Markov Decision Processes

A Brief Introduction and Overview

Jack L. King, Ph.D.
Genoa (UK) Limited
Presentation Outline

• Introduction to MDP’s
  – Motivation for Study
  – Definitions
  – Key Points of Interest
  – Solution Techniques

• Partially Observable MDP’s
  – Motivation
  – Solution Techniques

• Graphical Models
  – Description
  – Application Techniques
Introduction to Markov Decision Processes
Sequential Decision Process

- **Sequential Decision Process**
  - A series of decisions are made, each resulting in a reward and a new situation. The history of the situations is used in making the decision.

- **Markov Decision Process**
  - At each time period $t$ the system state $s$ provides the decision maker with all the information necessary for choosing an action $a$. As a result of choosing an action, the decision maker receives a reward $r$ and the system evolves to a (possibly different) state $s'$ with a probability $p$. 
Applications

- Inventory
- Maintenance
- Service (Queuing)
- Pricing
- Robot Guidance
- Risk Management
Key Points of Interest

1. Is there a policy that a decision maker can use to choose actions that yields the maximum rewards available?

2. Can such a policy (if it exists) be computed in finite time (is it computationally feasible)?

3. Are there certain choices for optimality or structure for the basic model that significantly impact 1.) and 2.)?
Definitions

S  set of possible world states
A  set of possible actions
R(s,a) real-valued reward function
T  description of each action’s effect in a state. T: SxA-->Prob(S). Each state and action specifies a new (transition) probability distribution for the next state.
π  a policy mapping from S to A {d^t(s)=a}
How to evaluate a policy?

- **Expected total rewards**
  - Leads to infinite values

- **Set finite horizon**
  - Somewhat arbitrary

- **Discount rewards**
  - Most studied and implemented
  - Gives weighting to earlier rewards
  - Interpretation in economics is clear and also can be used as a general stopping criteria.
Discounted Value Function

- Rewards are discounted for each period.
  \[ V = R_0 + \rho R_1 + \rho^2 R_2 + \ldots \rho^T R_T \]
  \[ 0 < \rho < 1 \]
- Value function partially orders the policy so at least one optimal policy exists.
- We are generally interested in stationary policies where the decision rules \( d_t \) are the same for all time periods.
Solving MDPs

• Enumerated Value Function

\[ V^i(\pi, s) = r(s, d^i(s)) + \rho \sum_{s' \in S} \tau(s, d^i(s), s')V^{i+1}(\pi, s') \]

• Dynamic Programming

\[ V_n(\pi, s) = r(s, d_n(s)) + \rho \sum_{s' \in S} \tau(s, d_n(s), s')V_{n-1}(\pi, s') \]

and \[ V_0(\pi, s) = 0 \] for all \( s \)

where starting state is \( s \) with \( n \) steps remaining.
Optimal Value Function for MDP

• Optimal Value function can be used to find the maximum reward policy.

\[ V_n^*(s) = \max_{a \in A} \left[ r(s, a) + \rho \sum_{s' \in S} \tau(s, a, s') V_{n-1}^*(s') \right] \]

• Policy iteration alternative.

\[ d_n^*(s) = \arg \max_{a \in A} V_{n}^{*,a}(s) \]
MDP Summary

• Important class of sequential decision processes.
• Well known algorithms for solution (Bellman equations with dynamic programming).
• Solutions are at best polynomial in actions and states \( \ldots O(s^3) \) when converted to LP.
• Requires detailed exposition of actions and states.
• Becomes intractable and unmanageable very quickly for interesting problems.
MDP References

• Bellman, *Dynamic Programming*, 1957 – Seminal work in dynamic programming and value functions (Bellman equations).

• Cormen et.al., *Introduction to Algorithms*, 1990 – Good description of algorithms for dynamic programming and greedy search (policy iteration).

• Ross, *Applied Probability Models with Application to Optimization*, 1992 -- Introduction to optimization and MDP’s, well written with good examples.
Example MDP

Should I refresh the browser when the network is up/down?
Example Calculations

\[
Q(\text{Up, refresh}) = 30 + 0.9V \times (\text{Up}) + 0.1V \times (\text{Down})
\]
\[
Q(\text{Up, don't}) = 100 + 0.9V \times (\text{Up}) + 0.1V \times (\text{Down})
\]
\[
Q(\text{Down, refresh}) = 70 + 0.3V \times (\text{Down}) + 0.7V \times (\text{Up})
\]
\[
Q(\text{Down, don't}) = 0 + 0.8V \times (\text{Down}) + 0.2V \times (\text{Up})
\]

\[
V \times (\text{Up}) = 100 + 90 + 7(\text{don't})
\]
\[
V \times (\text{Down}) = 70 + 21 + 135(\text{refresh})
\]

Policy is to refresh when the Network is down.
Partially Observable MDP’s

• POMDP addressed the problem of
  – You don’t normally know if the network is functioning well (e.g. up or down).
  – You only observe the response as you see text and pictures appear and the status bar display.
Solving POMDP’s

• Convert to MDP
  – Add a set of observations \( O \).
  – Add a probability distribution \( U(s,o) \) for each state.
  – Add an initial state distribution \( I \).

• States now are called Belief states and are updated using Bayes Rule.

• Then solve the POMDP as you would an MDP over the belief state probability distribution.
POMPD Equations

\[ r(s,a) = \sum_{s' \in S} \tau(s,a,s')O(a,s',z)R(s,a,s',z) \]

\[ \alpha(b,a) = \sum_{s \in S} b(s)r(s,a) \]

\[ B'(b,a) = \{ b'_z \mid z \in Z \} \]

\[ \psi(b,a,b') \]

- Relate states to an observation Z using probability distribution O.
- Reward function
- Successor State
- Transition Matrix
Optimal value function for POMDP

\[ V_n^*(b) = \max_{a \in A} \left[ \omega(b, a) + \rho \sum_{b' \in B'(b, a)} \psi(b, a, b') V_{n-1}^*(b') \right] \]

Explicit form of DP equation for POMDP

\[ V_n^*(b) = \max_{a \in A} \sum_{s \in S} b(s) r(s, a) + \rho \sum_{s \in S} \sum_{s' \in S} \sum_{z \in Z} b(s) \tau(s, a, s') o(a, s', z) V_{n-1}^* \left( b_z^a \right) \]
Solving MDPs

• POMDP solution approaches must be able to deal with continuous space of belief states.
  – Exact solutions, LP-based, $O(n^x)$
    • Enumeration
    • Pruning
    • Witness
  – Approximate Solutions
    • Heuristics
    • Grid-based
    • Stochastic Simulation
POMDP Summary

• Variation of MDP when state information is not explicitly known.
• Convert to MDP using belief states and solve as an MDP over belief states.
• Problem is now continuous and solution algorithms for MDP’s no longer work.
• Exact solutions are at best exponential in actions and states …O(s^a) when converted to LP.
• Requires detailed exposition of actions and belief states, along with starting values.
• Becomes intractable and unmanageable very quickly for interesting problems.
POMDP References

- Cassandra, Tony’s POMDP’s, website at Brown, tutorial, examples, see his PhD thesis for a nice survey of POMDP algorithms.
  http://www.cs.brown.edu/research/ai/pomdp/
- Ross, Applied Probability Models with Application to Optimization, 1992 -- Introduction to optimization, MDP’s and POMDP’s, well written with good examples.
Graphical Model

• Main ideas
  – Nodes represent variables and arcs are conditional dependencies.
  – Joint probability for model is factored using conditional probabilities between variables.
  – Model is structured as directed acyclic graph (DAG).

• Advantages
  – Factorization corresponds to data availability and probabilities can be built using historical frequency.
  – Independence between variables is exploited resulting in a more compact representation.
  – Efficient message-passing algorithms can be used resulting in tractability for interesting problems.
Graphical Model

Nodes represent the variables in the model, and the links show which variables depend on others.
Graphical Model References


• Lauritzen, 1996, Graphical Models, general description of graphical models, relationship to Markov models, very thorough.

• King, 2001, Operational Risk Measurement and Modelling. Example of application of graphical models to risk management.
Summary

• Markov Decision Process
  – An effective way to address complex problems that deal with uncertainty.
  – POMDP’s address the problem of observing the states of the process.
  – Graphical Models address the problem of representation and tractability of the models.