Learning Dynamic Bayesian Networks

Ashley Mills

ashley@igi.tugraz.at

Structure of talk

1. Representation of DBNs
2. Inference in DBNs
3. Parameter learning in DBNs
4. An application which uses DBNs
5. Retrospection

Representation in DBNs

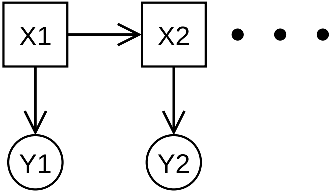
Introduction

- DBNs are extensions of BNs over potentially-infinite collections of RVs $Z_1, Z_2, \ldots..$
- Usually RVs are partitioned $Z_t = (U_t, X_t, Y_t)$ into inputs U_t, states X_t, and outputs Y_t.
- DBN is a pair $(B_1, B_{\rightarrow})$, where
  - B_1 is a prior which defines $P(Z_1)$
  - $B_{\rightarrow}$ is a 2TBN which defines $P(Z_t|Z_{t-1})$ via a DAG s.t.

$$P(Z_t|Z_{t-1}) = \prod_{i=1}^{N} P(Z_t^i|Pa(Z_t^i))$$

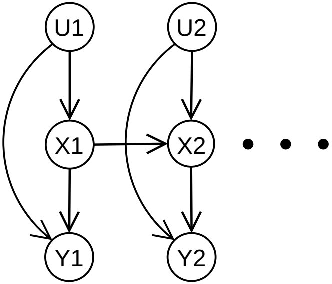
- $P(Z_{1:T}) = \prod_{t=1}^{T} \prod_{i=1}^{N} P(Z_t^i|Pa(Z_t^i))$

DBN example: Hidden Markov Model (HMM)



- $X_{t+1} \perp\!\!\!\perp X_{t-1} \,|X_t$ (Markov property)
- $Y_t \perp\!\!\!\perp Y_{t'}|X_t, \forall t' \neq t$
- $P(X,Y) = P(X_1)P(Y_1|X_1)\prod_{t=2}^{T} P(X_t|X_{t-1})P(Y_t|X_t)$
- $P(Y_t = y|X_t = i) = N(y;\mu_i,\Sigma_i)$

DBN example: Linear Gaussian Input-Output HMM



- $P(U, X, Y) =$
  $$P(X_1)P(Y_1|X_1, U_1)\prod_{t=2}^{T} P(X_t|X_{t-1}, U_t)P(Y_t|X_t, U_t)$$
- $P(X_1 = x) = N(x; x_0, V_0)$
  $P(X_{t+1} = x_{t+1}|X_t = x, U_t = u) = N(x_{t+1}; Ax + Bu, \Sigma_\alpha)$
  $P(Y_t = y|X_t = x, U_t = u) = N(y; Cx + Du, \Sigma_\beta)$
- Kalman filter: online computation of $P(X_t|y_{1:t}, u_{1:t})$.

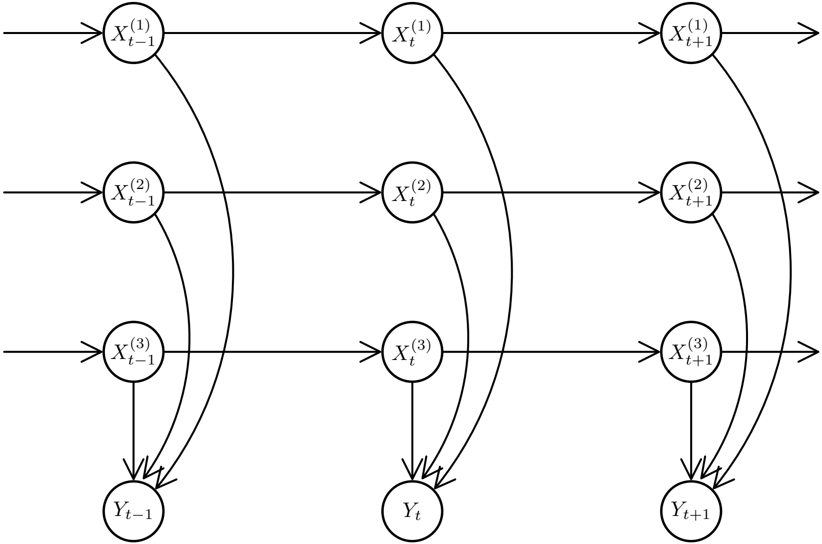
DBN example: Factorial HMM

- Imagine trying to model M objects each of which can occupy K positions.
- Doing this with standard HMM would require K^M states.
- In FHMM, state representation is distributed over M variables

$$X_t = X_t^{(1)}, ... X_t^{(m)}, ..., X_t^{(M)}$$

- Each of which can take on K values.
- State space is still K^M but we constrain transitions.

DBN example: Factorial HMM



DBN example: Factorial HMM

- Each state variable is independent

$$P(X_t|X_{t-1}) = \prod_{m=1}^{M} P(X_t^{(m)}|X_{t-1}^{(m)})$$

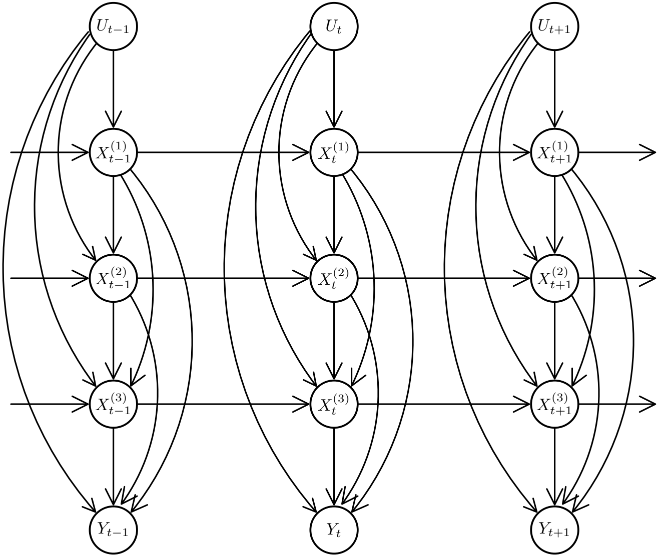
- Lets use linear-Gaussian D-dimensional observation vectors:

$$P(Y_t|X_t) = |R|^{-1/2}(2\pi)^{-D/2} \exp\left\{-\frac{1}{2}(Y_t - \mu_t)' R^{-1}(Y_t - \mu_t)\right\}$$

where

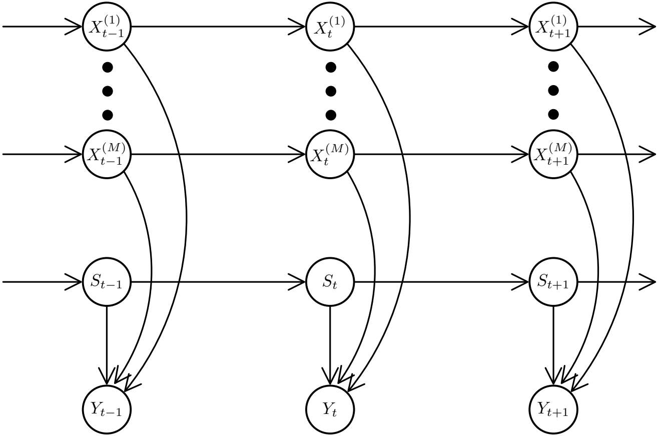
$$\mu_t = \sum_{m=1}^{M} W^{(m)} X_t^{(m)}$$

DBN example: Tree structured HMM



Stochastic decision tree with Markovian decision dynamics.

Switching State space model



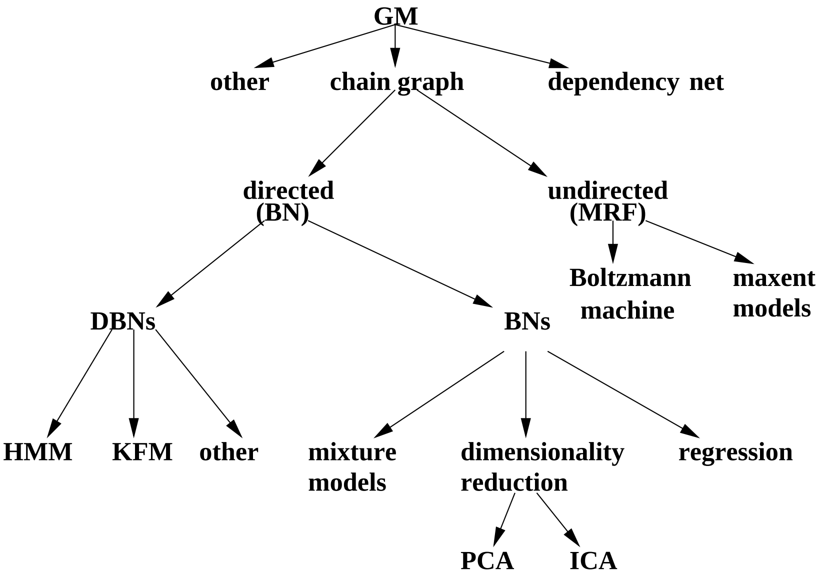
Switching State space model

$$
\begin{aligned}
P(\{S_t, X_t^{(1)}, ..., X_t^{(M)}, Y_t\}) = P(S_1) \prod_{t=2}^{T} P(S_t|S_{t-1}) \\
\times \prod_{m=1}^{M} P(X_1^{(m)}) \prod_{t=2}^{T} P(X_t^{(m)}|X_{t-1}^{(m)}) \\
\times \prod_{t=1}^{T} P(Y_t|X_t^{(1)}, ..., X_t^{(M)}, S_t)
\end{aligned}
$$

$$
\begin{aligned}
P(Y_t|X_t^{(1)}, ..., X_t^{(M)}, S_t = m) = \\
|R|^{-1/2}(2\pi)^{-D/2} \exp\left\{-\frac{1}{2}(Y_t - C^{(m)}X_t^{(m)})' R^{-1} (Y_t - C^{(m)}X_t^{(m)})\right\}
\end{aligned}
$$

Its a bit like mixture of linear Gaussian experts.

DBNs in context of GMs



Inference in DBNs

Inference in BNs

- Marginalize out variables not interested in, for example for FHMM

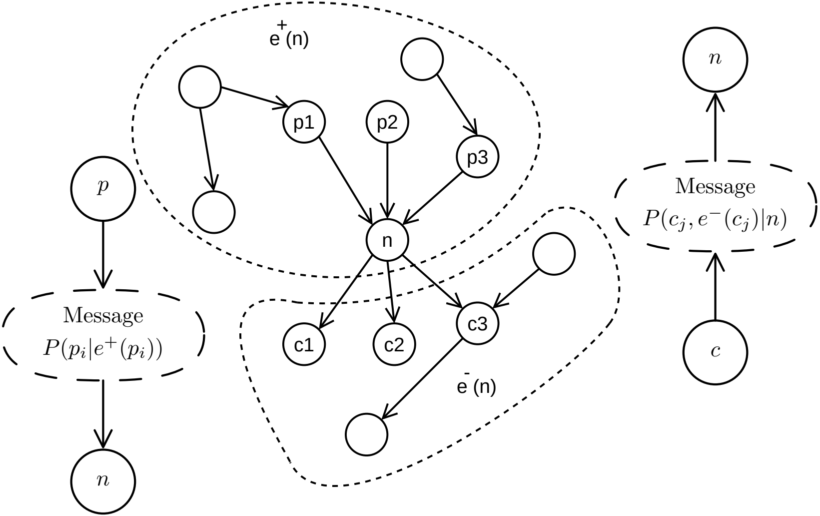
$$P(\{Y_t\}|\theta) = \sum_{\{X_t\}} P(\{X_t, Y_t\})|\theta)$$

  we have to marginalize out all possible state sequences, unless we exploit conditional independencies.
- Brute force marginalization requires at worst full joints.
- In general computating full joints is huge, and marginalization is huge.
- Efficient inference algorithms exploit conditional independencies to reduce complexity.

Exact inference

- Forward backward algorithm for HMMs.
- Belief propagation:
  - Pearl's message passing algorithm for polytrees (DAGs without undirected cycles).
  - Junction tree algorithm for general undirected networks (belief propagation on cliques).

Belief propagation



$$p(n|e) \propto \left[\sum_{\{p_1,..,p_k\}} P(n|p_1, ..., p_k) \prod_{i=1}^{k} P(p_i|e^+(p_i)) \right] \prod_{j=1}^{l} P(c_j, e^-(c_j)|n)$$

Approximate inference

- Sampling methods:
  - Importance sampling: draw random samples x from $P(X)$ and weight by likelihood $P(y|x)$, where y is evidence.
  - Markov Chain Monte Carlo
- Variational methods: for example approximate large sums of random variables by their means.
- Loopy belief propagation: apply Pearl's agorithm to the original graph even if it has undirected cycles.

Parameter Learning in DBNs

Parameter learning

Structure	Observability: Full	Observability: Partial
Known	Closed form	EM
Unknown	Local search	Structural EM

- Can either find a “best” set of parameters or infer a distribution.

Known structure, full observability

- Compute ML parameters using given sufficient statistics.
- For example, in a HMM, using the frequentist approach

$$P_{ML}(Y = \alpha | X = \beta) = \frac{Number\ of\ times\ Y_t = \alpha\ when\ X_t = \beta}{Number\ of\ times\ X_t = \beta}$$

- Dirichlet priors can be used to avoid assigning null probabilities to events absent from the training set.
- For Gaussian nodes, ML μ and Σ are just the sample μ and Σ.

Known structure, partial observability

- Sufficient statistics unavailable.
- Compute expected sufficient statistics (ESS) and treat as complete data case.
- EM: compute ESS given current θ, maximise likelihood of expected complete data with respect to θ, iterate.
- EM is gradient ascent, but general gradient ascent can be used.
- There is some debate over which is better.

Expectation Maximisation for HMMs (aka Baum Welch algorithm)

Given:

- N hidden states, M output symbols.
- Observation sequence $Y = \{Y_1 Y_2 ... Y_T\}$
- Prior selection of parameters $\lambda = (A, B, \pi)$ for state transitions $A = \{a_{ij}\}$, emission probabilities $B = \{b_j(k)\}$, and initial state distribution π_i.

Hidden:

- State sequence $X = \{X_1 X_2 ... X_T\}$
- Use indicator variables $\gamma_t(i)$ to model $P(X_t = S_i | Y, \lambda)$

Expectation Maximisation for HMMs

1. **E-STEP:**
   Use current λ to estimate state sequence via ESS.
   - Compute expected values for the state at time t:
     $\gamma_t(i) = P(X_t = S_i | Y, \lambda)$
   - Compute expected values for occurance of state tuples:
     $\varepsilon_t(i,j) = P(X_t = S_i, X_{t+1} = S_j | Y, \lambda)$.
   - These expectations are computed using the forward-backward functions.
2. **M-STEP:**
   Find new ML parameters $\bar{\lambda}$
   ML parameters $\bar{\pi}_i$, $\bar{a}_{ij}$, and $\bar{b}_j(k)$ are the expected values given the expected state sequence computed in the E-Step.
   - $\bar{\pi}_i = \gamma_1(i)$
   - $\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \varepsilon_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$
   - $\bar{b}_j(k) = \frac{\sum_{t=1\ [s.t.\ Y_t = v_k]}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}$

Unknown structure, full observability

- Local search over structure; need to define search space, scoring, and algorithm.
- ML estimate is complete graph so MAP estimate of score is used

$$Pr(G|D) = \frac{Pr(D|G)Pr(G)}{Pr(D)}$$

$$L = \log Pr(G|D) = \log Pr(D|G) + \log Pr(G) + c$$

- Give higher priors to simpler models.
- Marginal likelihood automatically penalizes complex models.

$$P(D|G) = \int_{\theta} P(D|G,\theta)P(\theta|G)$$

- Parameter independence allows likelihood decomposition:

$$P(D|G) = \prod_{i=1}^{n} \int P(X_i|Pa(X_i),\theta_i)P(\theta_i)d\theta_i$$

Unknown structure, partial observability

- Marginal likelihood is intractible and doesn't decompose

$$P(X|G) = \sum_Z \int_\theta P(X, Z|G, \theta)P(\theta|G)$$

- Can approximate marginal likelihood and use local search.
- Scoring functions exist (e.g BIC) which do decompose.
- Structural EM (local search within M step).

An application which uses DBNs:

A Comparison of HMMs and Dynamic Bayesian Networks for Recognizing Office Activities

– Nuria Oliver and Eric Horvitz

Layered HMMs and DBNs

- Model consists of layers.
- Each layer is connected to the next via its inferential results.
- Layers correspond to different levels of temporal detail and abstractness.
- Each layer of the heirarchy is trained independently.
- Paper discusses replacing top-level HMM with DBN.

Layered HMM Model: Raw signals

- Audio:
  - Two microphones capture audio and LPC coefficients are computed.
  - Coefficients are selected via PCA so 95% of variability is kept.
  - Energy, mean and variance of fundamental frequency, and zero crossing rate also extracted.
  - Sound source is localized using the Time Delay of Arrival method.
- Video:
  - Firewire camera 30FPS.
  - Extract: density of skin pixels, density of motion pixels, density of foreground pixels, and density of face pixels.
- Keyboard and Mouse:
  - History of last 1, 5, and 60 seconds of activity.

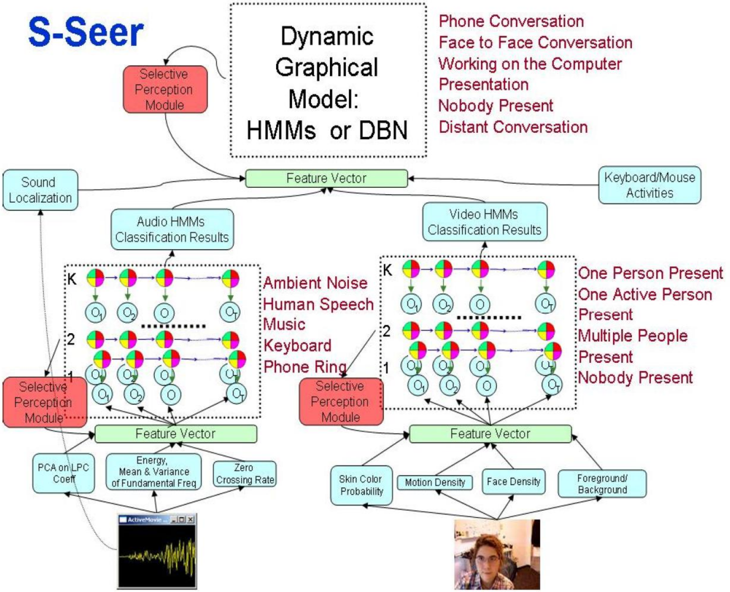
Layered HMM: First level

- Bank of discriminative audio and video signal classifier HMMs.
- One HMM trained for each class, ML model defines class of instance at runtime.
- Audio classes: human speech, music, silence, ambient noise, phone ringing, and keyboard typing.
- Video classes: nobody present, one person, one active person, and multiple people.

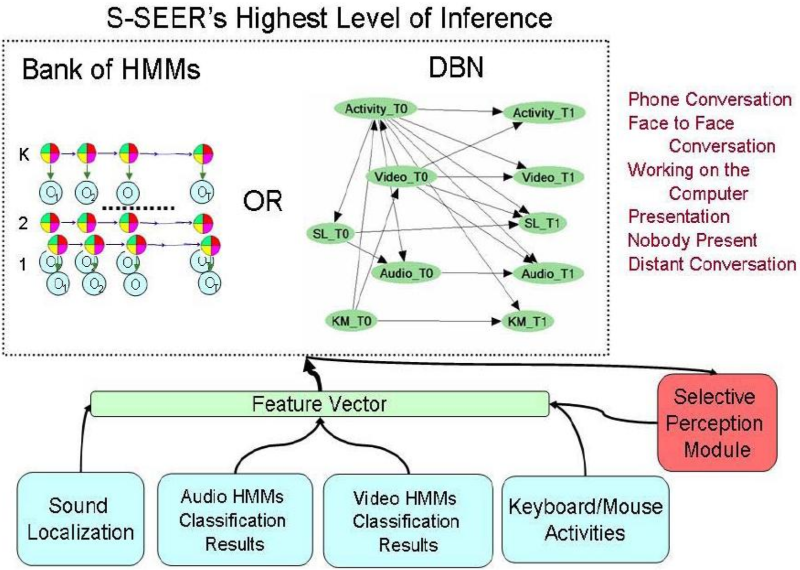
Layered HMM: Second level

- Objective is to model activities at increased temporal granularity:
  - Phone conversation, presentation, face-to-face conversation, user present but performing other activity, distant conversation, and nobody present.
- Using:
  - Audio and video inferences from level one.
  - Sound localization: left of monitor, center of monitor, right of monitor.
  - Keyboard/mouse activities: no activity, current mouse activity, current keyboard activity, both active in past second.

Layered HMM: Overview



Layered HMM: Top level



Layered HMM: comparison of second level modules

- Second level had either:
  1. Bank of discriminative HMMs.
  2. DBN with hidden "Activity" node.
- DBN and HMM top levels trained with 1800 samples (300 per activity).
- Average accuracy was 94.3% for HMM vs 97.7% for DBN without selective perception.
- Average accuracy was 92.2% for HMM vs 96.7% for DBN with selective perception.
- Performance of DBNs degrade less with selective perception because they are able to perform inference from past time slices.

Paper summary

- DBN can learn dependencies between variables that are assumed independent in HMMs.
- DBN provides a unified probability model.
- HMMs are simpler to train and are more efficient than arbitrary DBNs.
- They suggest to consider merits of each approach.

Retrospection

Retrospection

1. Representation of DBNs
2. Inference in DBNs
3. Parameter learning in DBNs
4. An application which uses DBNs

References

Slide by slide references

This presentation borrows from several sources, the table below indicates from exactly where that borrowing occurs on a slide-by-slide basis. Unlisted slides draw unspecifically.

Slides	Reference
16,18,20,25,26	[8]
4,13	[10]
6	[9]
7–12,17	[1]
23-24	[13]
28-35	[11]

[1] Zoubin Ghahramani.
Learning dynamic Bayesian networks.
Lecture Notes in Computer Science, 1387, 1998.

[2] Zoubin Ghahramani.
Graphical models: parameter learning, 2002.
http://www.gatsby.ucl.ac.uk/ zoubin/course05/.

[3] David Heckerman.
A tutorial on learning with bayesian networks.
Technical report, Microsoft Research, 1995.
http://research.microsoft.com/ heckerman/.

[4] E. Horvitz, J. Apacible, R. Sarin, and L. Liao.
Prediction, expectation, and surprise: Methods, designs, and study of a deployed traffic forecasting service.
In *Proceedings of the Conference on Uncertainty and Artificial Intelligence 2005*. AUAI Press, 2005.

[5] Finn V. Jensen.
An Introduction to Bayesian Networks.
Springer-Verlag, 1997.
First published by UCL Press, 1996.

[6] Michael I. Jordan, editor.
Learning in Graphical Models.
MIT Press, 1999.

[7] Kevin B. Korb.
Bayesian Artificial Intelligence.
Chapman and Hall/CRC Press UK, 2004.

[8] Kevin P. Murphy.
An introduction to graphical models, 2001.
http://www.cs.ubc.ca/ murphyk.

[9] Kevin P. Murphy.
Dynamic bayesian networks, 2002.
http://www.cs.ubc.ca/ murphyk.

[10] Kevin Patrick Murphy.
Dynamic Bayesian Networks: Representation, Inference and Learning.
PhD thesis, University of California Berkeley, 2002.

[11] N. Oliver and E. Horvitz.
A comparison of hmms and dynamic bayesian networks for recognizing office activities.
In *Proceedings of the Tenth Conference on User Modeling*, 2005.

[12] Judea Pearl.
Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.
Morgan Kaufmann Publishers, San Mateo, California, 1988.

[13] Lawrence R. Rabiner.
A tutorial on hidden markov models and selected applications in speech recognition.
Proceedings of the IEEE, 77(2), 1989.