Algorithms for Graphical Models (AGM)

Dynamic graphical models

$Date: 2006/12/05 09:14:49 $

AGM-16

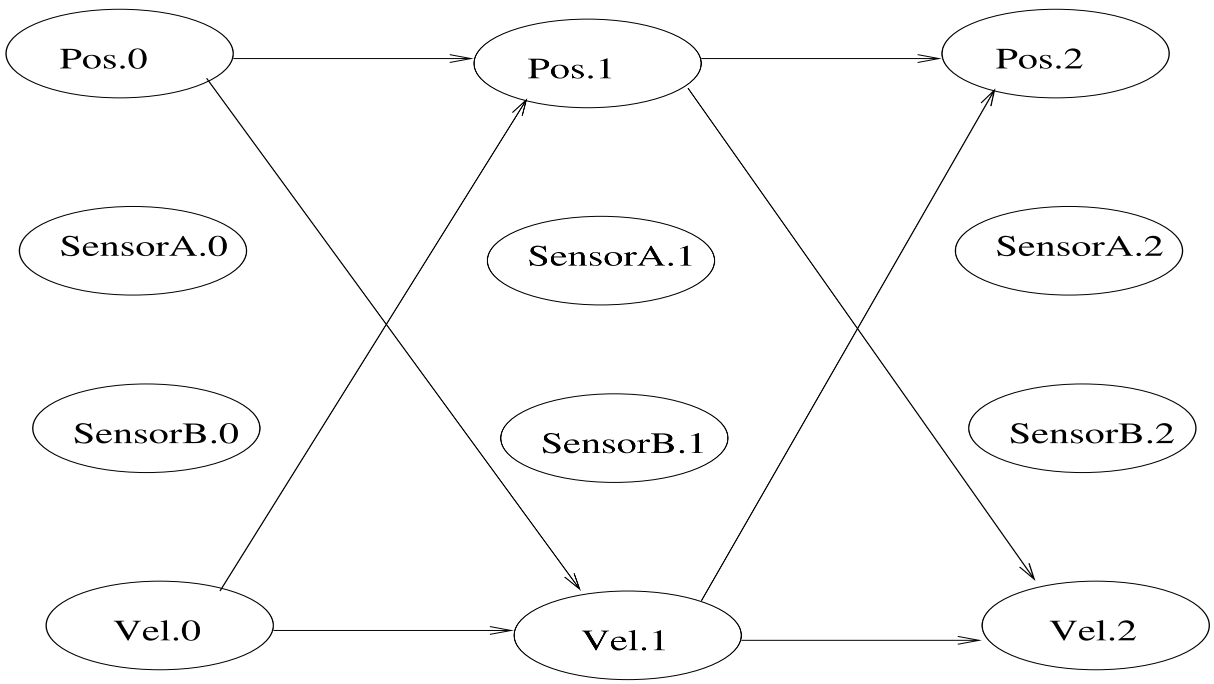
Overview

- Standard Bayesian nets model **static** situations with a **fixed** (finite) set of random variables

- Dynamic Bayesian networks model processes which evolve dynamically over time

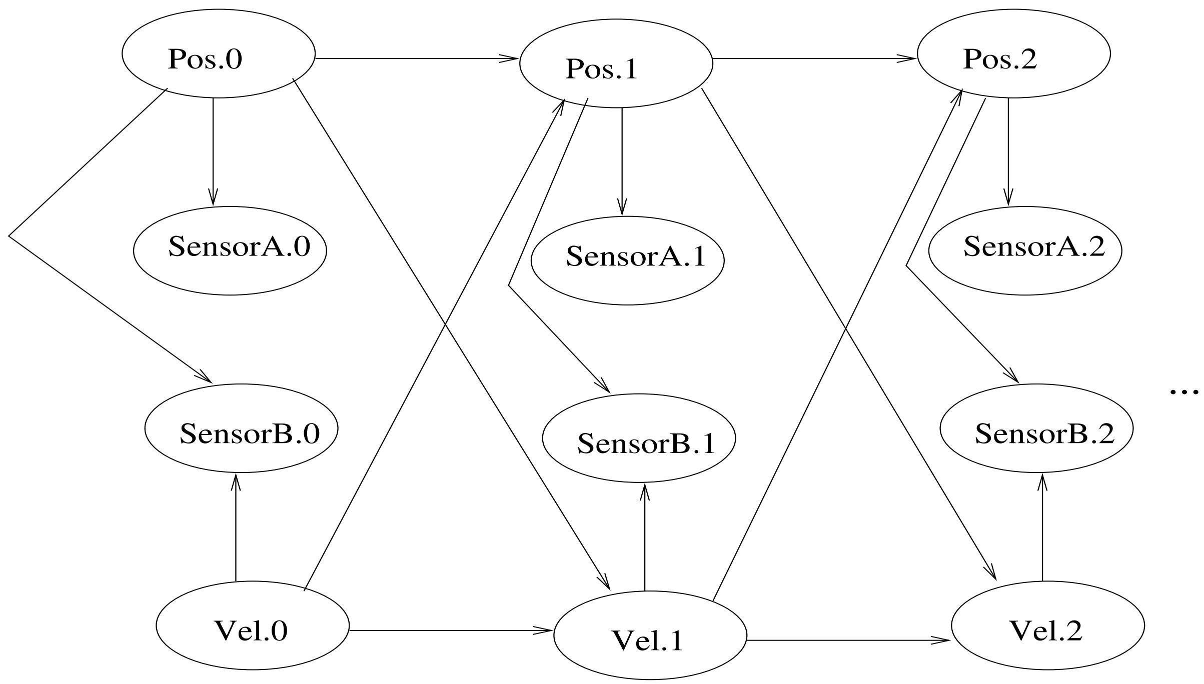
DBNs for robot navigation

Focussing on true position and velocity



DBNs for robot navigation

The full story



The problem

Let $\mathbf{X}^{(t)}$ represent the state of our world at time t

$$\mathbf{X}^{(t)} = (Pos_t, SensorA_t, SensorB_t, Vel_t, ThinkPos_t)$$

We have a distribution over trajectories:

$$P(\mathbf{X}^{(0)}, \mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(t)}) = \\ P(\mathbf{X}^{(0)})P(\mathbf{X}^{(1)}|\mathbf{X}^{(0)}) \ldots P(\mathbf{X}^{(t)}|\mathbf{X}^{(0)}, \mathbf{X}^{(1)}, \ldots \mathbf{X}^{(t-1)})$$

That's a lot of parameters

The solution (part1)

The Markov assumption: $\mathbf{X}^{(t+1)}$ is independent of $\mathbf{X}^{(t')}$ for $t' < t$ given $\mathbf{X}^{(t)}$

Our state variables are expressive enough to summarise all relevant information about the past

$$P(\mathbf{X}^{(0)}, \mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(t)}) =$$
$$P(\mathbf{X}^{(0)})P(\mathbf{X}^{(1)}|\mathbf{X}^{(0)})P(\mathbf{X}^{(2)}|\mathbf{X}^{(1)}) \ldots P(\mathbf{X}^{(t)}|\mathbf{X}^{(t-1)})$$

The solution (part2)

If all the $P(\mathbf{X}^{(t)}|\mathbf{X}^{(t-1)})$ were different that's an infinite number of probabilities to define!

So assume that $P(\mathbf{X}^{(t)}|\mathbf{X}^{(t-1)})$ is the same for every t.

The process is **time-invariant** or **stationary**

A dynamic Bayesian network

So just need: $P(\mathbf{X}^{(0)})$ (standard Bayesian network)

and $P(\mathbf{X}^{(t)}|\mathbf{X}^{(t-1)})$ a **network fragment** where the variables in $\mathbf{X}^{(t-1)}$ have no parents

What can we do with our DBN?

- Given a sequence of sensor readings, we can get a distribution over the true position and velocity

- This is our **belief state**

- Could also add variables for eg sensor failure

- Due to our assumptions it is quite easy to update our belief state when a new set of sensor readings come in.

Inference in DBNs

- Unfortunately, generally not possible to get a compact representation for the belief state, so have to resort to approximate inference:

  1. Keep track of just the high probability assignments in the belief state

  2. Stochastic simulation

Stochastic simulation

- Generate a ‘beam’ of M trajectories through the DBN

- Have to weight each trajectory by how likely the observed evidence is given the states in the trajectory.

- This is likelihood weighting (not MCMC). Problem is that the weight of each trajectory will get very small as time goes on

- Solution is the **Survival of Fittest algorithm**—kill off the low weight trajectories.

DBN software and slides

DBNs are not in `gPy` but are implemented in Kevin Murphy's *Bayes Net Toolbox for Matlab*.

Kevin's MATLAB software is at: `http://bnt.sourceforge.net`

There's a nice slide show on inference in DBNs at: `http://www-robotics.stanford.edu/%7Exb/uai98/uai98slides/index.html`

Digression: Relating attributes and relating things

- In the Asia model
  - we **do** (probabilistically) relate attributes of a patient
  - we **do not** represent relationships between patients

- In a DBN
  - we **do** (probabilistically) relate attributes of a state-of-the-world
  - also we **do** represent relationships between successive states-of-the-world

Making decisions in a dynamic world

Dynamic model + decisions = Partially Observable Markov Decision Process (POMDP)

Decision Process Not a passive observer, we (or the robot) decide between courses of action

Markov (Informally) What happens next depends on how things are now and what we do now

Partially Observable Don't know everything about the current state of the world.

POMDP overview

Transition model $P(S'|S,A)$. Where you end up depends on where you are and what you do.

Utilities

- States to avoid
- Goal states
- Time taken to get to goal states

Policies What to do

Example application: Sewer Robots

Description quoted from a GMD (as was) project on *sewer robots*:

Robot navigation in sewers is prone to suffering from uncertainty

1. of sensing (overlooking or mis-interpreting landmarks),
2. of information (out-dated or inaccurate maps),
3. of motion control (wrong turning at junctions)

Precision in any of these three uncertainty sources can help resolve uncertainty in the others.

AGM-16

Example application: Sewer Robots

The position uncertainty resulting from the possibility of wrong turning and wrong landmark classification is represented in a Partially Observable Markov Decision Process (POMDP), yielding a probability distribution over robot positions in the network that gets updated after moves from a landmark to a neighboring one and after classifying the recent landmark.

Using this POMDP model, the robot motion through the sewer according to a given path plan is monitored, deviations from the plan are detected with high probability, and corrections made accordingly.