TA Evaluations

Johnson, David

Johnson, Jordon

Kazemi, Seyed Mehran

Rahman, MD Abed

Wang, Wenyi

Please Also Complete Teaching Evaluations will close on Sun, June 25th
Decision Theory: Sequential Decisions

Computer Science cpsc322, Lecture 34

(Textbook Chpt 9.3)

June, 22, 2017
“Single” Action vs. Sequence of Actions

Set of primitive decisions that can be treated as a single macro decision to be made before acting.

- Agent makes observations
- Decides on an action
- Carries out the action
Lecture Overview

- Sequential Decisions
  - Representation
  - Policies
- Finding Optimal Policies
Sequential decision problems

• A **sequential decision problem** consists of a sequence of decision variables $D_1, \ldots, D_n$.

• Each $D_i$ has an **information set** of variables $pD_i$, whose value will be known at the time decision $D_i$ is made.

$$pD_3 = \{D_2, V_3, V_4\}$$
Sequential decisions: Simplest possible

- Only one decision! (but different from one-off decisions)
- Early in the morning. I listen to the weather forecast, shall I take my umbrella today? (I’ll have to go for a long walk at noon)
- What is a reasonable decision network?

A. Morning Forecast
   - Weather@12
     - Take Umbrella
   - Take Umbrella

B. Morning Forecast
   - Weather@12
     - Take Umbrella
     - Take Umbrella

C. None of these
Sequential decisions: Simplest possible

- Only one decision! (but different from one-off decisions)
- Early in the morning. Shall I take my **umbrella** today? (I’ll have to go for a **long walk** at noon)
- Relevant Random Variables?

![Diagram showing decision flow from forecast to weather at 12, then to umbrella decision, concluding with umbrella choice.](image)
Policies for Sequential Decision Problem: Intro

- A **policy** specifies what an agent should do under each circumstance (for each decision, consider the parents of the decision node)

In the *Umbrella* “degenerate” case:

```
D_1 ? T F
pD_1
```

- **R**  
- **C**  
- **S**

**One possible Policy**

```
\[
\begin{array}{ccc}
\rightarrow R & T & F & T \\
\rightarrow C & T & F & T \\
\rightarrow S & F & F & T \\
\end{array}
\]
```

- \( \text{dom}(pD) \)
- \( \text{dom}(D) \)
- \( \text{dom} = \text{dom}(D) \)
Sequential decision problems: “complete” Example

- A sequential decision problem consists of a sequence of decision variables $D_1, \cdots, D_n$.
- Each $D_i$ has an information set of variables $\rho D_i$ whose value will be known at the time decision $D_i$ is made.

No-forgetting decision network:
- decisions are totally ordered
- if a decision $D_b$ comes before $D_a$, then
  - $D_b$ is a parent of $D_a$
  - any parent of $D_b$ is a parent of $D_a$
Policies for Sequential Decision Problems

• A policy is a sequence of $\delta_1, \ldots, \delta_n$ decision functions $\delta_i : \text{dom}(pD_i) \to \text{dom}(D_i)$.

• This policy means that when the agent has observed $O \in \text{dom}(pD_i)$, it will do $\delta_i(O)$.

Example:

<table>
<thead>
<tr>
<th>Report</th>
<th>Check Smoke</th>
<th>SeeSmoke</th>
<th>Call</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>true</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>true</td>
<td>true</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>true</td>
<td>false</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>true</td>
<td>false</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>false</td>
<td>true</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>false</td>
<td>true</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>false</td>
<td>false</td>
<td>true</td>
<td>false</td>
</tr>
<tr>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
</tr>
</tbody>
</table>

How many policies?

$2^2 \times 2^8$
Lecture Overview

- Recap
- Sequential Decisions
- Finding Optimal Policies
When does a possible world satisfy a policy?

- A possible world specifies a value for each random variable and each decision variable.

- Possible world $w$ satisfies policy $\delta$, written $w \models \delta$ if the value of each decision variable is the value selected by its decision function in the policy (when applied in $w$).
When does a possible world satisfy a policy?

- Possible world $w$ satisfies policy $\delta$, written $w \models \delta$ if the value of each decision variable is the value selected by its decision function in the policy (when applied in $w$).

$\omega_1$

<table>
<thead>
<tr>
<th>VARs</th>
<th>Report</th>
<th>Check Smoke</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fire</td>
<td>true</td>
<td>false</td>
</tr>
<tr>
<td>Tampering</td>
<td>false</td>
<td>true</td>
</tr>
<tr>
<td>Alarm</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>Leaving</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>Report</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>Smoke</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>SeeSmoke</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>CheckSmoke</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>Call</td>
<td>true</td>
<td>true</td>
</tr>
</tbody>
</table>

$\delta$

<table>
<thead>
<tr>
<th>Report</th>
<th>Check Smoke</th>
<th>SeeSmoke</th>
<th>Call</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>true</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>true</td>
<td>true</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>true</td>
<td>false</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>true</td>
<td>false</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>false</td>
<td>true</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>false</td>
<td>true</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>false</td>
<td>false</td>
<td>true</td>
<td>false</td>
</tr>
<tr>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
</tr>
</tbody>
</table>

A. $\omega_1 \models \delta$
B. $\omega_n \not\models \delta$
C. Cannot tell
Expected Value of a Policy

• Each possible world \( w \) has a probability \( P(w) \) and a utility \( U(w) \).

• The expected utility of policy \( \delta \) is

\[
\sum_{w \in \delta} P(w) \times U(w)
\]

• The optimal policy is one with the max expected utility.
Lecture Overview

- Recap
- Sequential Decisions
- Finding Optimal Policies (Efficiently)
Complexity of finding the optimal policy: how many policies?

• How many assignments to parents?

  \[2^k \leq 2 \leq 2^3\]

• How many decision functions? (binary decisions)

  \[2^2 \cdot 2^3 = \text{product}\]

• How many policies?

If a decision \(D\) has \(k\) binary parents, how many assignments of values to the parents are there?

\[2^k\]

If there are \(b\) possible actions (possible values for \(D\)), how many different decision functions are there?

\[\sqrt{\frac{2^k}{b}}\]

If there are \(d\) decisions, each with \(k\) binary parents and \(b\) possible actions, how many policies are there?

\[\binom{b^{2^k}}{d}\]
Finding the optimal policy more efficiently: VE

1. Create a factor for each conditional probability table and a factor for the utility.

2. **Sum out** random variables that are not parents of a decision node.

3. **Eliminate** (aka sum out) the decision variables

4. **Sum out** the remaining random variables.

5. **Multiply the factors**: this is the expected utility of the optimal policy.
Eliminate the decision Variables: step 3 details

- Select a variable $D$ that corresponds to the latest decision to be made
  - this variable will appear in only one factor with its parents
- Eliminate $D$ by maximizing. This returns:
  - A new factor to use in VE, $\max_D f$
  - The optimal decision function for $D$, $\arg\max_D f$
- Repeat till there are no more decision nodes.

**Example: Eliminate CheckSmoke**

<table>
<thead>
<tr>
<th>Report</th>
<th>CheckSmoke</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>true</td>
<td>-5.0</td>
</tr>
<tr>
<td>true</td>
<td>false</td>
<td>-5.6</td>
</tr>
<tr>
<td>false</td>
<td>true</td>
<td>-23.7</td>
</tr>
<tr>
<td>false</td>
<td>false</td>
<td>-17.5</td>
</tr>
</tbody>
</table>

**New factor**

<table>
<thead>
<tr>
<th>Report</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>-5.0</td>
</tr>
<tr>
<td>false</td>
<td>-17.5</td>
</tr>
</tbody>
</table>

**Decision Function**

<table>
<thead>
<tr>
<th>Report</th>
<th>CheckSmoke</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>false</td>
<td>false</td>
</tr>
</tbody>
</table>
VE elimination reduces complexity of finding the optimal policy

- We have seen that, if a decision \( D \) has \( k \) binary parents, there are \( b \) possible actions. If there are \( d \) decisions, then there are: \( (b^{2^k})^d \) policies.
- Doing variable elimination lets us find the optimal policy after considering only \( d \cdot b^{2^k} \) policies (we eliminate one decision at a time).
- VE is much more efficient than searching through policy space.
- However, this complexity is still doubly-exponential; we’ll only be able to handle relatively small problems.
Generic

Figure 9.8: Decision network for diagnosis

to select what test to apply
and then what treatment to prescribe
Learning Goals for today’s class

You can:

- Represent **sequential decision problems** as decision networks. And explain the **non forgetting property**
- Verify whether a **possible world** satisfies a policy and define the **expected value of a policy**
- Compute the **number of policies** for a decision problem
- Compute the **optimal policy** by Variable Elimination

CPSC 322, Lecture 4

Slide 21
Big Picture: Planning under Uncertainty

- Probability Theory
- Decision Theory
- Markov Decision Processes (MDPs)
  - Fully Observable MDPs
  - Partially Observable MDPs (POMDPs)

One-Off Decisions/
Sequential Decisions

- Decision Support Systems
  (medicine, business, …)
- Economics
- Control Systems
- Robotics

Some Applications

you know you know a little
Cpsc 322 Big Picture

Environment

Deterministic
- Arc Consistency
- Search
- Vars + Constraints
- SLS

Stochastic
- Belief Nets
- Var. Elimination
- Markov Chains

Problem

Constraint Satisfaction

Static

Query

Sequential

Representation

Reasoning Technique

Planning

Search

CPSC 322, Lecture 2  Slide 23
**422 big picture**

<table>
<thead>
<tr>
<th>Deterministic</th>
<th>Stochastic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logics</td>
<td>Belief Nets</td>
</tr>
<tr>
<td>First Order Logics</td>
<td>Approx. : Gibbs</td>
</tr>
<tr>
<td>Ontologies</td>
<td>Markov Chains and HMMs</td>
</tr>
<tr>
<td>Full Resolution</td>
<td>Forward, Viterbi…</td>
</tr>
<tr>
<td>SAT</td>
<td>Approx. : Particle Filtering</td>
</tr>
<tr>
<td></td>
<td>Undirected Graphical Models</td>
</tr>
<tr>
<td></td>
<td>Markov Networks</td>
</tr>
<tr>
<td></td>
<td>Conditional Random Fields</td>
</tr>
<tr>
<td></td>
<td>Markov Decision Processes and Partially Observable MDP</td>
</tr>
<tr>
<td></td>
<td>• Value Iteration</td>
</tr>
<tr>
<td></td>
<td>• Approx. Inference</td>
</tr>
<tr>
<td></td>
<td>Reinforcement Learning</td>
</tr>
</tbody>
</table>

**StarAI (statistical relational AI)**

Hybrid: Det + Sto

- Prob CFG
- Prob Relational Models
- Markov Logics

**Applications of AI**

**Representation**

- Reasoning
- Technique
More AI ..

Machine Learning
Knowledge Acquisition
Preference Elicitation

<table>
<thead>
<tr>
<th>Deterministic</th>
<th>Stochastic</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Logics</strong></td>
<td><strong>Belief Nets</strong></td>
</tr>
<tr>
<td>First Order Logics</td>
<td>Markov Chains and HMMs</td>
</tr>
<tr>
<td>Ontologies</td>
<td></td>
</tr>
</tbody>
</table>

| **Planning**        | **Undirected Graphical Models** |
|                     | Markov Networks              |
|                     | Conditional Random Fields    |
|                     | Markov Decision Processes    |
|                     | Partially Observable MDP     |

| **Query**           | **Reinforcement Learning**   |

Where are the components of our representations coming from?

The probabilities?
The utilities?
The logical formulas?

From people and from data!

StarAI (statistical relational AI)
Hybrid: Det + Sto

Prob CFG
Prob Relational Models
Markov Logics

Machine Learning
Knowledge Acquisition
Preference Elicitation

CPSC 422, Lecture 35
Some of our Grad Courses

522: Artificial Intelligence II: Reasoning and Acting Under Uncertainty

Sample Advanced Topics····
Relational Reinforcement Learning for Agents in Worlds with Objects, relational learning.

- Probabilistic Relational Learning and Inductive Logic Programming at a Global Scale,
Some of our Grad Courses

503: Computational Linguistics I / Natural Language Processing

Sample Advanced Topics...

- Topic Modeling (LDA) – Large Scale Graphical Models
- Discourse Parsing by Deep Learning (Neural Nets)
- Abstractive Summarization
Other AI Grad Courses: check them out

532: Topics in Artificial Intelligence (different courses)
• User–Adaptive Systems and Intelligent Learning Environments
• Foundations of Multiagent Systems

540: Machine Learning

505: Image Understanding I: Image Analysis
525: Image Understanding II: Scene Analysis

515: Computational Robotics
Announcements

Assignment 4. Due Sunday, June 25th @ 11:59 pm. Late submissions will not be accepted, and late days may not be used.

FINAL EXAM: Thu, Jun 29 at 7-9:30 PM Room: BUCH A101

Final will comprise: 10 –15 short questions + 3–4 problems

- Work on all practice exercises (including 9.B) and sample review questions and problems (will be posted over the weekend)
- While you revise the learning goals, work on review questions — I may even reuse some verbatim 😊
- Come to remaining Office hours! (schedule for next week will be posted on piazza) My office hour tomorrow will be at 2PM