \in \Theta} Pr(\mathbf{I}|\mathbf{I}^o, \theta = k, \pi^n, \pi^r) Pr(\theta = k|\mathbf{I}^o, \pi^n, \pi^r).$$

How do we get each part?

Grand Information Extraction: Belief on Restaurant State θ

The stationary state distribution given θ and policy π is:

$$\left[Pr(\mathbf{I}|\theta = k, \pi^n, \pi^r)\right] = \left[Pr(\mathbf{I}'|\mathbf{I}, \theta = k, \pi^n, \pi^r)\right]\left[Pr(\mathbf{I}|\theta = k, \pi^n, \pi^r)\right]$$

We can also derive the probability of observed state $\mathbf{I}^o$

$$Pr(\mathbf{I}^o|\theta = k, \pi^n, \pi^r) = \sum_{\mathbf{I} \in \mathcal{I}^o_{\mathbf{I}^o}} Pr(\mathbf{I}|\theta = k, \pi^n, \pi^r).$$

Bayesian rule helps us derive the restaurant state conditioning on the observed state $\mathbf{I}^o$.

Belief on Restaurant State θ given observed state $\mathbf{I}^o$

$$Pr(\theta = k|\mathbf{I}^o, \pi^n, \pi^r) = \frac{Pr(\mathbf{I}^o|\theta = k, \pi^n, \pi^r)Pr(\theta = k)}{\sum_{k' \in \Theta} Pr(\mathbf{I}^o|\theta = k', \pi^n, \pi^r)Pr(\theta = k')}.$$

Grand Information Extraction: True State $\mathbf{I}$

We can further derive the probability of state $\mathbf{I}$ conditioning on the observed state $\mathbf{I}^o$ and $\theta = k$ by

$$Pr(\mathbf{I}|\mathbf{I}^o, \theta = k, \pi^n, \pi^r) = \frac{Pr(\mathbf{I}|\theta = k, \pi^n, \pi^r)}{\sum_{\mathbf{I}' \in \mathcal{I}^o_{\mathbf{I}^o}} Pr(\mathbf{I}'|\theta = k, \pi^n, \pi^r)}.$$

Finally, the belief is given by

$$g_{\mathbf{I}|\mathbf{I}^o} = \sum_{k \in \Theta} Pr(\mathbf{I}|\mathbf{I}^o, \theta = k, \pi^n, \pi^r) Pr(\theta = k|\mathbf{I}^o, \pi^n, \pi^r).$$

No need for a separate belief state

- Reduce state complexity
- No information loss due to belief quantization

Equilibrium Conditions

Nash Equilibrium

The Nash equilibrium of H-CRG is $\pi^*(\mathbf{I}^o)$ if

$$W^{I*}(\mathbf{I}, x, \pi^*) = R(\mathbf{I}, x) + (1-\mu)\sum_{\mathbf{I}'} Pr(\mathbf{I}'|\mathbf{I}, \pi^n, \pi^*, x) W^{I*}(\mathbf{I}', x, \pi^*),$$

$$W^*(\mathbf{I}^o, x) = \sum_{\mathbf{I}\in\mathcal{I}^o_{\mathbf{I}^o}} g_{\mathbf{I}|\mathbf{I}^o,\pi^n,\pi^*} W^I(\mathbf{I}, x),$$

$$\pi^*(\mathbf{I}^o) = \arg\max_x \sum_{\mathbf{I}'^o} Pr(\mathbf{I}'^o|\mathbf{I}^o, \pi^*, x) W^*(\mathbf{I}'^o, x),$$

for all $\mathbf{I}$, $\mathbf{I}^o$, $x \in \{1, 2, ..., M\}$.

An additional step to calculate the expected utility conditioning on the observed state $\mathbf{I}^o$ based on the belief on hidden state

Solutions: Modified value-iteration algorithm

Algorithm 1 Value-Iteration Algorithm for Nash Equilibrium

Initialize π^r, W, W^I;
while 1 **do**
    Update $g_{\mathbf{I}|\mathbf{I}^o,\pi^n,\pi^r}$
    for all $\mathbf{I}^o$ **do**
        Update $\pi^{r'}$, $W^{I'}$, then ; W';
    end for
    $W^d \leftarrow W' - W$
    if $\max W^d - \min W^d < \epsilon$ **then**
        Break
    else
        $W \leftarrow W'$, $W^I \leftarrow W^{I'}$, $\pi^r \leftarrow \pi^{r'}$
    end if
end while
Output π^r, W, and W^I

App: Cooperative Sensing without Signal Exchanges

Cooperative Sensing in CR Networks (S-CRG and D-CRG)

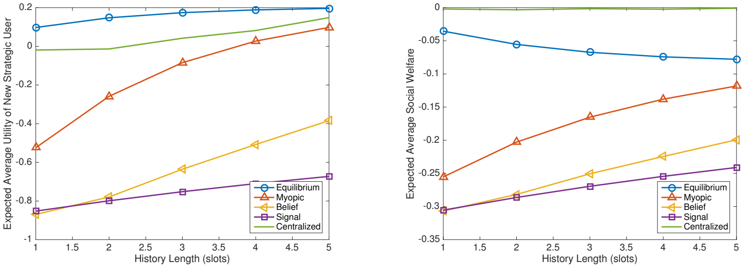
- Aggregate sensing results $\rightarrow$ require control channel

Proposed H-CRG Approach

- Secondary users detect not only the activity of primary users but also the access attempts of other secondary users
  1. A secondary user will first wait for few slots and detect the access attempts of other secondary users in the channels.
  2. Then, it will detect the activity of primary user through traditional channel sensing
  3. Finally, it decides whether to access and which channel to access
- No need for control channel

Simulation Results

Two channels, 8 users, sensing accuracy = 0.85

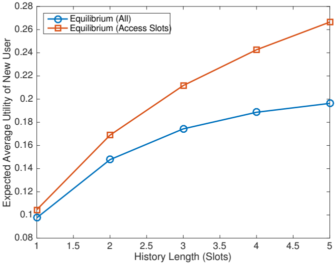


(a) Expected Individual Utility (b) Average Social Welfare

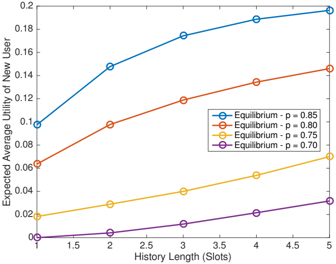
- Longer history length, higher utility
- H-CRG outperforms Centralized solution in terms of new user utility

Simulation Results

Two channels, 8 users, sensing accuracy $= 0.85$



(a) Gain vs. Cost for p $= 0.85$

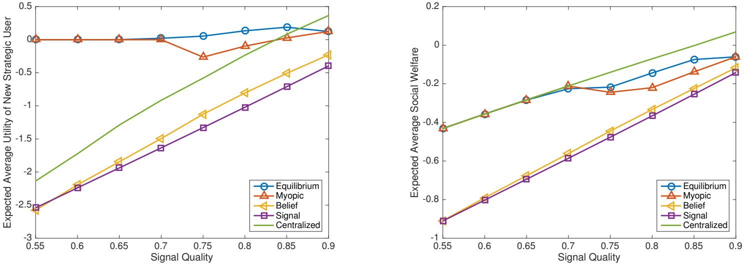


(b) Gain Gained from History Length

- The gain from increased history length diminishes

Simulation Results

Two channels, 8 users, history length = 4

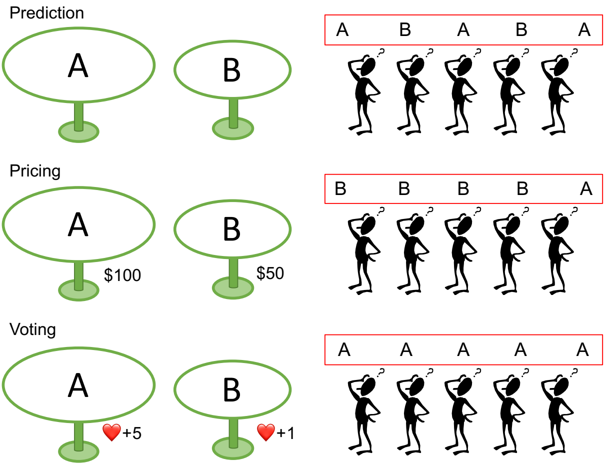


(a) Expected Individual Utility (b) Average Social Welfare

- Performance increases with the signal quality
- H-CRG outperforms others in in terms of new user utility

Outline

Managing Sequential Decision Making



Prediction → Control with pricing → Self-management with voting

Behavior Prediction: Deal Selection on Groupon

Deals on Groupon

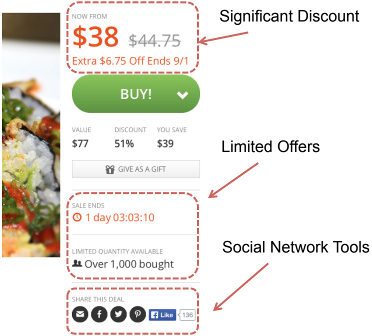
- Significant discount
- Limitation on time/quantity

Learning from External Signals

- Reviews on Internet
- Comments shared by friends
- Rating on Yelp

Network Externality

- Impact on qualify of service



Objective

Optimal deal pricing strategy and deal selection prediction

Groupon - Yelp Dataset

Data Collection

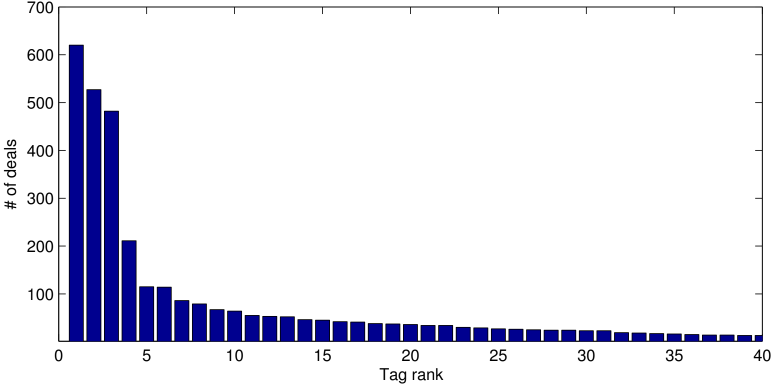
Targets: Groupon deals offered in Washington D.C. area and the corresponding Yelp Records
Duration: Dec. 2012 to July. 2014 (19 months)
Method: RESTful APIs offered by Groupon and Yelp

Groupon : 3 times per day
Yelp : 3 times per day

Dataset Size

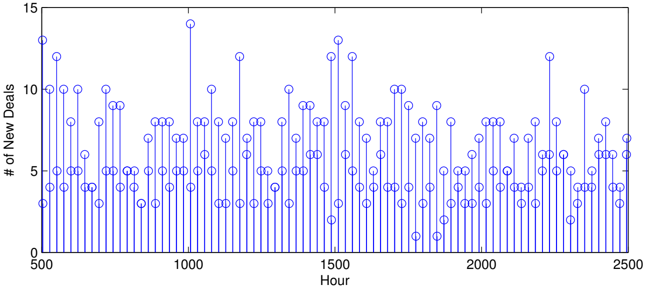
Deals 6509 (2389 with valid Yelp record(s))
Yelp Records 1857 vendors, 24239 new reviews

Tags



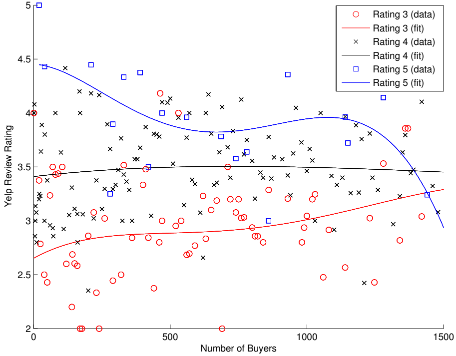
1: Beauty and Spas, 2: Restaurants, 3: Arts and Entertainment

Deal Arrival and Duration



- New deal comes in batches
  - 04:00 AVG 6.73, STD 2.37
  - 05:00 AVG 4.72, STD 2.08
- Deal online duration: AVG 9.21, STD 20.14

Network Externality in Deal Valuation (Restaurant)



- Externality effect depends on the original rating of the vendor
- Nonlinear, not simple positive or negative

Data-Driven System Model

Users

- Arrive exponentially
- Buy one deal when arrive

Deals

- New deals' arrival follows a batched Poisson process
- Sales end when sold out or expire
  - Users learn the quality from social medias such as Yelp
  - Externality (positive and/or negative)

Solution: Dynamic Chinese Restaurant Game

System Model

Online deal set $\mathcal{D}^t = \{d_1, d_2, ...\} \in \mathcal{D}^{all}$

- New deal from all vendors (one-to-one to deals) $\mathcal{D}^{all}$ follows Batch Poisson arrival distribution λ_d
- Deal goes offline following exponential distribution with μ
- Each deal d has a price of $p_d \rightarrow$ controlled by vendor

Users

- Poisson arrival with λ_u
- Leave after selected deal off-line

State: $\mathbf{s}^t = \{\mathcal{D}^t, \mathbf{n}^t, \mathbf{b}^t\}$

- $\mathbf{n}^t = \{n_d^t | d \in \mathcal{D}^{all}\}$: grouping (approximate, rounded by 100)
- $\mathbf{b}^t = \{\mathbf{b}_d^t | d \in \mathcal{D}^{all}\}$: (Quantized) beliefs on deal's real rating (unknown by users)

Review Process and Bayesian Learning

Reviews in Yelp

Average Rating of the deal d's vendor: $r_d^{avg,t} \in \{1,2,3,4,5\}$
*May not be the real rating r_d of the vendor
Review rating: $w_d \in \{1,2,3,4,5\}$

- Poisson arrival with λ_{w_d}
- Accuracy: $Pr(w_d|r_d)$

Bayesian Learning

When new review w'_d arrives

$$b_{d,X}^{t} = Pr(r_d = X | r_d^{avg,t-1}, \{w_d\}, w'_d)$$
$$= \frac{Pr(\{w'_d\}|r_d = X) b_{d,X}^{t-1}}{\sum_{X'=1}^{5} Pr(\{w'_d\}|r_d = X') b_{d,X'}^{t-1}}$$

Utility: Complex Network Externality

User's Utility Function: $U(d) = U(r_d, n_d^*, p_d)$

- r_d: the original rating of the vendor before the deal
- n_d^*: the number of users purchased deal d before the deal is offline
- $U(d)$ experiences externality from n_d^*
  - Could be positive, negative, or complex

Externality on Rating from Groupon's deal

Depends on the vendor's original rating, category, and price

E.g. for restaurant deals with price $< \$50$:

- $r \leq 3$: Increasing
- $3 < r \leq 4$: slightly concave
- $4 < r$: Generally decreasing

State Transition (Observed by users who choose d)

Three possible events to change the state:

1. A new user arrives (λ_u)
2. A new review w_d on d arrives ($\sum \lambda_{w_d}$)
3. A deal d' arrives or goes offline (λ_d)

State Transition

$$Pr(\mathbf{s}' = \{\mathbf{n}', \mathbf{b}', \mathcal{D}'\} | \mathbf{s}, \pi, d) =$$
$$\begin{cases} \lambda_u \Lambda(\lambda_u), & n'_d = n_d + 1; \\ \lambda_{w_j} \sum_X Pr(w_j | r_j = X) b_{j,X}, & \mathbf{b}'_j \text{ is updated with } w_j; \\ \lambda \prod_{d' \in \delta} \rho_{d'} \prod_{d'' \in \mathcal{D}^{all} \setminus (\delta \cup \mathcal{D})} (1 - \rho_{d''}), & \mathcal{D}' = \mathcal{D} \cup \delta, \mathcal{D} \cap \delta = \emptyset; \\ \mu, & \mathcal{D}' = \mathcal{D} \setminus \{d'\}, d' \neq d; \\ 1 - \lambda_u \Lambda(\lambda_u) - \lambda - \sum_{j \in \mathcal{D}_{all}} \lambda_{w_j} - |D|\mu, & \mathbf{s}' = \mathbf{s}; \\ 0, & \text{else.} \end{cases}$$

Equilibrium Conditions

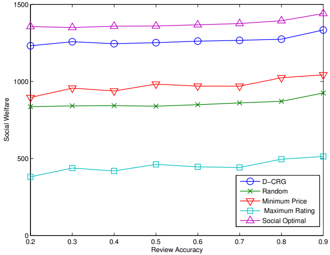
Equilibrium Conditions

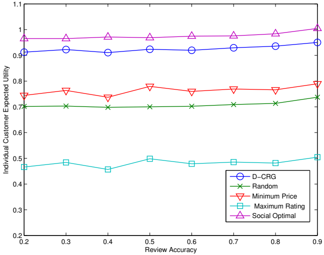
$$W(\mathbf{s}, d) = E[u(d)|\mathbf{s}, \pi] = \mu E[U(r_d, n_d, p_d)|\mathbf{s}]$$
$$+(1-\mu)\sum_{\mathbf{s}' \in \mathcal{S}} \frac{Pr(\mathbf{s}'|\mathbf{s}, \pi, d)}{(1-\mu)} W(\mathbf{s}', d).$$

$$\pi(\mathbf{s}^t) \in \arg\max_{d \in \mathcal{D}^t} W(\mathbf{s}^t, d)$$

Choose the deal that maximizes the expected utility
*The utility is realized when the selected deal is offline (first term)

Simulation Results





- D-CRG performs significantly better than other strategies
  - Considering network externality
  - Rational decisions improve the utilities of the customers
- Some degradation from the social optimal (price of anarchy)

Experiments - Are Customers Rational?

Strategy	Accuracy
Random	0.2777
Maximum Rating	0.2789
Minimum Price	0.3147
Proposed (Myopic)	0.2867
Proposed (Fully Rational)	0.3273

- Customers behave in a rational way
  - But still deviate in some cases
- Rooms to improve their utility / social welfare
  - Deal suggestion, promotion based on better strategies
  - Social optimal solution: selfish customers may refuse to follow
  - D-CRG: incentive compatible to the selfish objectives of customers

Pricing: Video Multicasting Service Subscription

Now we know how users make decisions. Next?

- Can we regulate them?

Heterogeneous video delivery over wireless networks

- Live video streaming
- Internet Protocol TV (IPTV)

Challenge: maintain the quality of service

- Scarce resource in wireless networking
- Heavy loading from heterogeneous demands
  - SD, HD, UHD,...

Solution: Scalable Video Coding Multicasting Service

- More layers received and decoded, better quality
- Using broadcasting characteristic of wireless communication

SVC Multicasting System

Scalable Video Coding

Multiple layers for the same video frame

- More layers received and decoded, better quality
- Layer k can be decoded only when $1 \sim k-1$ are received

One-hop Video Multicasting System

Multiple users request the live broadcasting video(s)

- Users requesting the same video should receive the same data
- Using broadcasting characteristic of wireless communication

The required resource (time/channel/power) to transmit a layer to a group is dominated by the user with worst channel quality

Subscription-based Delivery System

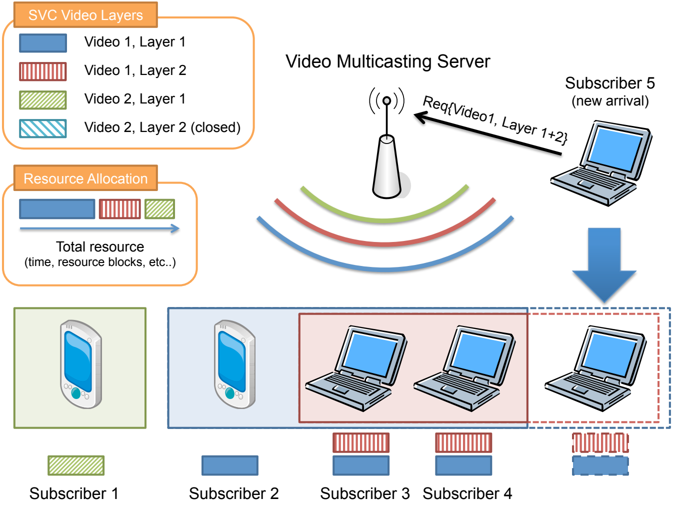
We consider a subscriptions-based system here

- A subscription on the video/layers is required to join the system
- A payment is required to have the subscription

Users with their own preferences on the videos

- Each user requests one of these videos according to their preferences (news, sports, movies,...)
- Receives more layers $\rightarrow$ better quality $\rightarrow$ higher valuation on the service
- Users may have different abilities in decoding the layers $\rightarrow$ different requests in receiving layers

An Illustration of SVC Multicasting System



Objective

Rational demands and economic value of SVC multicasting system

- How do rational users determine their requests to the video/layers?
- How much will these users pay for the service?
- How to optimize the revenue of the system?
- Approach: **Sequential Decision Making**

System Model

SVC Multicasting Server

A video server capable of serving at most N users, providing J videos, each with K layers

Resource Constraints

Total available resource in a time slot: R^{total}

- $R_{j,k}(\underline{g_{j,k}})$: the required resource to transmit layer k of video j to n customers, where the lowest supported quality is $\underline{g_{j,k}}$

Constraints: $R^{total} \geq \sum R_{j,k}(\underline{g_{j,k}})$

This static resource allocation problem is NP-hard.

Subscription System

Stochastic Arrival-Departure Subscribers

- A type $t \in \mathcal{T}$ subscriber has her specific target video(s) $j^t \in \mathcal{J}^t$ and decoding ability (maximum decode-able layer) k^t
  - Baseball games on Smartphone
  - Action Movies on HDTV
- $v^t(j,k)$: a type t user's valuation on video j with maximum layer k

$$v^t(j,k) = \begin{cases} v_j(k), & j \in \mathcal{J}^t,\ k \leq k^t; \\ v_j(k^t), & j \in \mathcal{J}^t,\ k > k^t; \\ 0, & \text{else.} \end{cases}$$

Payment for subscriptions

System state $s = \{n_{j,k}\}$, where $n_{j,k}$ denotes the number of users subscribing video j layer k

One-time charge

- A payment $P^{e}_{j,k}(\mathbf{s}^{a})$ is charged as soon as the user's subscription (j, k) is accepted

Per-slot charge

- Per-slot charge: At each time slot, as long as the user stays in the system with a valid subscription, he is charged with a price of $P_{j,k}(\mathbf{s}^{t})$

Subscription Game

Players: service provider and subscribers

- Service provider determines the **service price** that maximizes the expected revenue
- Rational subscribers maximize their own expected utilities by **choosing the best subscription** (or not to subscribe)

Utility functions

- Service provider: Expected revenue
- Subscribers with type t: Expected utility

$$\mathbb{E}[u^t(j,k)] = -c(\mathbf{s},j,k,0) + \sum_{l=l_a}^{l_d} (\mathbb{E}[v^t(j,\overline{k})|\mathbf{s}=\mathbf{s}^l, \overline{k} \leq k] - c(\mathbf{s}^l,j,k,1))$$

$c(\mathbf{s},j,k,0)$ is the entrance fee and $c(\mathbf{s},j,k,1)$ is the per-slot charge

Equilibrium Finding: subscriber's decisions given the service price
$\rightarrow$ service provider's optimal pricing strategy

Equilibrium Finding

Second Stage: subscribers

- Service prices are given
- Multi-dimensional Markov Decision Process

First Stage: service provider

- The response of subscribers are known
- Revenue Maximizing by average-reward Markov Decision Process

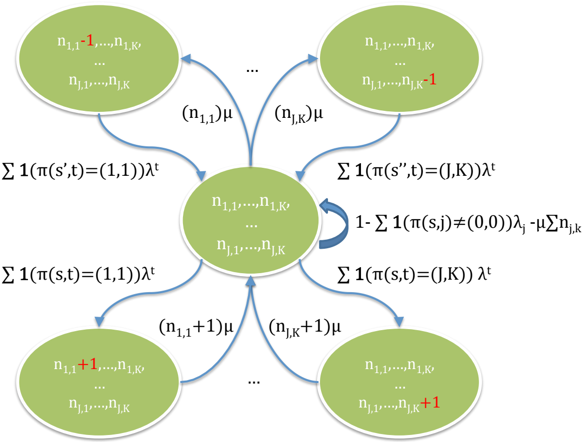
Subscribers: Utility Maximization

Service price $\{P_{j,k}(\mathbf{s})\}$ are given $\rightarrow$ What are the decisions of the subscribers?

Multi-Dimensional Markov Decision Process

- State: $s = \{n^s_{j,k}\} \in \mathcal{S}$,
  - $n_{j,k}$: the number of customers subscribing video j with maximum subscribed layer k
  - Boundary constraints: $\sum_{j,k} n_{j,k} \leq N$
- Action: $a = (j,k) \in \mathcal{A}^t$, which is the type t user's subscription
- Policy: $\pi(s,t) : \mathcal{S} \times \mathcal{T} \mapsto \mathcal{A}^t$
- State Transition Probability : $Pr(\mathbf{s}'|\mathbf{s}, \pi)$
- Immediate Reward: $R(s,j,k) = \mathbb{E}[U(j,k)|s]$

An Illustration on State Transition



Equilibrium Conditions for Subscribers

Given the service price $c(\mathbf{s}, j, k, \cdot)$, the resulting equilibrium is described by

Equilibrium Condition

$$W(\mathbf{s}, j, k) = R(\mathbf{s}, j, k) + (1-\mu) \sum Pr(\mathbf{s}'|\mathbf{s}, \pi, j, k) W(\mathbf{s}', j, k) \quad (5)$$

$$\pi(\mathbf{s}, t) \in \arg \max_{(j,k)\in \mathcal{A}^t} W(\mathbf{s} + \mathbf{e}_{j,k}, j, k) \quad (6)$$

The equilibrium conditions fully describe the rational choices of subscribers under the service price.

Service Provider: Revenue Maximization

Optimal Pricing for Average Revenue Maximization

$$\max_{\{P_{j,k}(\mathbf{s})\},\pi} \lim_{N\to\infty} \frac{1}{N}\mathbb{E}[\sum_{l=1}^{N} Q(\mathbf{s}^l)] = \sum_{\mathbf{s}\in\mathcal{S}} Pr(\mathbf{s}|\pi) \sum_{j,k} n^s_{j,k} P_{j,k}(\mathbf{s}),$$

where

$W(\mathbf{s}+\mathbf{e}_{\pi(\mathbf{s},t)}, \pi(\mathbf{s},t)) \geq 0, \ \forall \mathbf{s}, t,$

$W(\mathbf{s}, j, k) = R(\mathbf{s}, j, k) + (1-\mu)\sum Pr(\mathbf{s}'|\mathbf{s}, \pi, j, k) W(\mathbf{s}', j, k)$

$\pi(\mathbf{s}, t) \in \arg\max_{(j,k)\in\mathcal{A}^t} W(\mathbf{s}+\mathbf{e}_{j,k}, j, k)$

Pricing Strategy and Revenue-Maximized Policy

- Price $\{P_{j,k}(\mathbf{s})\} \Longleftrightarrow$ Policy π
- An average-reward Markov Decision Process with **dynamic immediate reward** $\rightarrow$ Dynamic iterative-update algorithm

Optimal Pricing in One-time Charge Scheme

Assuming the policy π is given, what is the optimal price?

$$\max_{\{P^e_{j,k}(\mathbf{s})\}} Pr(\mathbf{s}|\pi) \sum_{t \in \mathcal{T}, \pi(\mathbf{s},t) \neq (0,0)} \lambda^t P^e_{\pi(\mathbf{s},t)}(\mathbf{s}),$$

subject to

$$\pi(\mathbf{s}, t) \in \arg\max_{(j,k) \in \mathcal{A}^t} W(\mathbf{s} + \mathbf{e}_{j,k}, j, k) - P^e_{j,k}(\mathbf{s})$$

Optimal pricing

$$\forall \mathbf{s}, t, \;\; N(\mathbf{s}) < N, \;\; P^e_{\pi(\mathbf{s},t)}(\mathbf{s}) = W(\mathbf{s} + \mathbf{e}_{\pi(\mathbf{s},t)}, \pi^*(\mathbf{s}, t)).$$

Optimal Pricing in Per-slot Charge Scheme

$$\max_{\mathcal{P}} \sum_{\mathbf{s}\in\mathcal{S}} Pr(\mathbf{s}|\pi) \sum_{j\in\mathcal{J},k\in\mathcal{K}} n^{s}_{j,k} P_{j,k}(\mathbf{s})$$

subject to

$$\begin{aligned}
&(I-(1-\mu)\mathcal{P}_t(\pi^s))^{-1}(\mathcal{V}-\mathcal{P}) = \mathcal{W},\\
&W(\mathbf{s}+\mathbf{e}_{\pi(\mathbf{s},t)}, \pi(\mathbf{s},t)) \geq 0,\ \forall\pi(\mathbf{s},t) \neq (0,0)\\
&W(\mathbf{s}+\mathbf{e}_{\pi(\mathbf{s},t)}), \pi(\mathbf{s},t)) - W(\mathbf{s}+\mathbf{e}_{j,k}, j, k) \geq 0,\\
&\forall\pi(\mathbf{s},t) \neq (0,0), (j,k) \in \mathcal{A}^t\\
&W(\mathbf{s}+\mathbf{e}_{\pi(\mathbf{s},t)}, \pi(\mathbf{s},t)) \leq 0,\\
&\forall\pi(\mathbf{s},t) = (0,0), (j,k) \in \mathcal{A}^t
\end{aligned}$$

which is a linear optimization problem

Revenue Maximization

Assuming the optimal pricing under a given policy π is applied, the immediate expected revenue in state s is given by

$$Q^*(\mathbf{s}, \pi) = \sum_{t \in \mathcal{T}, \pi(\mathbf{s},t) \neq (0,0)} \lambda^t W(\mathbf{s} + \mathbf{e}_{\pi(\mathbf{s},t)}, \pi(\mathbf{s}, t)),$$

which is a function of state **and policy**. Therefore, the original revenue maximization problem can be re-written as

$$\max_{\pi} \lim_{N \to \infty} \frac{1}{N} \sum_{l=1}^{N} \mathcal{P}^{l-1}(\pi) Q^*(\mathbf{s}, \pi),$$

- A average-reward Markov decision process with **dynamic immediate reward**

Revenue Maximization - Reducing to traditional MDP

Theorem

Let $Rev^{one,*}$ and $Rev^{per,*}$ be the optimal revenue of the proposed system under one-time charge and per-slot charge schemes. Then, $Rev^{one,*} = Rev^{per,*}$.

Optimal pricing in Per-slot charge scheme

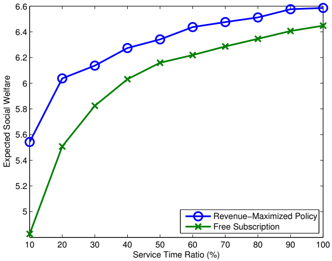
$$P^*_{j,k}(\mathbf{s}) = V_{j,k}(\mathbf{s}), \ \forall \mathbf{s} \in \mathcal{S}, j \in \mathcal{J}, k \in \mathcal{K}.$$

The per-state expected revenue becomes

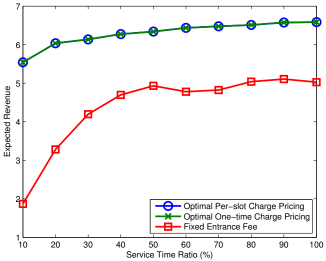
$$Q^*(\mathbf{s}) = \sum_{j,k} n^s_{j,k} P^*_{j,k}(\mathbf{s}) = \sum_{j,k} V_{j,k}(\mathbf{s})$$

The revenue maximization problem becomes a MDP with immediate reward function $Q^*(\mathbf{s})$ independent from the policy, which can be solved easily.

System Performance v.s. Service Time Ratio



(a) Average Social Welfare



(b) Average Revenue

- The rationality from users indeed reduce the system efficiency
- The revenue is highest under the optimal pricing strategies

Social Computing: Answering vs. Voting

Social Computing Applications

- Stack Overflow
- Reddit

Two forms of actions

- Creating piece of content (answer)
- Rating existing content (vote)

The answering-voting externality

- Sequential actions
- The utility of answering depends on the vote of future users

Goal: analyze sequential user behavior under the presence of answering-voting externality

System Model

Users

- A countable infinite set of users who act sequentially
- Each user has a randomly drawn tyep $\sigma = (\sigma_A, \sigma_V)$
  - $\sigma_A \in [0, 1]$: user's ability in answering a question
  - $\sigma_V \in [V_{max}, V_{min}]$: user's preference toward voting

States

- m: number of answers received
- $c(m)$: cost to create a new answer given the received answers

Actions

- A: answer the question with quality q
- V: vote on a solution with quality q' (up with prob. q' and down with $1 - q'$)
  - Answers gets $R_u > 0$ if voted up and $R_d < 0$ if voted down
- N: do nothing

Sequential Decision Making Game

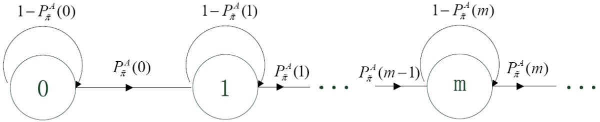
Utility

$$u(m,\sigma,\theta,\pi) = \begin{cases} -c(m) + \delta_{g_\pi}(m+1,\sigma_A), & \text{if } \theta = A; \\ \sigma_V + R_V - C_V, & \text{if } \theta = V; \\ 0, & \text{if } \theta = N; \end{cases}$$

Long-term expected utility one can get from answering the question, given other's answering and voting strategy

$$g_\pi(m,q) = \frac{P_\pi^V(m)}{m}[(R_u+R_d)q-R_d]+\delta[P_\pi^A(m)g_{H_0}(m+1,q)+(1-P_\pi^A(m))g_\pi(m,q)]$$

State Transitions

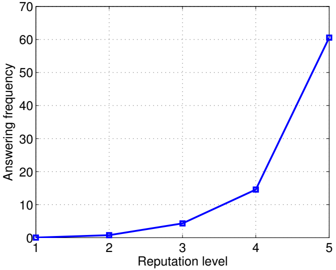


Equilibrium

There exists a pure strategy and unique equilibrium that has a threshold structure in each state

$$\theta^* = \begin{cases} A, & \text{if } \sigma_A > \hat{a}(m, \sigma_V); \\ V, & \text{if } \sigma_A \leq \hat{a}(m, \sigma_V) \text{ and } \sigma_V \leq \hat{\sigma}_V \text{ and } m \leq 1; \\ N, & \text{otherwise}; \end{cases}$$

A dynamic programming algorithm to obtain the equilibrium



Dataset

Stack Overflow

- Questions that have been posted from 01/01/2013 to 03/31/2013
- Exclude closed questions or those receive no answers
- Include all related users, answers and votes

Statistics

Questions	430,749
Answers	731,679
Votes	1,327,883
Users	136,125 (at least)

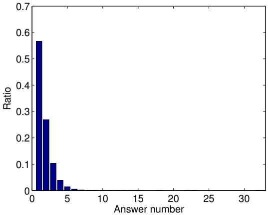
Saturation Phenomenon

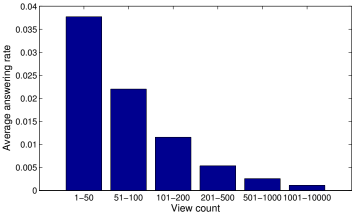
Theoretical results

- After reaching a certain state, no users will have incentive to choose action A
- There always exists an equilibrium

Observations from data

- Left: distribution of answer count
- Right: the average answering rate by different view count intervals



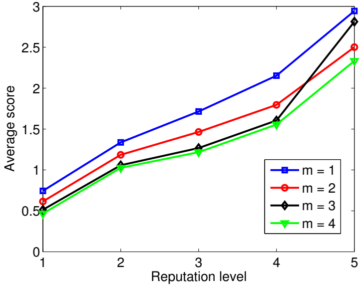


Advantage of Higher Ability

Theoretical results

- The long-term expected reward for answering $g_\pi(m, \sigma_A)$ is strictly increasing in user ability σ_A

Observations from data

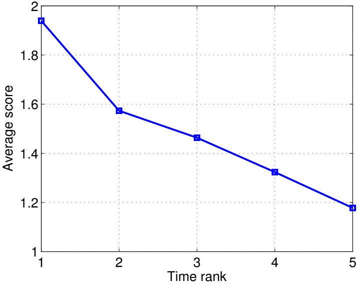


Advantage of Answering Earlier

Theoretical results

- The long-term expected reward is decreasing in m
- The threshold of user ability for answering is increasing in m

Observations from the data



Incentive Mechanism Design

Objective of System Designer

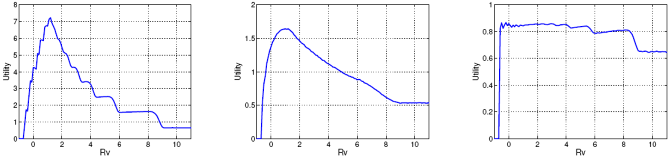
$$U^s = K^{-a} \sum_{k=1}^{K} \beta^{t_k} q_k$$

K: number of answers, t_k, q_k: arrival time and quality of $k-th$ answer

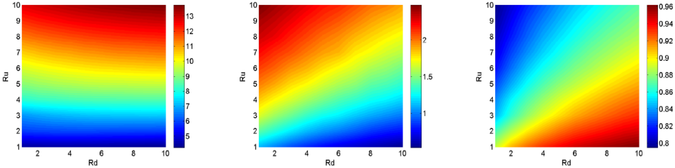
- Use case 1: $\alpha = 0, \beta = 1$
  - Focus on sum of qualities and diversity of answers
- Use case 2: $\alpha = 0, \beta < 1$
  - Time-sensitive and diversity
- Use case 3: $\alpha = 1, \beta = 1$
  - Individual quality
  - Long-lasting values

Design Principles

Principle I: Voting should be encouraged, but not too much

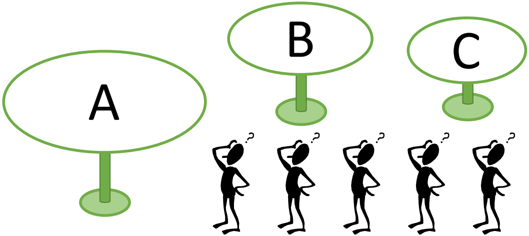


Principle II: Higher reward/punishment ratio → better diversity and timeliness



Outline

What You have Learned



- Nash Game
- Bayesian Game
- Table Selection Problem

Game Theory Basic

- Static System: Influence from Others
- Dynamic System: Predicting the Future

Network Externality

- Acquired: Learning from Signals
- Observed: Learning from Actions

Sequential Learning

- Prediction
- Pricing
- Voting

Management

What You can Do Next?

New Applications

- Fog/Edge Computing
- FinTech / BlockChain
- AI Network

New Challenges

- Scalablility
- Decision Order: Act or Wait?
- Heterogeneous Observation Space

Acknowledgement

- K.J. Ray Liu, UMD
- Chunxiao Jiang, Tsinghua
- Biling Zhang, BUPT
- Yang Gao, Facebook
- Yu-Han Yang, Google

References

- Chih-Yu Wang, Yan Chen, K.J. Ray Liu, "Hidden Chinese Restaurant Game: Grand Information Extraction for Stochastic Network Learning," IEEE Transactions on Signal and Information Processing over Networks, volume 3, number 2, pages 330-345, June 2017.
- Yan Chen, Chunxiao Jiang, Chih-Yu Wang, Yang Gao, K. J. Ray Liu, "Decision Learning: Data Analytic Learning with Strategic Decision Making," IEEE Signal Processing Magazine, volume 33, number 1, pages 37-56, January 2016.
- Biling Zhang, Yan Chen, Chih-Yu Wang, K.J. Ray Liu, "A Chinese Restaurant Game for Learning and Decision Making in Cognitive Radio Networks," Computer Networks, volume 91, pages 117-134, November 2015.
- Yang Gao, Yan Chen, and K.J. Ray Liu, "Understanding Sequential User Behavior in Social Computing: to Answer or to Vote?", IEEE Trans. on Network Science and Engineering, volume 2, number 3, pages 112 - 126, July 2015
- Chih-Yu Wang, Yan Chen, Hung-Yu Wei, K. J. Ray Liu, "Scalable Video Multicasting: A Stochastic Game Approach with Optimal Pricing," IEEE Transactions on Wireless Communications, volume 14, number 5, pages 2353-2367, May 2015.
- Chunxiao Jiang, Yu-Han Yang, Yan Chen, Chih-Yu Wang, K.J. Ray Liu,"Dynamic Chinese Restaurant Game: Theory and Application to Cognitive Radio Networks, IEEE Transactions on Wireless Communications, Vol. 13, Issue 14, pp.1960-1973, Apr. 2014

References

- Yu-Han Yang, Chunxiao Jiang, Yan Chen, Chih-Yu Wang, K.J. Ray Liu, Wireless Access Network Selection Game with Negative Externality, IEEE Transactions on Wireless Communications, Vol. 12, Issue 10, pp. 5048-5060, Oct. 2013
- Chih-Yu Wang, Yan Chen, K.J. Ray Liu, Sequential Chinese Restaurant Game, IEEE Trans. on Signal Processing, Vol. 61, Issue 3, pp. 571-584, Feb. 2013
- Chih-Yu Wang, Yan Chen, K.J. Ray Liu, Chinese Restaurant Game, IEEE Signal Processing Letter, Vol. 19, Issue 12, pp. 898-901, Dec. 2012
- Chih-Yu Wang, Yan Chen, Hung-Yu Wei, and K. J. Ray Liu, Optimal Pricing in Stochastic Scalable Video Coding Multicasting System, IEEE International Conference on Computer Communications (INFOCOM) Mini Conference, Turin, Italy, Apr. 2013
- Chunxiao Jiang, Yan Chen, Yu-Han Yang, Chih-Yu Wang, and K. J. Ray Liu, Dynamic Chinese Restaurant Game in Cognitive Radio Networks, IEEE International Conference on Computer Communications (INFOCOM), Turin, Italy, Apr. 2013
- Biling Chang, Yan Chen, Chih-Yu Wang, and K. J. Ray Liu, Learning and Decision Making with Negative Externality for Opportunistic Spectrum Access, IEEE Global Telecommunications Conference (GLOBECOM), California, U.S.A, Dec. 2012

Sequential Decision Making: A Tutorial

Yan Chen, Chih-Yu Wang

School of Elect. Eng., University of Electronic Science and Technology of China
Research Center for Information Technology Innovation, Academia Sinica