An Information-Theoretic Approach to Model Identification in Interactive Influence Diagrams

Yifeng Zeng
Aalborg University, Denmark

Prashant Doshi
University of Georgia, USA

Speaker

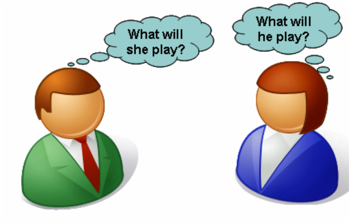
Muthukumaran C.
University of Georgia, USA

Outline

- Problem Statement
- Related Work
- Interactive Influence Diagrams (I - ID)
- Bayesian Model Identification
- Information-Theoretic Model Identification
- Experimental Results

Guess Your Opponent!

- Repeated Games
  - Observe previous actions
  - Predict next actions
  - Win the rewards
- Model Opponent
  - How and What will he/she play?

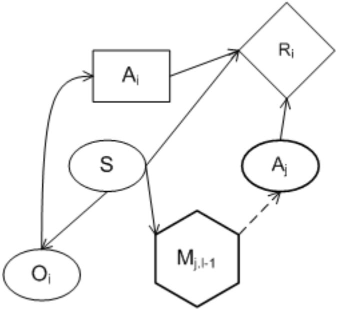


Review

- Carmel&Markovitch(1996)
  - Model agents' strategies using finite state automata
- Suryadi&Gmytrasiewicz(1999)
  - Learn influence diagrams to be consistent with observations
- Saha *et al.*(2005)
  - Approximate agents' decision functions using Chebyshev polynomials

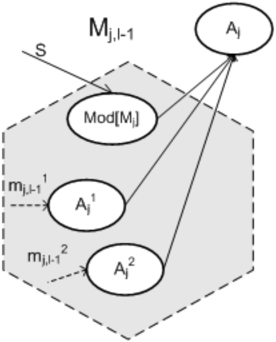
Interactive Influence Diagram (I - ID, Doshi *et al.* 2007)

- A generic level l Interactive-ID (I-ID) for agent i situated with one other agent j
  - **Model Node:** $M_{j,l-1}$
    - Models of agent j at level $l-1$
  - **Policy link:** dashed line
    - Distribution over agent j's actions given its models
  - **Beliefs on** $M_{j,l-1}$**:** $P(M_{j,l-1}|s)$
    - Be updated over time



Details of the Model Node

- Members of the model node
  - Different chance nodes: solutions of models $m_{j,l-1}$
  - $Mod[M_j]$ represents the different models of agent j
- CPT of the chance node A_j is a multiplexer
  - Assumes the distribution of each of the action nodes (A_j^1, A_j^2) depending on the value of $Mod[M_j]$



Public Good (PG) Game

- There are two agents initially endowed with X_T amount of resources. Each agent may choose: Fully Contribute (*FC*), Partially Contribute (*PC*) the resources to a public pot, or not contribute (*D*: called defect here)

Public Good (PG) Game

- There are two agents initially endowed with X_T amount of resources. Each agent may choose: Fully Contribute (*FC*), Partially Contribute (*PC*) the resources to a public pot, or not contribute (*D*: called defect here)
- The value of resources in the public pot is discounted by c_i ($\leq$1) for each agent i, where c_i is the marginal private return

Public Good (PG) Game

- There are two agents initially endowed with X_T amount of resources. Each agent may choose: Fully Contribute (*FC*), Partially Contribute (*PC*) the resources to a public pot, or not contribute (*D*: called defect here)
- The value of resources in the public pot is discounted by c_i ($\leq$1) for each agent i, where c_i is the marginal private return
- In order to encourage contributions, the contributing agents punish free riders P but incur a small cost c_p for administering the punishment

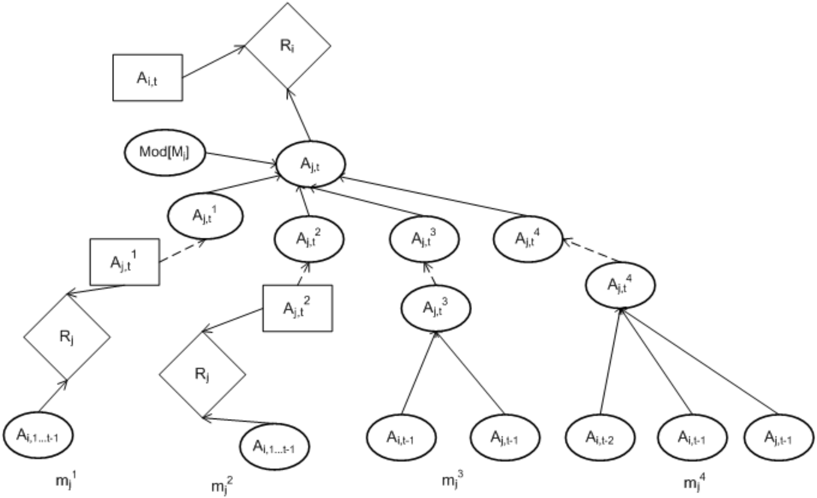
Payoff Matrix

i, j	FC	PC	D
FC	$2c_iX_T$, $2c_jX_T$	$\frac{3}{2}X_Tc_i - \frac{1}{2}c_p$, $\frac{1}{2}X_T + \frac{3}{2}X_Tc_j - \frac{1}{2}P$	$c_iX_T - c_p$, $X_T + c_jX_T - P$
PC	$\frac{1}{2}X_T + \frac{3}{2}X_Tc_i - \frac{1}{2}P$, $\frac{3}{2}X_Tc_j - \frac{1}{2}c_p$	$\frac{1}{2}X_T + c_iX_T$, $\frac{1}{2}X_T + c_jX_T$	$\frac{1}{2}X_T + \frac{1}{2}c_iX_T - \frac{1}{2}P$, $X_T + \frac{1}{2}c_jX_T - P$
D	$X_T + c_iX_T - P$, $c_jX_T - c_p$	$X_T + \frac{1}{2}c_iX_T - P$, $\frac{1}{2}X_T + \frac{1}{2}c_jX_T - \frac{1}{2}P$	X_T, X_T

Table: PG game with punishment. Based on punishment, P, and marginal return, c_i, agents may choose to contribute than defect.

Agent j's Types

- m_j^1: A reciprocal agent who contributes only when it expects the other agent to contribute as well
  - Low values of c_i
- m_j^2: An altruistic agent who prefers to contribute during the play
  - High values of c_i
- m_j^3: Relies on both its own and opponent actions in the previous time step
- m_j^4: Relies more on the past interaction - up to two previous time steps


R_i
A_i,t
Mod[M_j]
A_j,t
A_j,t^1
A_j,t^2
A_j,t^3
A_j,t^4
A_j,t^1
A_j,t^2
A_j,t^3
A_j,t^4
R_j
R_j
A_i,1...t-1
A_i,1...t-1
A_i,t-1
A_j,t-1
A_i,t-2
A_i,t-1
A_j,t-1
m_j^1
m_j^2
m_j^3
m_j^4

Two Cases

- Case 1: $m_j^* \in M_j$ (Traditional)
  - Bayesian Model Identification
- Case 2: $m_j^* \notin M_j$
  - Information-Theoretic Model Identification

Bayesian Learning (Traditional)

$$Pr(m_j^n|o_i^t) = \frac{Pr(o_i^t|m_j^n)Pr(m_j^n|o_{1:t-1})}{\sum_{m_j \in M_j} Pr(o_i^t|m_j)Pr(m_j)} \quad (1)$$

- If an agent's prior belief assigns a non-zero probability to the true model of the other agent, its posterior beliefs updated using Bayesian learning will converge with probability 1
- Don't always converge to the true model of the other agent
  - Observationally equivalent models

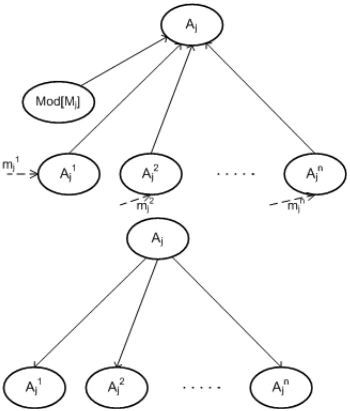
Observational Equivalence

- Two *j*'s Models
  - **Model 1:** Select *FC* for an infinite number of steps, but if at any time *i* chooses *PC*, *j* would also do so at the next time step and then continue selecting *PC*
  - **Model 2:** Play tit-for-tat strategy: *j* performs the action which *i* did in the previous time step
- *i* selects *FC* for an infinite number of times

Relevant Models

- *Relevant* model m_j^n
  - A relevant model predicts an action that is likely to correlate with a particular observed action of the other agent
    - $Pr(a_j^1|m_j^n, a_j^*) \geq Pr(a_j^1|m_j^n, \bar{a}_j^*)$, where $a_j^1 \in OPT(m_j^n)$
  - We interpret the existence of a mutual pattern as evidence that the candidate model shares some behavioral aspects of the true model
- Assign large probabilities to m_j^n in $Mod[M_j]$ over time

Learning Naive Bayesian Models



Time	A_j^1	A_j^2	...	A_j^n	A_j
1	FC	D	...	D	PC
2	D	PC	...	FC	FC
3	FC	PC	...	D	PC
4	FC	FC	...	PC	D
5	D	PC	...	PC	FC
6	PC	FC	...	D	FC
.	...	...	...	...	...

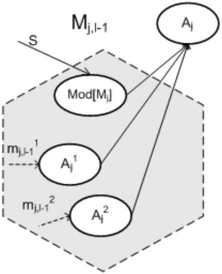
Figure: History of interaction

Mutual Information as Model Weight

$$\begin{aligned} MI(m_j^n, m_j^*) &\stackrel{def}{=} Pr(A_j^n, A_j) log[\frac{Pr(A_j^n, A_j)}{Pr(A_j^n)Pr(A_j)}] \\ &= Pr(A_j^n|A_j)Pr(A_j) log[\frac{Pr(A_j^n|A_j)}{Pr(A_j^n)}] \end{aligned} \quad (2)$$

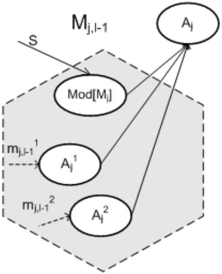
- A_j^n: the chance node mapped from m_j^n
- A_j: the observed actions generated by m_j^*

Model Weight Update



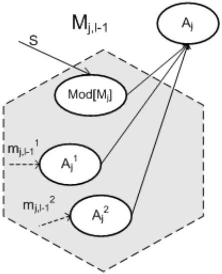
Step 1: Update the training set using *i*'s observations and model m_j^p solutions

Model Weight Update



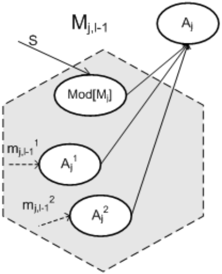
Step 1: Update the training set using *i*'s observations and model m_j^p solutions
Step 2: Learn the parameters of the *naive BN* including the chance nodes $A_j^1, \ldots, A_j^n$, and A_j

Model Weight Update



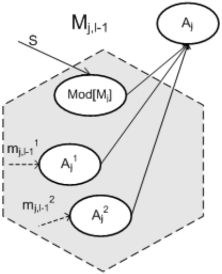
Step 1: Update the training set using *i*'s observations and model m_j^p solutions
Step 2: Learn the parameters of the *naive BN* including the chance nodes $A_j^1, \ldots, A_j^n$, and A_j
Loop

Model Weight Update



Step 1: Update the training set using *i*'s observations and model m_j^p solutions
Step 2: Learn the parameters of the *naive BN* including the chance nodes $A_j^1, \ldots, A_j^n$, and A_j
Loop
Step 3: Compute $MI(m_j^p, m_j^*)$

Model Weight Update

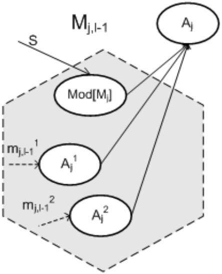


Step 1: Update the training set using i's observations and model m_j^p solutions
Step 2: Learn the parameters of the *naive BN* including the chance nodes $A_j^1, \ldots, A_j^n$, and A_j
Loop

Step 3: Compute $MI(m_j^p, m_j^*)$

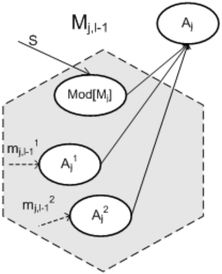
Step 4: Obtain $Pr(A_j|A_j^p)$ from the learned *naive BN*

Model Weight Update



Step 1: Update the training set using *i*'s observations and model m_j^p solutions
Step 2: Learn the parameters of the *naive BN* including the chance nodes $A_j^1, \ldots, A_j^n$, and A_j
Loop
Step 3: Compute $MI(m_j^p, m_j^*)$
Step 4: Obtain $Pr(A_j|A_j^p)$ from the learned *naive BN*
Step 5: Populate CPD row of the chance node A_j using $Pr(A_j|A_j^p, m_j^p)$

Model Weight Update



Step 1: Update the training set using *i*'s observations and model m_j^p solutions

Step 2: Learn the parameters of the *naive BN* including the chance nodes $A_j^1, \ldots, A_j^n$, and A_j

Loop

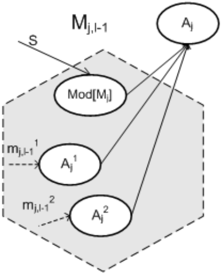
Step 3: Compute $MI(m_j^p, m_j^*)$

Step 4: Obtain $Pr(A_j|A_j^p)$ from the learned *naive BN*

Step 5: Populate CPD row of the chance node A_j using $Pr(A_j|A_j^p, m_j^p)$

Step 6: Normalize $MI(m_j^p, m_j^*)$

Model Weight Update



Step 1: Update the training set using *i*'s observations and model m_j^p solutions
Step 2: Learn the parameters of the *naive BN* including the chance nodes $A_j^1, \ldots, A_j^n$, and A_j
Loop
Step 3: Compute $MI(m_j^p, m_j^*)$
Step 4: Obtain $Pr(A_j|A_j^p)$ from the learned *naive BN*
Step 5: Populate CPD row of the chance node A_j using $Pr(A_j|A_j^p, m_j^p)$
Step 6: Normalize $MI(m_j^p, m_j^*)$
Step 7: Populate CPD of the chance node $Mod[M_j]$ using MI

Some Properties

- **Property 1**
  - Irrelevance: $Pr(a_j|m_j^n, a_j^*) = Pr(a_j|\ m_j^n, \bar{a}_j^*)$
    - $MI(m_j^n, m_j^*) = 0$

Some Properties

- **Property 1**
  - Irrelevance: $Pr(a_j|m_j^n, a_j^*) = Pr(a_j|\ m_j^n, \bar{a}_j^*)$
    - $MI(m_j^n, m_j^*) = 0$
- **Property 2**
  - Relevance Ordering(m_j^n is more relevant than m_j^p):
    $Pr(a_j^1|m_j^n, a_j^*) \geq Pr(a_j^1|m_j^p, a_j^*)$ and
    $Pr(a_j^1|m_j^n, \bar{a}_j^*) \leq Pr(a_j^1|m_j^p, \bar{a}_j^*)$
    - Larger *MI* is assigned to m_j^n: $MI\ (m_j^n, m_j^*) \geq MI\ (m_j^p, m_j^*)$

Some Properties

- **Property 1**
  - Irrelevance: $Pr(a_j|m_j^n, a_j^*) = Pr(a_j|\ m_j^n, \bar{a}_j^*)$
    - $MI(m_j^n, m_j^*) = 0$
- **Property 2**
  - Relevance Ordering(m_j^n is more relevant than m_j^p):
    $Pr(a_j^1|m_j^n, a_j^*) \geq Pr(a_j^1|m_j^p, a_j^*)$ and
    $Pr(a_j^1|m_j^n, \bar{a}_j^*) \leq Pr(a_j^1|m_j^p, \bar{a}_j^*)$
    - Larger MI is assigned to m_j^n: $MI\ (m_j^n, m_j^*) \geq MI\ (m_j^p, m_j^*)$
- **Property 3**
  - Convergence
    - Given that the true model $m_j^* \in M_j$ and is assigned a non-zero probability, the normalized distribution of mutual information of the models converges with probability 1

MI Equivalence

- One example
  - True model: j always plays FC
  - Candidate model: j always plays D
  - Both models are assigned equal MI
    - Dependency is elicited between D and FC
- Set of MI equivalence $\supseteq$ Set of Observational equivalence
- NOT affect prediction performance
  - The perceived dependency classifies D into FC through the learned parameters $Pr(A_j|A_j^p)$

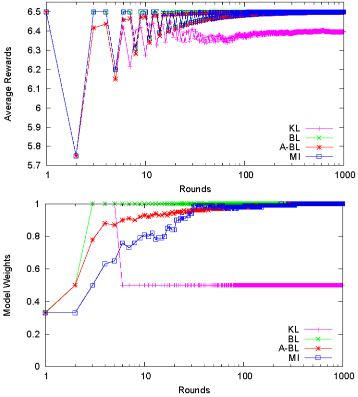
Method Evaluation

- Methods
  - Bayesian Learning (*BL*)
  - Mutual Information (*MI*)
  - Adaptation Bayesian Learning ($A - BL$)
    - Restart the BL process when the likelihoods become zero by assigning candidate models prior weights using the frequency with which the observed action has been predicted by the candidate models so far
  - KL Divergence
    - Measure difference between A_j^n and A_j distributions

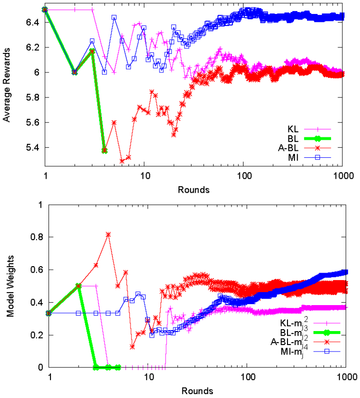
Method Evaluation

- Methods
  - Bayesian Learning (*BL*)
  - Mutual Information (*MI*)
  - Adaptation Bayesian Learning ($A - BL$)
    - Restart the BL process when the likelihoods become zero by assigning candidate models prior weights using the frequency with which the observed action has been predicted by the candidate models so far
  - KL Divergence
    - Measure difference between A_j^n and A_j distributions
- Scenarios
  - PG Games
  - Negotiation Games (4 types of opponents)

Case 1: $m_j^* = m_j^4$, M_j=$\{m_j^1, m_j^3, m_j^4\}$



Case 2: $m_j^* = m_j^1$, M_j=$\{m_j^2, m_j^3, m_j^4\}$



Conclusions

- I-ID in Repeated Games
- Two Cases for Model Identification in I-ID
- MI Complements BL