CS 188: Artificial Intelligence

Dynamic Bayes Nets and Particle Filters

Instructor: Stuart Russell and Peyrin Kao

University of California, Berkeley
Dynamic Bayes Nets (DBNs)

- We want to track multiple variables over time, using multiple sources of evidence
- Idea: Repeat a fixed Bayes net structure at each time
- Variables at time $t$ can have parents at time $t-1$
DBNs and HMMs

- Every HMM is a single-variable DBN
- Every discrete DBN is an HMM
  - HMM state is Cartesian product of DBN state variables

\[ X_{t} \xrightarrow{} X_{t+1} \]
\[ Y_{t} \xrightarrow{} Y_{t+1} \]
\[ Z_{t} \xrightarrow{} Z_{t+1} \]

- Sparse dependencies => exponentially fewer parameters in DBN
  - E.g., 20 Boolean state variables, 3 parents each;
    DBN has \( 20 \times 2^3 = 160 \) parameters, HMM has \( 2^{20} \times 2^{20} = \sim 10^{12} \) parameters
Exact Inference in DBNs

- Variable elimination applies to dynamic Bayes nets
- Offline: “unroll” the network for $T$ time steps, then eliminate variables to find $P(X_T|e_{1:T})$
  
  
- Online: eliminate all variables from the previous time step; store factors for current time only
- Problem: largest factor contains all variables for current time (plus a few more)
Application: ICU monitoring

- **State**: variables describing physiological state of patient
- **Evidence**: values obtained from monitoring devices
- **Transition model**: physiological dynamics, sensor dynamics
- **Query variables**: pathophysiological conditions (a.k.a. bad things)
Toy DBN: heart rate monitoring

- Parameter variable
- State variable
- Sensor variable
- Sensor state variable
The enhanced heart-rate DBN’s inferences on data from a healthy 40-year-old.
ICU data: 22 variables, 1min ave
Blood pressure measurement
One-second vs one-minute data

“Bag” artifacts at 1-second resolution

Zeroing artifact at 1-second resolution

“Bag” artifacts as observed in one-minute average data

Zeroing artifact as observed in 1-minute average data
Sample blood-draw dataset no. 11

- Bag pressure estimate
- Valve open to air
- Valve open to bag
- BP estimate
- Observed BP
Detection of “bag” events

ROC curve for hypertension detection (SBP>160mmHg)
Particle Filtering
We need a new algorithm!

- When $|X|$ is more than $10^6$ or so (e.g., 3 ghosts in a 10x20 world), exact inference becomes infeasible
- Likelihood weighting fails completely – number of samples needed grows \textit{exponentially} with $T$
We need a new idea!

- The problem: sample state trajectories go off into low-probability regions, ignoring the evidence; too few “reasonable” samples
- Solution: kill the bad ones, make more of the good ones
- This way the population of samples stays in the high-probability region
- This is called *resampling* or survival of the fittest
- Represent belief state by a set of samples
  - Samples are called *particles*
  - Time per step is linear in the number of samples
  - But: number needed may be large

- This is how robot localization works in practice
Our representation of $P(X)$ is now a list of $N << |X|$ particles.

$P(x)$ approximated by number of particles with value $x$

- So, many $x$ may have $P(x) = 0$!
- More particles => more accuracy (cf. frequency histograms)
- Usually we want a low-dimensional marginal

  - E.g., “Where is ghost 1?” rather than “Are ghosts 1,2,3 in [2,6], [5,6], and [8,11]?”

<table>
<thead>
<tr>
<th>Particles:</th>
</tr>
</thead>
<tbody>
<tr>
<td>(3,3)</td>
</tr>
<tr>
<td>(2,3)</td>
</tr>
<tr>
<td>(3,3)</td>
</tr>
<tr>
<td>(3,2)</td>
</tr>
<tr>
<td>(3,3)</td>
</tr>
<tr>
<td>(3,2)</td>
</tr>
<tr>
<td>(1,2)</td>
</tr>
<tr>
<td>(3,3)</td>
</tr>
<tr>
<td>(3,3)</td>
</tr>
<tr>
<td>(2,3)</td>
</tr>
</tbody>
</table>
- Particle $j$ in state $x_t^{(j)}$ samples a new state directly from the transition model:

\[ x_{t+1}^{(j)} \sim P(X_{t+1} \mid x_t^{(j)}) \]

- Here, most samples move clockwise, but some move in another direction or stay in place.
Particle Filtering: Update step

- After observing $e_{t+1}$:
  - As in likelihood weighting, weight each sample based on the evidence
    \[ w^{(j)} = P(e_{t+1} | x_{t+1}^{(j)}) \]
  - Particles that fit the data better get higher weights, others get lower weights
  - Normalize the weights across all particles

Particles:

- $(3,2)$  $w=.9$
- $(2,3)$  $w=.2$
- $(3,2)$  $w=.9$
- $(3,1)$  $w=.4$
- $(3,3)$  $w=.4$
- $(3,2)$  $w=.9$
- $(1,3)$  $w=.1$
- $(2,3)$  $w=.2$
- $(2,2)$  $w=.4$
Particle Filtering: Resample

- Rather than tracking weighted samples, we **resample**

- \( N \) times, we choose from our weighted sample distribution (i.e., draw with replacement)

- Now the update is complete for this time step, continue with the next one (with weights reset to \( 1/N \))
Summary: Particle Filtering

- **Particles**: track samples of states rather than an explicit distribution

Consistency: see proof in AIMA Ch. 14
Particle filtering on umbrella model

![Graph showing the average absolute error over time step for different models.]

- LW(25)
- LW(100)
- LW(1000)
- LW(10000)
- ER/SOF(25)
Robot Mapping

- **SLAM: Simultaneous Localization And Mapping**
  - Robot does not know map or location
  - State $x_t^{(l)}$ consists of position+orientation, map!
  - (Each map usually inferred exactly given sampled position+orientation sequence: RBPF)

DP-SLAM, Ron Parr
Particle Filter SLAM – Video 2

[Demo: PARTICLES-SLAM-fastslam.avi]