A graphical model for sequential teams

Aditya Mahajan and Sekhar Tatikonda
Dept of Electrical Engineering
Yale University

Presented at: ConCom Workshop, June 27, 2009
A glimpse of the result
Structural results in sequential teams

- Example: MDP (Markov decision process)
  - Controlled MC: \( \Pr(x_t | x_1, \ldots, x_{t-1}, u_1, \ldots, u_{t-1}) = \Pr(x_t | x_{t-1}, u_{t-1}) \)
  - Controller: \( u_t = g_t(x_1, \ldots, x_t, u_1, \ldots, u_{t-1}) \)
  - Reward: \( r_t = \rho_t(x_t, u_t) \)
  - Objective: Maximize \( \mathbb{E}\left\{ \sum_{t=1}^{T} R_t \right\} \)

- Structural results
  - Without loss of optimality, \( u_t = g_t(x_t) \)
Graphically ... original
Graphically ... structural results
Structural results in sequential teams

○ Example: real-time source coding

▷ Source: First order Markov source \( \{x_t, t = 1, \ldots \} \)
▷ Real-time source coder: \( y_t = c_t(x_1, \ldots, x_t, y_1, \ldots, y_{t-1}) \)
▷ Finite memory decoder: \( \hat{x}_t = g_t(y_t, m_{t-1}) \)
▷ \( m_t = l_t(y_t, m_{t-1}) \)
▷ Cost: \( d_t = \rho_t(x_t, \hat{x}_t) \)


○ Structural Results

▷ Without loss of optimality, \( y_t = c_t(x_t, m_{t-1}) \)
Graphically \ldots original
Graphically . . . structural results
The main idea

- Represent a sequential team as a directed graph
- Simplify the graph
Sequential teams – Salient features

- A team is sequential if and only if there exists a partial order between the system variables.

- There is no loss of optimality in restricting attention to non-randomizing decision makers.

- Data available at a DM can be ignored if it is independent of the future rewards conditioned on other data at the DM.

- Variables functionally determined from the data available at a DM can be assumed to be observed at the DM.
Graphical models – Salient features

- Any partial order gives rise to a DAG (Directed Acyclic Graph)

- A DAFG can be used to efficiently check for conditional independence using d-separation

- A DAFG can be used to efficiently check for conditional independence with deterministic nodes using D-separation
Match between features of sequential teams and graphical models
The rest is a matter of details . . .
The model

- Components of a sequential team

  ▶ A set $N$ of indices of system variables $\{X_n, n \in N\}$.
    Finite sets $\{X_n, n \in N\}$ of state spaces of $X_n$
    - $A \subset N$, variables generated by DM
    - $N \setminus A$, variables generated by nature
    - $R \subset N$, reward variables

  ▶ Information sets $\{I_n, n \in N\}$, such that $I_n \subseteq \{1, \ldots, n\}$. $I_n = \prod_{i \in I_n} X_i$

  ▶ $F_{N \setminus A} = \{f_n, n \in N \setminus A\}$, where $f_n$ is a conditional PMF $X_n$ given $I_n$

  ▶ Design: $G_A = \{g_n, n \in A\}$, where $g_n$ is a decision rule from $I_n$ to $X_n$
The model

- Probability measure induced by a design
  $$P^G_A(X_N) = \prod_{n \in N \setminus A} f_n(X_n|I_n) \prod_{n \in A} I[X_n = g_n(I_n)]$$

- Optimization problem
  
  Minimize $$E\left\{ \sum_{n \in R} X_n \right\}$$, where the expectation is with respect to $$P^G_A$$. 
Representation as a graphical model

○ Directed Acyclic Factor Graph

○ Nodes

▷ Variable node $n \equiv$ system variable $X_n$
▷ Factor node $\tilde{n} \equiv$ conditional PMF $f_n$ or decision rule $g_n$

○ Edges

▷ $(i, \tilde{n})$, for each $n \in N$ and $i \in I_n$
▷ $(\tilde{n}, n)$, for each $n \in N$

○ Acyclic Graph

▷ Sequential team $\Rightarrow$ partial order on variable nodes $\Rightarrow$ acyclic graph
Graphical models – Terminology

- **parents**(n)
  - \( \{m : m \rightarrow n\} \)
  - Parents of a control (factor) node = data observed by controller

- **children**(n)
  - \( \{m : n \rightarrow m\} \)
  - Children of a control node = control action

- **ancestors**(n)
  - \( \{m : \exists \text{ directed path from } m \text{ to } n\} \)
  - Ancestors of a control node = all nodes that affect the data observed

- **descendants**(n)
  - \( \{m : \exists \text{ directed path from } n \text{ to } m\} \)
  - Descendants of a control node = all nodes affected by the control action
Graphical Models – Example
Graphical Models – Variable nodes

Reward nodes

Non-reward nodes
Graphical Models – Factor nodes

Control Factors

Stochastic Factors
Graphical Models — Parents and Children

Parents

Children

Control factor node
Graphical Models – Ancestors and descendants

Ancestors

Descendants

Control factor node
**Structural results**

- The main idea

  If some data available at a DM is independent of future rewards given the control action and other data at the DM, then that data can be ignored.

  Can we automate this process?
Struct. result $\equiv$ cond. independence

Graphical models can easily test conditional independence
Conditional independence

- Three canonical graphs to verify $x \perp z \mid y$

- Blocking of a trail

A trail from $a$ to $b$ is blocked by $C$ if $\exists$ a node $v$ on the trail such that either:

- either $\rightarrow v \rightarrow$, $\leftarrow v \leftarrow$, or $\leftarrow v \rightarrow$, and $v \in C$
- $\rightarrow v \leftarrow$ and neither $v$ nor any of $v$'s descendants are in $C$. 
Conditional independence

- d-separation

  A is d-separated from B by C if all trails from A to B are blocked by C

- Conditional independence

  For any probability measure \( P \) that factorizes according to a DAFG,

  \[ A \text{ d-separated from } B \text{ by } C \implies X_A \text{ is conditionally independent of } X_B \text{ given } X_C, \ P \text{ a.s.} \]

- Efficient algorithms to verify d-separation

  ▶ Moral graph ▶ Bayes Ball
Automated Structural results

○ First attempt

▷ Dependent rewards: \( R_d(\tilde{n}) = R \cap \text{descendants}(\tilde{n}) \)

▷ Irrelevant data: At a control node \( \tilde{n} \), and parent \( i \) is irrelevant if \( R_d(\tilde{n}) \) is \( d \)-separate from \( i \) given parents(\( \tilde{n} \)) \( \cup \) children(\( \tilde{n} \)) \( \setminus \{i\} \)

▷ Requisite data: All parents that are not irrelevant

○ Structural result

▷ Without loss of optimality, we can remove irrelevant data.

\[ u_n = g_n(\text{requisite}(\tilde{n})) \]
Structural Results for MDP – Step 1
Structural Results for MDP – Step 1

Pick node $g_3$.

- Original $u_3 = g_3(x_1, x_2, x_3, u_1, u_2)$
- $\text{requisite}(g_3) = \{x_3\}$
- Thus, $u_3 = g_3(x_3)$
Structural Results for MDP – Step 2
Structural Results for MDP – Step 2

- Pick node \( g_2 \).

  - Original \( u_2 = g_2(x_1,x_2,u_1) \)
  - \( \text{requisite}(g_2) = \{x_2\} \)
  - Thus, \( u_2 = g_2(x_2) \)
Structural Results for MDP — Simplified
\[ u_n = g_n(\text{requisite}(\tilde{n})) \]

Does not work for all problems . . .
even when structural simplification is possible
A real-time source coding problem


Mathematical Model

- Source: First order Markov source \( \{x_t, \ t = 1, \ldots \} \)
- Real-time source coder: \( y_t = c_t(x(1:t), y(1:t-1)) \)
- Finite memory decoder: \( \hat{x}_t = g_t(y_t, m_{t-1}) \)
- \( m_t = l_t(y_t, m_{t-1}) \)
- Cost: \( d_t = \rho_t(x_t, \hat{x}_t) \)
Model for real-time comm – Does not simplify
Need to take care of deterministic variables!
**Functionally determined nodes**

- **Functionally determined**
  - $X_B$ is functionally determined by $X_A$ if $X_B \perp X_N \mid X_A$

- **Conditional independence with functionally determined nodes**
  - Can be checked using **D-separation**
  - Similar to d-sep: in the defn of blocking change “in C” by “is func detm by C”

- **Blocking of a trail (version that takes care of detm nodes)**
  - A trail from $a$ to $b$ is blocked by $C$ if $\exists$ a node $v$ on the trail such that either:
    - either $\to v \to$, $\leftarrow v \leftarrow$, or $\leftarrow v \to$, and $v$ is functionally determined by $C$
    - $\to v \leftarrow$ and neither $v$ nor any of $v$’s descendants are in $C.$
Automated Structural results

- Second attempt
  - Irrelevant data: Change d-separation by D-separation
  - Requisite data: All parents that are not irrelevant

- Structural result
  - Without loss of optimality, we can remove irrelevant data and add appropriate functionally determined data

\[ u_n = g_n(\text{requisite}(\tilde{n}), \text{functionally}_\text{detm}(\tilde{n}) \cap \text{ancestors}(R_d(\tilde{n}))) \]
Let's try this!
Structural Results for Dec MDP – Step 1
Structural Results for Dec MDP – Step 2

d1 d2 d3
f1 ρ1 f2 ρ2 f3 ρ3
x1 y1 ˆx1 m1 x2 y2 ˆx2 m2 x3 y3 ˆx3
c1 g1 l1 c2 g2 l2 c3 g3
Structural Results for Dec MDP – Step 3
Structural Results for Dec MDP – Step 4
Structural Results for Dec MDP – Step 5
Structural Results for Dec MDP – Step 6
Structural Results for Dec MDP – Step 7
Structural Results for Dec MDP – Step 8
Structural Results for Dec MDP – Step 9
Structural Results for Dec MDP – Step 10
Structural Results for Dec MDP – Step 11
Structural Results for Dec MDP – Step 12
Structural Results for Dec MDP – Step 13
Structural Results for Dec MDP – Step 14

d
1
d
2
d
3

f
1
ρ
1

f
2
ρ
2

f
3
ρ
3

x
1
y
1
ˆx
1
m
1

x
2
y
2
ˆx
2
m
2

x
3
y
3
ˆx
3

C:
c
1
G:
g
1
L:
l
1
c
2
G:
g
2
L:
l
2
c
3
G:
g
3
Structural Results for real-time communication

○ Graphically

○ Mathematically

▷ Original Encoder: \( y_t = c_t(x_1, \ldots, x_t, y_1, \ldots, y_{t-1}) \)

▷ New encoder: \( y_t = c_t(x_t, m_{t-1}) \)
Automated Structural results

- Simplify Once
  - For each control node
    - Find irrelevant nodes and functionally determined nodes.
    - Remove edges from irrelevant nodes, add edges from functionally determined nodes.

- Find fixed point
  - Keep on simplifying until the graph does not change

- Software Implementation
  - A EDSL to find structural results
    http://pantheon.yale.edu/~am894/code/teams/
Conclusion
Conclusion

An automated method to derive structural results for sequential teams

○ Future Directions

▷ Belief States
▷ Sequential decomposition
Thank you