To Infinity and Beyond

... or ...

Bayesian Nonparametric Approaches for Reinforcement Learning in Partially Observable Domains

Finale Doshi-Velez
December 2, 2011
To Infinity and Beyond

… or …

Bayesian Nonparametric Approaches for Reinforcement Learning in Partially Observable Domains

Finale Doshi-Velez
December 2, 2011
Outline

- Introduction: The partially-observable reinforcement learning setting
- Framework: Bayesian reinforcement learning
- Applying nonparametrics:
  - Infinite Partially Observable Markov Decision Processes
  - Infinite State Controllers*
  - Infinite Dynamic Bayesian Networks*
- Conclusions and Continuing Work

* joint work with David Wingate
Outline

- Introduction: The partially-observable reinforcement learning setting
- Framework: Bayesian reinforcement learning
- Applying nonparametrics:
  - Infinite Partially Observable Markov Decision Processes
  - Infinite State Controllers
  - Infinite Dynamic Bayesian Networks
- Conclusions and Continuing Work
Example: Searching for Treasure
Example: Searching for Treasure

I wish I had a GPS...
Example: Searching for Treasure

Partial Observability: can't tell where we are just by looking...
Example: Searching for Treasure
Example: Searching for Treasure

Reinforcement Learning:
don't even have a map!
The Partially Observable Reinforcement Learning Setting

\[ a_{t-1}, a_t, a_{t+1}, a_{t+2}, \ldots \]

\[ o_{t-1}, o_t, o_{t+1}, o_{t+2}, \ldots \]

\[ r_{t-1}, r_t, r_{t+1}, r_{t+2}, \ldots \]
The Partially Observable Reinforcement Learning Setting

Given a history of actions, observations, and rewards.
The Partially Observable Reinforcement Learning Setting

Given a history of actions, observations, and rewards

How can we act in order to maximize long-term future rewards?
The Partially Observable Reinforcement Learning Setting

... a_{t-1} a_t a_{t+1} a_{t+2} ...

... o_{t-1} o_t o_{t+1} o_{t+2} ...

... r_{t-1} r_t r_{t+1} r_{t+2} ...

recommender systems
The Partially Observable Reinforcement Learning Setting

clinical diagnostic tools
Motivation: the Reinforcement Learning Setting

Key Challenge: The *entire history* may be needed to make near-optimal decisions
The Partially Observable Reinforcement Learning Setting

All past events are needed to predict future events
General Approach: Introduce a statistic that induces Markovianity

The representation summarizes the history
General Approach: Introduce a statistic that induces Markovianity

Key Questions:
- What is the **form** of the statistic?
- How do you **learn** it from limited data? (prevent overfitting)
History-Based Approaches

Idea: build the statistic directly from the history

Examples:
- U-Tree\(^1\) (learn with statistical tests)
- Probabalistic Deterministic Finite Automata\(^2\) (learned via validation sets)
- Predictive State Representations\(^3\) (learned via eigenvalue decompositions)

1. e.g. McCallum, 1994
2. e.g. Mahmud, 2010
3. e.g. Littman, Sutton, and Singh, 2002
Hidden-Variable Approaches

Idea: system is Markov if certain hidden variables are known

Examples: POMDPs (and derivatives)\(^1\) learned via
- Expectation-Maximization (validation sets)
- Bayesian methods (using Bayes rule)\(^2\)

Our Focus: in the Bayesian setting, “belief” \(p(s_t)\) is a sufficient statistic

1. e.g. Sondik 1971, Kaelbling, Littman, and Cassandra 1995, McAllester and Singh 1999
2. e.g. Ross, Chaib-draa, Pineau 2007, Poupart and Vlassis, 2008
Outline

- Introduction: The partially-observable reinforcement learning setting
- Framework: Bayesian reinforcement learning
- Applying nonparametrics:
  - Infinite Partially Observable Markov Decision Processes
  - Infinite State Controllers
  - Infinite Dynamic Bayesian Networks
- Conclusions and Continuing Work
Formalizing the Problem

The agent maintains a representation of how the world works as well as the world's current state.
Model-Based Approach

The agent maintains a representation of how the world works as well as the world's current state.
Being Bayesian

If the agent has an accurate world representation, we can keep a distribution over current states..
Being (more) Bayesian

If the world representation are unknown, can keep distributions over those too.

\[
p(m|h) \propto p(h|m)p(m)
\]
Why is this problem hard?

Lots of unknowns to reason about!
Why is this problem hard?

Lots of unknowns to reason about!

... and just how many unknowns are there? (Current methods struggle reasoning about too many unknowns.)
Why is this problem hard?

Lots of unknowns to reason about!

... and how many unknowns are there? Current methods struggle trying to reason about too many unknowns.
We'll address these challenges via Bayesian Nonparametric Techniques

Bayesian models on an infinite-dimensional parameter space
We'll address these challenges via Bayesian Nonparametric Techniques

Bayesian models on an infinite-dimensional parameter space

Already talked about keeping distributions $p(\ m \mid h )$ over representations
We'll address these challenges via Bayesian Nonparametric Techniques

Bayesian models on an infinite-dimensional parameter space

- Already talked about keeping distributions $p(\mathbf{m} | \mathbf{h})$ over representations
- Now place the prior $p(\mathbf{m})$ over representations with an infinite number of parameters in $\mathbf{m}$
Outline

• Introduction: The partially-observable reinforcement learning setting

• Framework: Bayesian reinforcement learning

• Applying nonparametrics:
  – Infinite Partially Observable Markov Decision Processes*
  – Infinite State Controllers
  – Infinite Dynamic Bayesian Networks

• Conclusions and Continuing Work

* Doshi-Velez, NIPS 2009
Being (more) Bayesian

If the world representation are unknown, can keep distributions over those too.

Possible world representation
Possible world state now

action
observation, reward

World

Need to choose what type of representation
We represent the world as a partially observable Markov Decision Process (POMDP)

Transition Matrix: $T(s'|s,a)$

Input: actions (observed, discrete)

Latent state: (hidden, discrete, finite)

Reward Matrix: $R(r|s,a)$

Output: observations, rewards (observed)

Observation Matrix: $O(o|s',a)$

Input: actions (observed, discrete)

Latent state: (hidden, discrete, finite)

Output: observations, rewards (observed)
“Learning” a POMDP means learning the parameter values

Transition Matrix: $T(s'|s,a)$

Reward Matrix: $R(r|s,a)$

Observation Matrix: $O(o|s',a)$

Ex.: $T(⊙|s,a)$ is a vector (multinomial)
Being Bayesian means putting distributions over the parameters

**Transition Matrix:**

\[ T(s'|s,a) \]

**Reward Matrix:**

\[ R(r|s,a) \]

**Observation Matrix:**

\[ O(o|s',a) \]

Ex.: \( T(\circ|s,a) \) is a vector (multinomial)

The conjugate prior \( p( T(\circ|s,a) ) \) is a Dirichlet distribution
Making things nonparametric: the Infinite POMDP

Transition Matrix: $T(s'|s,a)$

Reward Matrix: $R(r|s,a)$

Observation Matrix: $O(o|s',a)$

Ex.: $T(\circ|s,a)$ is a vector (infinite multinomial)

The conjugate prior $p(T(\circ|s,a))$ is a Dirichlet process

(built from the HDP-HMM)
Generative Process
(based on the HDP-HMM)

1. Sample the base transition distribution $\beta$:
   $$\beta \sim \text{Stick}(\gamma)$$

2. Sample the transition matrix in rows $T(\cdot|s,a)$:
   $$T(\cdot|s,a) \sim \text{DP}(\beta, \alpha)$$

3. For each state-action pair, sample observation and reward distributions from a base distribution:
   $$\Omega(o|s,o) \sim \text{HO}$$
   $$R(r|s,o) \sim \text{HR}$$
Model Complexity Grows with Data: Lineworld Example
Model Complexity Grows with Data: Loopworld Example

Number of States in Loopworld POMDP

Episode Number

Number of States
Incorporating Data and Choosing Actions

All Bayesian reinforcement learning approaches alternate between two stages, belief monitoring and action selection.
Incorporating Data and Choosing Actions

All Bayesian reinforcement learning approaches alternate between two stages, belief monitoring and action selection.

- **Belief monitoring**: maintain the posterior

\[ b(s, m|\mathcal{H}) = b(s|m, \mathcal{H}) b(m|\mathcal{H}) \]

Issue: we need a distribution over infinite models! Key idea: only need to reason about parameters of states we've seen.
High-level Plan: Apply Bayes Rule

\[ P(\text{model}|\text{data}) \propto P(\text{model}|\text{world}) \cdot P(\text{model}) \]

What's likely given the data? Represent this complex distribution by a set of samples from it...

How well do possible world models match the data?

A priori, what models do we think are likely?
High-level Plan: Apply Bayes Rule

\[ P(\text{model}|\text{data}) \propto P(\text{model}|\text{world}) \cdot P(\text{model}) \]

What's likely given the data? Represent this complex distribution by a set of samples from it...

How well do possible world models match the data?

A priori, what models do we think are likely?
High-level Plan: Apply Bayes Rule

\[ P(\text{model}|\text{data}) \propto P(\text{model}|\text{world}) \cdot P(\text{model}) \]

What's likely given the data?
Represent this complex distribution by a set of samples from it...

How well do possible world models match the data?

A priori, what models do we think are likely?
Inference: Beliefs over Finite Models

Estimate the parameters

$O(o|s')$ $T(s'|s,o)$

Estimate the state sequence:

Diagram with arrows and nodes
Inference: Beliefs over Finite Models

Estimate the parameters:

Discrete case, use Dirichlet-multinomial conjugacy:

Transition Prior: $\beta$

State-visit counts:

Posterior:

Estimate the state sequence:
Inference: Beliefs over Finite Models

Estimate the parameters:
Discrete case, use Dirichlet-multinomial conjugacy:

- Transition Prior: $\beta$
- State-visit counts:
- Posterior:

Estimate the state sequence:
Forward filter (e.g. first part of Viterbi algorithm) to get marginal for the last state; backwards sample to get a state sequence.
Inference: Beliefs over Infinite Models

Estimate $T, \Omega, R$ for visited states

Pick a slice variable $u$ to cut infinite model into a finite model.

Estimate the state sequence

($\text{Beam Sampling, Van Gael 2008}$)
Inference: Beliefs over Infinite Models

Estimate $T, \Omega, R$ for visited states

Pick a slice variable $u$ to cut infinite model into a finite model.

Estimate the state sequence

($\Omega(o | s', a)$, $T(s' | s, a)$, $\Omega$ (Beam Sampling, Van Gael 2008))
All Bayesian reinforcement learning approaches alternate between two stages, belief monitoring and action selection.

- **Belief monitoring**: maintain the posterior
  \[ b(s, m|h) = b(s|m, h) b(m|h) \]

  Issue: we need a distribution over infinite models! Key idea: only need to reason about parameters of states we've seen.

- **Action selection**: use a basic stochastic forward search (we'll get back to this...)

Incorporating Data and Choosing Actions
Results on Standard Problems

Rewards for Gridworld

Graph showing the rewards over iterations of experience for different algorithms, with each line representing a specific algorithm.
Results on Standard Problems

Rewards for Gridworld

Rewards for Tiger

Rewards for Network

Rewards for Shutte
Results on Standard Problems

Per-Iteration Running-Time Compared to iPOMDP

Relative Running Time Compared to iPOMDP

FFBS  FFBS-Big  EM  EM-Big  iDMM-AO  iDMM-DA  iDMM-Bipartite  UTree
Results on Standard Problems

Per-Iteration Running-Time Compared to iPOMDP

Relative Running Time Compared to iPOMDP

FFBS  FFBS-Big  EM  EM-Big  iDMM-AO  iDMM-OA  iDMM-Bipartite  UTree
Results on Standard Problems

Per-Iteration Running-Time Compared to iPOMDP

Relative Running Time Compared to iPOMDP

- FFBS
- FFBS-Big
- EM
- EM-Big
- IDMM-AO
- IDMM-OA
- IDMM-Bipartite
- UTree
Outline

• Introduction: The partially-observable reinforcement learning setting
• Framework: Bayesian reinforcement learning
• Applying nonparametrics:
  – Infinite Partially Observable Markov Decision Processes
  – Infinite State Controllers*
  – Infinite Dynamic Bayesian Networks
• Conclusions and Continuing Work

* Doshi-Velez, Wingate, Tenenbaum, and Roy, NIPS 2010
Leveraging Expert Trajectories

Often, an expert (could be another planning algorithm) can provide near-optimal trajectories. However, combining expert trajectories with data from self-exploration is challenging:

- Experience provides **direct information about the dynamics**, which indirectly suggests a policy.
- Experts provide **direct information about the policy**, which indirectly suggests dynamics.
Suppose we're turning data from an expert's demo into a policy...
Policy Priors

Suppose we're turning data from an expert's demo into a policy... a prior over policies can help avoid overfitting...
Policy Priors

Suppose we're turning data from an expert's demo into a policy... a prior over policies can help avoid overfitting... but the demo also provides information about the model

\[ \pi(\text{history}) \]

\[ \rho(\pi) \]
Policy Priors

Suppose we're turning data from an expert's demo into a policy... a prior over policies can help avoid overfitting... but the demo also provides information about the model

\[ p(\pi) \]
Policy Priors

Suppose we're turning data from an expert's demo into a policy... a prior over policies can help avoid overfitting... but the demo also provides information about the model

But if we assume the expert acts near optimally with respect to the model, don't want to regularize!
Policy Prior Model

Policy Prior → Expert Policy → Dynamics Model → Agent Policy → Agent Data → Dynamics Prior → Policy Prior
Policy Prior: What it means

Models with simple dynamics

Model Space
Policy Prior: What it means

Models with simple control policies.
Policy Prior: What it means

Joint Prior: models with few states, also easy to control.
Policy Prior Model

- Policy Prior
- Dynamics Prior
- Expert Policy
- Dynamics Model
- Agent Policy
- Expert Data
- Agent Data
Modeling the Model-Policy Link

Apply a Factorization:
the probability of the expert policy $\pi$
$p(\pi | m, \text{policy prior})$
is proportional to $f(\pi, m) g(\pi, \text{policy prior})$

Many options for $f(\pi, m)$, assume we want something like $\delta(\pi^*, \pi)$ where $\pi^*$ is the optimal policy under $m$
Modeling the World Model

Represent the world model with an infinite POMDP
Modeling the Policy

Represent the policy as a (in)finite state controller:

- \( o_t \) (Observation)
- \( n_t \) (State)
- \( a_t \) (Action)
- \( n_{t+1} \) (Next State)

Transition Matrix:

\[ T(n'|n,o) \]

Emission Matrix:

\[ P(a|n,o) \]

Policy Prior

Expert Policy

World Model

Agent Policy

Expert Data

Agent Data
Doing Inference

- Policy Prior
- Dynamics Prior
- Expert Policy
- Dynamics Model
- Agent Policy
- Expert Data
- Agent Data
Some parts aren't too hard...
Some parts aren't too hard...
But: Model-Policy Link is Hard
Sampling Policies Given Models

Suppose we choose \( f() \) and \( g() \) so that the probability of an expert policy, 
\( p(\pi \mid m, \text{data}, \text{policy prior}) \) is proportional to

\[
f(\pi, m) g(\pi, \text{data}, \text{policy prior})
\]

\( \delta(\pi^*, \pi) \) where \( \pi^* \) is opt(\( m \))

iPOMDP prior + data

where the policy \( \pi \) is given by a set of

- transitions \( T(n'|n,o) \)
- emissions \( P(a|n,o) \)
Looking at a Single $T(n'|n,o)$

Consider the inference update for a single distribution $T(n'|n,o)$:

$$f(\pi, m) \cdot g(\pi, \text{data}, \text{policy prior})$$

Easy with Beam Sampling if we have Dirichlet-multinomial conjugacy (data just adds counts to the prior)

- $\beta$: prior/mean transition from the iPOMDP
- node-visit counts: how often $n'$ occurred after seeing $o$ in $n$
- posterior Dirichlet for $T(n'|n,o)$
Looking at a Single $T(n'|n,o)$

Consider the inference update for a single distribution $T(n'|n,o)$:

$$f(\pi, m)g(\pi, data, policy\ prior)$$

Approximate $\delta(\pi^*,\pi)$ with Dirichlet counts, using Bounded Policy Iteration (BPI) (Poupart and Boutilier, 2003)

Current policy has some value for $T(n'|n,o)$

One step of BPI changes $T'(n'|n,o) = T(n'|n,o) + a$ (keeps node alignment)

More steps of BPI change $T^*(n'|n,o) = T(n'|n,o) + a^*$ (nodes still aligned)
Combine with a Tempering Scheme

Consider the inference update for a single distribution $T(n'|n,o)$:

$$f(\pi, m)g(\pi, \text{data}, \text{policy prior})$$

Approximate $\delta(\pi^*,\pi)$ with Dirichlet counts/BPI

Distribution for $T^*(n|n,o)$ scaled by $k$

$\beta$: prior/mean transition from the iPOMDP

Node-visit counts: how often $n'$ occurred after seeing $o$ in $n$

Posterior Dirichlet for $T(n'|n,o)$
Combine with a Tempering Scheme

Consider the inference update for a single distribution $T(n'|n,o)$:

$$f(\pi, m)g(\pi, data, policy\ prior)$$

Approximate $\delta(\pi^*,\pi)$ with Dirichlet counts/BPI

Distribution for $T^*(n|n,o)$ scaled by $k$ + $\beta$: prior/mean transition from the iPOMDP + node-visit counts: how often $n'$ occurred after seeing $o$ in $n$ = posterior Dirichlet for $T(n'|n,o)$
Combine with a Tempering Scheme

Consider the inference update for a single distribution $T(n'|n,o)$:

$$f(\pi, m)g(\pi, data, policy\ prior)$$

Approximate $\delta(\pi^*, \pi)$ with Dirichlet counts/BPI

Distribution for $T^*(n'|n,o)$ scaled by $k$

$\beta$: prior/mean transition from the iPOMDP

Node-visit counts: how often $n'$ occurred after seeing $o$ in $n$

Posterior Dirichlet for $T(n'|n,o)$
Sampling Models Given Policies

Apply Metropolis-Hastings Steps:

1. Propose a new model \( m' \) from \( q(m') = g( m \mid \text{all data, prior} ) \)

2. Accept the new value with probability

\[
\min \left( 1, \frac{f(\pi, m') g(m', D, p_M)}{f(\pi, m) g(m, D, p_M)} \cdot \frac{g(m, D, p_M)}{g(m', D, p_M)} \right) = \min \left( 1, \frac{f(\pi, m')}{f(\pi, m)} \right)
\]

- Likelihood ratio: \( p(m')/p(m) \)
- Proposal ratio: \( q(m)/q(m') \)
Sampling Models Given Policies

Apply Metropolis-Hastings Steps:

1. Propose a new model $m'$ from $q(m') = g(m | \text{all data, prior})$

2. Accept the new value with probability

$$
\min(1, \frac{f(\pi, m') \cdot g(m', D, p_M) \cdot g(m, D, p_M)}{f(\pi, m) \cdot g(m, D, p_M) \cdot g(m', D, p_M)}) = \min(1, \frac{f(\pi, m')}{f(\pi, m)})
$$

We still have a problem: If $f()$ is strongly peaked, will never accept!
Sampling Models Given Policies

Apply Metropolis-Hastings Steps:

1. Propose a new model \( m' \) from \( q(m') = g( m | \text{all data, prior} ) \)
2. Accept the new value with probability

\[
\min(1, \frac{f(\pi, m') g(m', D, p_M) \cdot g(m, D, p_M)}{f(\pi, m) g(m, D, p_M) \cdot g(m', D, p_M)}) = \min(1, \frac{f(\pi, m')}{f(\pi, m)})
\]

We still have a problem: If \( f() \) is strongly peaked, will never accept!

Temper again...

\[
f(\pi, m) = \delta(\pi^*, m)
\]

\[
f(\pi, m) = \exp(a \cdot V_m(\pi))
\]
Example Result
Same trend for Standard Domains
Results on Standard Problems
Outline

• Introduction: The partially-observable reinforcement learning setting
• Framework: Bayesian reinforcement learning
• Applying nonparametrics:
  – Infinite Partially Observable Markov Decision Processes
  – Infinite State Controllers
  – Infinite Dynamic Bayesian Networks*
• Conclusions and Continuing Work

* Doshi-Velez, Wingate, Tenenbaum, and Roy, ICML 2011
Dynamic Bayesian Networks
Making it infinite...
The iDBN Generative Process
Observed Nodes Choose Parents

Treat each parent as a dish in the Indian Buffet Process: Popular parents more likely to be chosen.

(time t) -> (time t+1)
Hidden Nodes Choose Parents

Treat each parent as a dish in the Indian Buffet Process: Popular parents more likely to be chosen.
Hidden Nodes Choose Parents

Treat each parent as a dish in the Indian Buffet Process: Popular parents more likely to be chosen.
Hidden Nodes Choose Parents

Key point: we only need to instantiate parents for nodes that help predict values for the observed nodes.
Other infinite nodes are still there, we just don't need them to explain the data.
Instantiate Parameters

Sample transition CPTs ~ HDP for all hidden nodes

Sample observation CPTs ~ H0 for all observed nodes
Inference

General Approach: Blocked Gibbs sampling with the usual tricks (tempering, sequential initialization, etc.)

- Resample factor-factor connections: \( p(P_{an} | P_{ak}, X, \beta) \) using Gibbs sampling.
- Resample factor-observation connections: \( p(P_{ak} | P_{an}, X, \beta) \) using Gibbs sampling.
- Resample transitions: \( p(T | P_{ak}, X, \beta) \) using Dirichlet-multinomial.
- Resample observations: \( p(\Omega | P_{an}, X, \beta, Y) \) using Dirichlet-multinomial.
- Resample state sequence: \( p(X | P_{an}, P_{ak}, \beta, T, \Omega, Y) \) using Factored frontier – Loopy BP.
- Add / delete factors: \( p(P_{an} | P_{ak}, X, \beta) \) using Metropolis-Hastings birth/death.
### Inference

**General Approach:** Blocked Gibbs sampling with tempering, sequential initialization, etc.

| Resample factor-factor connections | $p(P_{an}|P_{ak}, X, \beta)$ | Gibbs sampling |
|------------------------------------|---------------------------------|----------------|
| Resample factor-observation connections | $p(P_{ak}|P_{an}, X, \beta)$ | Gibbs sampling |
| Resample transitions | $p(T|P_{ak}, X, \beta)$ | Dirichlet-multinomial |
| Resample observations | $p(\Omega|P_{an}, X, \beta, Y)$ | Dirichlet-multinomial |
| Resample state sequence | $p(X|P_{an}, P_{ak}, \beta, T, \Omega, Y)$ | Factored frontier – Loopy BP |
| Add / delete factors | $p(P_{an}|P_{ak}, X, \beta)$ | Metropolis-Hastings birth/death |

Common to all DBN inference
Inference

**General Approach:** Blocked Gibbs sampling with (tempering, sequential initialization, etc.)

<table>
<thead>
<tr>
<th>Step</th>
<th>Probability Model</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resample factor-factor connections</td>
<td>$p(P_{an}</td>
<td>P_{ak}, X, \beta)$</td>
</tr>
<tr>
<td>Resample factor-observation connections</td>
<td>$p(P_{ak}</td>
<td>P_{an}, X, \beta)$</td>
</tr>
<tr>
<td>Resample transitions</td>
<td>$p(T</td>
<td>P_{ak}, X, \beta)$</td>
</tr>
<tr>
<td>Resample observations</td>
<td>$p(\Omega</td>
<td>P_{an}, X, \beta, Y)$</td>
</tr>
<tr>
<td>Resample state sequence</td>
<td>$p(X</td>
<td>P_{an}, P_{ak}, \beta, T, \Omega, Y)$</td>
</tr>
<tr>
<td>Add / delete factors</td>
<td>$p(P_{an}</td>
<td>P_{ak}, X, \beta)$</td>
</tr>
</tbody>
</table>

Specific to iDBN
*only 5% computational overhead!*
Example: Weather Data

Time series of US precipitation patterns...
Weather Example: Small Dataset

A model with just five locations quickly separates the east cost and the west coast data points.
Weather Example: Full Dataset

On the full dataset, we get regional factors with a general west-to-east pattern (the jet-stream).
Weather example: Full Dataset

Training and test performance (lower is better)
Outline

• Introduction: The partially-observable reinforcement learning setting
• Framework: Bayesian reinforcement learning
• Applying nonparametrics:
  – Infinite Partially Observable Markov Decision Processes
  – Infinite State Controllers
  – Infinite Dynamic Bayesian Networks
• Conclusions and Continuing Work
When should we use this?
(And what are the limitations?)

- Predictive accuracy is the priority.
  (learned representations aren't always interpretable, and they are not optimized for maximizing rewards)

- When the data is limited or fundamentally sparse...
  otherwise a history-based approach might be better.
  (most reasonable methods perform well with lots of data, and Bayesian methods require more computation)

- When the “true” model is poorly understood...
  otherwise use calibration and system identification.
  (current priors are very general, not easy to combine with detailed system or parameter knowledge)
Continuing Work

- Action-selection: when do different strategies matter?

- Bayesian nonparametrics for history-based approaches: improving probabilistic-deterministic infinite automata

- Models that match realworld properties.
Summary

In this thesis, we introduced a novel approach to learning hidden-variable representations for partially-observable reinforcement learning using Bayesian nonparametric statistics. This approach allows for

- The representation to scale in sophistication with the complexity in the data
- Tracking uncertainty in the representation
- Expert trajectories to be incorporated
- Complex causal structures to be learned
Standard Forward-Search to Determine the Value of an Action:

For some action $a_1$
Consider what actions are possible after those observations ...
... and what observations are possible after those actions ...

For some action $a_1$
Use highest-value branches to determine the action's value

Average over possible observations

Choose action with highest value

Average over possible observations
Forward-Search in Model Space

For some action $a_1$
Forward-Search in Model Space

For some action $a_1$
Forward-Search in Model Space

For some action $a_1$

weight of models $b(m)$
For some action $a_1$

Each model updates its belief $b(s)$ and weights over models $b(m)$ also change.
For some action $a_1$, Leaves use values of beliefs for each model; sampled models are small, quick to solve.

Each model updates its belief $b(s)$ and weights over models $b(m)$ also change..
Generative Process

First, sample overall popularities, observation and reward distributions for each state.
Generative Process

First, sample overall popularities, observation and reward distributions for each state-action.
Generative Process

First, sample overall popularities, observation and reward distributions for each state-action.
Generative Process

For each action, sample transition matrix using the state popularities as a base distribution.

<table>
<thead>
<tr>
<th>Start state</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>Destination state</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>...</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>...</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>...</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>...</td>
</tr>
</tbody>
</table>

Transition probability
Thought Example: Ride or Walk?

Suppose we initially think that each scenario is equally likely.
We gather some data...

- **data:** Bus come 1/2 times

  It's now the time for the bus to arrive, but it's not here. What do you do?
Compute the posterior on scenarios.

Bayesian reasoning tells us B is \(~3\) times more likely than \(\omega\)
and decide if we're sure enough:

Bayesian reasoning tells us B is ~3 times more likely than o ... but if we really prefer the bus, we still might want more data.
Results on Some Standard Problems

<table>
<thead>
<tr>
<th>Metric</th>
<th>States</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem</td>
<td>True</td>
</tr>
<tr>
<td>Tiger</td>
<td>2</td>
</tr>
<tr>
<td>Shuttle</td>
<td>8</td>
</tr>
<tr>
<td>Network</td>
<td>7</td>
</tr>
<tr>
<td>Gridworld</td>
<td>26</td>
</tr>
</tbody>
</table>
Results on Some Standard Problems

<table>
<thead>
<tr>
<th>Metric</th>
<th>States</th>
<th>Relative Training Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem</td>
<td>True</td>
<td>iPOMDP</td>
</tr>
<tr>
<td>Tiger</td>
<td>2</td>
<td>2.1</td>
</tr>
<tr>
<td>Shuttle</td>
<td>8</td>
<td>2.1</td>
</tr>
<tr>
<td>Network</td>
<td>7</td>
<td>4.36</td>
</tr>
<tr>
<td>Gridworld</td>
<td>26</td>
<td>7.36</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>EM</th>
<th>FFBS</th>
<th>FFBS-big</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tiger</td>
<td>0.41</td>
<td>0.70</td>
<td>1.50</td>
</tr>
<tr>
<td>Shuttle</td>
<td>1.82</td>
<td>1.02</td>
<td>3.56</td>
</tr>
<tr>
<td>Network</td>
<td>1.56</td>
<td>1.09</td>
<td>4.82</td>
</tr>
<tr>
<td>Gridworld</td>
<td>3.57</td>
<td>2.48</td>
<td>59.1</td>
</tr>
</tbody>
</table>
## Results on Some Standard Problems

<table>
<thead>
<tr>
<th>Metric</th>
<th>States</th>
<th>Relative Training Time</th>
<th>Test Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem</td>
<td></td>
<td>EM</td>
<td>FFBS</td>
</tr>
<tr>
<td>Tiger</td>
<td>True</td>
<td>iPOMDP</td>
<td>0.41</td>
</tr>
<tr>
<td>Shuttle</td>
<td>2</td>
<td>2.1</td>
<td>1.82</td>
</tr>
<tr>
<td>Network</td>
<td>7</td>
<td>4.36</td>
<td>1.56</td>
</tr>
</tbody>
</table>
Results on Standard Problems
Results on Standard Problems
Results on Standard Problems
Summary of the Prior