Probabilistic Inference for Planning

Gerhard Neumann\textsuperscript{1}

\textsuperscript{1}Graz University of Technology, Institute for Theoretical Computer Science

April 29, 2010
Outline

Probabilistic Inference for Motion Planning

Markov Decision Processes
  - EM for finite time MDPs
  - Infinite Horizon MDPs as Mixture Model

EM for Dynamic Baysian Networks
  - Application to POMDPS

References
Why use inference for Planning?

- Big Problem: Integration...
  - Many different methods...
  - but also many different representations!

- Probabilistic Inference might help to bring these fields together.
Why use inference for Planning?

- Probabilistic frameworks are nice...
  - Uncertainty information can be very useful.
- We can use structured representations...
  - Dynamic Bayesian Networks, Factor Graphs...
  - ... but what is a structured representation for a planning problem?
- There are fancy algorithms and approximate methods to exploit this structure...
  - Message-Passing, Loopy Belief-propagation, Variational Inference, Extended Kalman Filters...
- Can eventually also be done by (spiking) neurons...
  - Gibbs-Sampling, SEM(?)...
How to use Inference for Planning?

Traditional Inference: Estimate unobserved variables

\[ P(X|Y) \propto P(Y|X)P(X) \]

- Observation: \( Y \)
- Estimate true state \( X \), given observation \( Y \)
How to use inference for planning?

- You know where you are
- You know where you want to be in the future
  - 'mental observation' of future events
- calculate posterior over intermediate actions
How to use inference for planning?

- Compute posterior over actions, controls signals, trajectories.
  - Unobserved variables...
- Conditioned on targets, constraints, reward-events...
  - Observed variables...
- Distributed Representations:
  - arbitrary networks for goals, constraints, observations...
  - mixed discrete/continuous representations...
- No distinction between sensor and motor, perception and action!
Overview: Probabilistic Inference & Planning

History...

- (Dayan & Hinton, 1997): EM for the immediate reward case only
  - Fixed time horizon $T$
  - Solution not optimal in the sense of maximum reward
Overview: Probabilistic Inference & Planning

Interesting Topics ...

- (Toussaint et al., 2010): EM for solving (PO)MDPs: Me
- Planning in continuous spaces: Elmar
  - (Toussaint & Goerick, 2010): Inference for robot trajectories
  - (Toussaint, 2009): Approximate Inference Control (AICO)
- (Hoffman et al., 2007): Markov Chain Monte Carlo (MCMC) methods for policy learning.
- (Kappen et al., 2009): Optimal control as graphical model inference problem: Stefan K.
  - Sub-class of MDPs can be solved without Maximization-step!
- (Vlassis & Toussaint, 2009; Vlassis et al., 2009): Model-free RL: Georg Krempel, Stefan Habenschuss
Markov Decision Processes (MDPs)

- Start with theoretical framework for standard MDPs...
- ... subsequently extend to more structured representations (Dynamic Bayesian Networks).
Markov Decision Processes (MDPs)

- Random Variables: States $s_t$, actions $a_t$ and rewards $r_t$
- The world:
  - $P(s_0)$... initial state distribution
  - $P(s_{t+1}|a_t, s_t)$... transition probabilities
  - $P(r_t|a_t, s_t)$... reward probabilities
- The agent:
  - $\pi_{as} \equiv P(a_t|s_t; \pi)$... policy
MDPs: Optimality

Traditional definition of optimal behavior:
- **Performance measure**: (Discounted) sum of expected rewards
  \[ V^\pi = \mathbb{E}\left[ \sum_{t=0}^{\infty} \gamma^t r_t ; \pi \right] \]
- **Optimality**: \( \pi^* = \arg\max_\pi V^\pi \)

Learning in graphical models:
- **Policy** \( \pi \): Parameters of the DBN
- **Train parameters** \( \pi \) to maximize the likelihood of data

Interesting Questions:
- Ok...this is pure planning, so what’s the ’data’?
- Can we relate expected reward to likelihood?
- How do we solve infinite horizon problems with graphical models?
MDPs : Simplified Case

- Finite time MDP: We are only interested in the reward of the last time step
EM for finite time MDPs

MDPs : Simplified Case

- Maximum Likelihood : Optimize parameters of the DBN ($\pi$) to maximize the likelihood of observing the data
- Data : reward $r_T = 1$
  - Probability of binary reward event : 
    $$P(r_T = 1|s_t, a_t) = \frac{E[r_t|s_t, a_t] - R_{\text{min}}}{R_{\text{max}} - R_{\text{min}}}$$
- Latent variables : $s_0:T$, $a_0:T$
  - Many more latent variables than observed variables!
  - Latent Variables $\rightarrow$ use EM
- Likelihood of 'data' :
  $$P(r_T = 1; \pi) = \sum_{a_0:T, s_0:T} P(r_T = 1, a_0:T, s_0:T; \pi) = E[r_T; \pi]$$
MDPs : EM for the simplified case

Distribution over latent variables \( q(s_0:T, a_0:T) \)

- **E-step**: compute posterior over latent variables conditioned on 'data' \( r_T = 1 \)

\[
q(s_0:T, a_0:T; \pi) = \frac{P(s_0:T, a_0:T | r_T = 1; \pi)}{P(r_T = 1; \pi)}
\]

\[
= \frac{P(r_T = 1 | s_T, a_T) P(s_0:T, a_0:T; \pi)}{P(r_T = 1; \pi)}
\]

- **M-step**: compute new policy (optimum of expected data log-likelihood)

\[
\pi^{\text{new}} = \arg\max_{\pi} Q(\pi, \pi^{\text{old}})
\]

\[
= \sum_{s_0:T, a_0:T} q(s_0:T, a_0:T; \pi^{\text{old}}) \log P(r_T = 1, s_0:T, a_0:T; \pi)
\]
MDPs : EM for the simplified case

Interesting properties of the E-step:

- Compute posterior over states and actions conditioned on 'data' \( r_T = 1 \)
- Internal simulation & mental imagery:
  - Imagine to observe event \( r_T = 1 \)
  - Internally simulate the trajectory \( a_0:T, s_0:T \) to get there

But: Given the (currently fixed) policy parameters \( \pi \)

- E-Step: From all possible trajectories with the current policy, enforce the ones with high reward probabilities
- M-Step: Adapt policy parameters to match new trajectory distribution
Simplified MDP : M-step

Using the Markov property and the stationary of the process, the \( Q \)–function simplifies to:

\[
\sum_{s,a} \left[ P(r_T|s,a)P(a|s;\pi_{\text{old}})a_T(s) \right] \log P(a|s;\pi) \\
+ \sum_{s',s,a,t} \left[ b_{T-t}(s')P(s'|s,a)P(a|s;\pi_{\text{old}})a_t(s) \right] \log P(a|s;\pi) + \text{const}
\]

... where we defined:

- \( a_t(s) = P(s_t = s;\pi_{\text{old}}) \) ... State occupancy : probability of being in state \( s \) after \( t \) timesteps
- \( b_T(s) = P(r_T|s,\tau;\pi_{\text{old}}) \) ... Future reward : probability of being in state \( s \) and observing \( r_T \) in \( \tau \) timesteps.

The E-step only needs to evaluate the terms in the brackets for the M-step.
EM for finite time MDPs

MDPs : EM for the simplified case

E-step : Forward and Backward Messages:

- $a_t(s) = P(s_t = s; \pi_{\text{old}})$ ... probability of being in state $s$ in timestep $t$
  - $a_t(s_t) = \sum_{a, s_{t-1}} a_{t-1}(s_{t-1})P(s_t|s_{t-1}, a)P(a|s_{t-1}; \pi_{\text{old}})$
  - $a_0(s) = P(s_0 = s)$
  - Represents the forward messages in the graphical model!

- $b_{\tau}(s) = P(r_\tau|s', \tau; \pi_{\text{old}})$ ... probability of being in state $s$ and observing $r_\tau$ in $\tau$ timesteps.
  - $b_{\tau}(s_t) = \sum_{a, s_{t+1}} b_{\tau+1}(s_{t+1})P(s_{t+1}|s_t, a)P(a|s_t; \pi_{\text{old}})$
  - $b_{\tau}(s) = \sum_a P(r_\tau|s, a)P(a|s; \pi_{\text{old}})$
  - Represents the backward messages!
MDPs: General Case

- We want to care about all rewards $V^\pi = \mathbb{E} \left[ \sum_{t=0}^{\infty} \gamma^t r_t ; \pi \right]$
- Introduce a mixture model...
Mixture of finite-time MDPs:

- For each MDP, the agent only collects reward in the final step, but...
- ... there is an MDP for each time-horizon $T = 1 \ldots \infty$
- Each MDP is chosen randomly according to $P(T)$
- $T$ is an additional hidden variable with prior $P(T)$
Infinite Horizon MDPs as mixture model

- joint distribution:
  \[ P(r_T = 1, a_0:T, s_0:T, T; \pi) = P(T)P(r_T = 1, a_0:T, s_0:T | T; \pi) \]

- use geometric prior \( P(T) = (1 - \gamma)\gamma^T \)

- maximization of likelihood \( r = 1 \)
  \[ P(r = 1; \pi) = \sum_{T, a_0:T, s_0:T} P(T)P(r_T = 1, a_0:T, s_0:T | T; \pi) = (1 - \gamma)\sum_T \gamma^T \mathbb{E}[r_T; \pi] = (1 - \gamma)V^\pi \]

- Maximization of likelihood is equivalent to maximization of discounted future return!!
EM for MDP mixture models : M-Step

Using the Markov property and the stationarity of the process, the $Q$–function simplifies to:

$$
\sum_{s,a} \left[ P(r|s,a)P(a|s; \pi^{\text{old}})\alpha(s) \right] \log P(a|s; \pi) \\
+ \sum_{s',s,a} \left[ \beta(s')P(s'|s,a)P(a|s; \pi^{\text{old}})\alpha(s) \right] \log P(a|s; \pi) + \text{const}
$$

$\alpha$: Averaged forward messages from all length processes:

- $\alpha(s) = \sum_{t=0}^{\infty} a_t(s)P(T = t)$
- probability of being in state $s$
- no dependence on $t$ any more: $t$ is integrated out
EM for MDP mixture models: M-Step

$$
\sum_{s, a} \left[ P(r | s; a) P(a | s; \pi^{\text{old}}) \alpha(s) \right] \log P(a | s; \pi) \\
+ \sum_{s', s, a} \left[ \beta(s') P(s' | s, a) P(a | s; \pi^{\text{old}}) \alpha(s) \right] \log P(a | s; \pi) + \text{const}
$$

$\beta$ : Averaged backward messages from all length processes:

- $\beta(s) = \frac{1}{1 - \gamma} \sum_{\tau=0}^{\infty} b_\tau(s) P(T = \tau + 1)$
  - probability of observing $r$ in the future
  - no dependence on $\tau$ any more: $\tau$ is integrated out
  - for unstructured MDPs calculating $\beta(s)$ is equivalent to Policy Evaluation:

$$V_0^\pi(s) = \sum_a P(a | s; \pi) \left[ r(s, a) + \sum_{s'} \gamma P(s' | s, a) V_{k-1}^\pi(s') \right]$$
M-Step with tabular policy

Tabular policy: $P(a|s; \pi) = \pi_{as}$

- $\pi_{as}$ is constraint to normalize over $a$ for each $s$.

The M-step is then given by:

$$\pi_{as}^{\text{new}} = \pi_{as}^{\text{old}} \left[ P(r = 1|a, s) + \sum_{s'} \beta(s') P(s'|a, s) \right]$$

- Actions with higher probability of immediate + future rewards become more probable.

- Greedy Version (optimal policy is deterministic):
  $$\pi_{as} = 1 \text{ for greedy action } \arg\max_a P(r = 1|a, s) + \sum_{s'} \beta(s') P(s'|a, s)$$

- Equivalent to greedy policy used in RL (e.g. policy iteration)

- The state-occupancy-probabilities $\alpha$ are not needed in the tabular MDP case!
EM for MDPs with tabular policies

- Equivalent to policy iteration
  - E-step / Policy Evaluation: Fix current policy, evaluate V-function
  - M-step / Policy Improvement: Use greedy policy w.r.t new V-function estimate
  - The state-occupancy-probabilities $\alpha$ can be neglected

- Ok... now we have a fancy theory for a simple thing...
  - What do we gain?
  - The same algorithm also works in more structured domains (any kind of DBN)
  - $\alpha$ is still useful if a different parametrization of the policy is used.
EM for Dynamic Bayesian Networks

Dynamic Bayesian Networks

- Distributed Representation of the state
- Several random variables at each time slice
  - $s_t^1, \ldots, s_t^k, a_t^1, \ldots a_t^k$
- Instead of using a transition matrix $P$, $P$ is now represented as a list of factors over the variables $s_t, a_t, s_{t+1}$
- For exact inference use an elimination algorithm (e.g. Junction Tree method, (Murphy, 2002)) for the forward and backward messages
Application to Partially Observable MDPs (POMDPs)

World:
- initial world state distribution: $P(s_0 = s)$
- world state transitions: $P(s' | a, s)$
- observations: $P(y | s)$
- rewards: $P(r | a, s)$

Agent:
- Use an internal memory variable $b$
  - initial distribution: $P(b_0 = b) = \nu_b$
  - memory transition: $P(b' | b, y) = \lambda_{b' | b, y}$
  - reactive policy: $P(a | b, y) = \pi_{aby}$

Challenging planning task...
Application to Partially Observable MDPs (POMDPs)

Graphical Model:
Application to Partially Observable MDPs (POMDPs)

Update rules for the M-step:

$$\pi_{aby}^{\text{new}} = \frac{\pi_{aby}^{\text{old}}}{C_{by}} \sum_s \left[ P(r|a, s) + P(y|s) \sum_{b', s'} \beta(b', s') \lambda_{b'by} P(s'|a, s) \alpha(b, s) \right]$$

$$\lambda_{b'by}^{\text{new}} = \frac{\lambda_{b'by}^{\text{old}}}{C_{b'y}} \sum_{s', a, s} \beta(b', s') P(s'|a, s) \pi_{aby} P(y|s) \alpha(b, s)$$

$$\nu_b^{\text{new}} = \frac{\nu_b^{\text{old}}}{C_b} \sum_s \beta(b, s) P(s_0 = s)$$

► State occupancy $\alpha(b, s)$ is now needed
Experiments

Training the memory gate to primitive reactive behaviors

- Turtle: Move forward, turn right, turn left, wait
- State Space: positions × orientations
- Observations: Adjacent walls relative to the turtles orientation (4 bit)
Experiments

Turtle has to find through a maze:

- It has to **remember how many junctions** it has already passed.
- At junctions turn left, right, right, left.
- Train POMDP controller with $B = 3$ memory states.
Conclusion

- Maximization expected future reward $\iff$ likelihood maximization
  - arbitrary reward signals
  - no fixed time horizon...
  - ...done by introducing a mixture of DBNs.
- Unstructured MDP : Standard Policy Iteration
- But : Generalizes to arbitrary DBNs
  - consider structured representation of the environment (e.g. factorization or hierarchies)
  - as well as in the agent (e.g. hierarchical policies or multiple agents)
- Inference techniques can be used to exploit such structure
  - Variational approaches, message passing, approximate belief representations...
The End

Thanks for your attention
Application to POMDPs

Planning by Probabilistic Inference.
_In: Proc. of the 9th Int. Workshop on Artificial Intelligence and Statistics._

Using expectation-maximization for reinforcement learning.
_Neural Comput., 9_(2), 271–278._

Hoffman, Matthew, Doucet, Arnaud, de Freitas, Nando, & Jasra, Ajay. 2007.
Bayesian Policy Learning with Trans-Dimensional MCMC.
_In: NIPS._

*Optimal control as a graphical model inference problem._


Toussaint, Marc, & Goerick, Christian. 2010. *A Bayesian View on Motor Control and Planning.*

Graphical Models for Planning and Imitation in Uncertain Environments.

Vlassis, Nikos, & Toussaint, Marc. 2009.  
*Model-free reinforcement learning as mixture learning.*

Vlassis, Nikos, Toussaint, Marc, Kontes, Georgios, & Piperidis, Savas. 2009.  
*Learning model-free robot control by a Monte Carlo EM algorithm.*
Expectation Maximization - revisited

Definitions:

- let $X$ be a set of hidden variables and
- let $Y$ be a set of observed variables
- let $P(X, Y; \theta)$ be a parametrized probabilistic model

We want to find the parameters $\theta$ which maximize the (log) likelihood $P(Y; \theta)$ of the observed variables.
Free energy view on EM

Instead of maximizing $P(Y; \theta)$, we maximized lower bound $F(q, \theta) \leq P(Y; \theta)$

$$F(q, \theta) = \log P(Y; \theta) - D_{KL}(q(X)||P(X|Y; \theta))$$

$$= \ldots$$

$$= \sum_X q(x) \log P(X, Y; \theta) + H(q)$$
Free energy view on EM

- E-step: find $q$ that maximizes $F(q, \theta)$ for fixed $\theta^{old}$
  - find $q$ which minimizes Kullback-Leibler Divergence
  - Makes lower bound tight!
  - $q(X; \theta) = P(X|Y; \theta)$

- M-step: find $\theta$ which maximizes expected complete data log-likelihood
  $Q(\theta, \theta^{old}) = \sum_X q(X; \theta^{old})P(X, Y; \theta)$