

My Plan

Remaining Lectures

12 April: Learning Time Series (DBNs)

12 - 15 April: California

Phone: 650-325-7143

16 April: Review: Graphical Models and Inference

19 April: Review: Learning in Graphical Models

Student Presentations

20 Minutes

21 April: Maria, Noam

23 April: Mark, Liang

Last Time I

Learning Local Structure

Default Tables

Decision Trees

Take Home Lesson:

When data is complete

Estimation problem decomposes.

Optimize individual probability distributions for MDL/BDe

Not limited to any specific local structure

MDL framework straightforward to extend for GAM, other models.

BDe less straightforward: Need a "nice" prior (conjugate)

When data is incomplete

Estimation problem doesn't decompose

Use EM to complete the sample and optimize distributions independently.

Last Time II

Gaussian Networks

Nifty Shachter/Kenley technique for computing precision matrices:

$$W(i) = \frac{1}{v_i} \quad W(i+1) = \begin{pmatrix} W(i) + \frac{\bar{b}_{i+1}\bar{b}_{i+1}^T}{v_{i+1}} & -\frac{\bar{b}_{i+1}}{v_{i+1}} \\ -\frac{\bar{b}_{i+1}^T}{v_{i+1}} & \frac{1}{v_{i+1}} \end{pmatrix}$$

BDe for Gaussian Belief Networks

Conjugate distribution

Prior

Precision W: Wishart

Mean m: F(W)

X: N(m,W)

Posterior

X: t-distribution. Approximate as gaussian

Last Time II

Prior

Prior network.

Marginal for X is T-distributed, but use gaussian for prior network and assess an equivalent number of counts for both the variance and the mean.

(These are not separable)

Model Selection (P{D})

Closed Form Solution.

$$P\{X[m+1] | X[1] \dots, X[m]\} = T[X; \alpha_w - n + 1, T^{-1}]$$

$$= \frac{\Gamma\left(\frac{\alpha_w + 1}{2}\right) T^{-\frac{1}{2}}}{\Gamma\left(\frac{\alpha_w - n + 1}{2}\right) (\alpha_w - n + 1)^{\frac{n}{2}}} \left(1 + \frac{1}{(\alpha_w - n + 1)} (X[m] - \mu)^T T^{-1} (X[m] - \mu)\right)^{\frac{\alpha_w + 1}{2}} T^{-1} = \frac{\alpha_w (\alpha_w - n + 1)}{\alpha_w + 1} T^{-1}$$

Bayesian Estimate

Last Time II

BDe Prior

Marginal for X is T. Approximate as Gaussian.

Equivalent samples

Separate counts for the joint variance and the unconditional mean. (These are not separable)

Model Selection ($P\{D\}$)
$$P\{D|G\} = (2\pi)^{-nM/2} \left(\frac{\alpha_\mu}{\alpha_\mu'} \right)^{n/2} \frac{c(n, \alpha_w)}{c(n, \alpha_w')} |T|^{n/2} |T'|^{n/2}$$

Bayesian Estimate

“Gaussian”: $\mu = \nu$
$$W^{-1} = \frac{\alpha_\mu + 1}{\alpha_\mu (\alpha_w - n - 1)} T$$

Posterior

$$\begin{aligned} \nu' &= \frac{\alpha_\mu \nu + M \bar{x}}{\alpha_\mu + M} & \bar{x} &= \frac{1}{M} \sum_m x[m] & T' &= T + M S_M + \frac{\alpha_\mu M}{\alpha_\mu + M} (\nu - \bar{x})(\nu - \bar{x})^T \\ \alpha_\mu' &= \alpha_\mu + M & \alpha_w' &= \alpha_w + M & S_M &= \frac{1}{M} \sum_m (x[m] - \bar{x})(x[m] - \bar{x})^T \end{aligned}$$

Last Time II

Gaussian Learning Take Home Lesson

Prior and marginal are complicated

BUT, the formulas for updating the parameters of the NW, $P\{D\}$, $P\{X|X_1 \dots X_N\}$, MAP are closed form.

Function of sufficient statistics (surprise...)

Extensions

Probably works for all distribution families (need conjugate for BDe)

Define model selection criterion

$P\{D\}$ and MDL penalty, if applicable.

Define ML or MAP estimate for distribution.

Dynamic Belief Networks

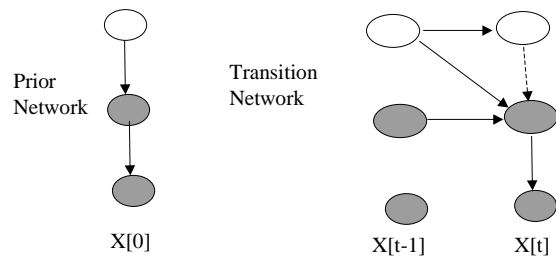
Two Topics:

General DBN Learning

Linear Dynamic Systems

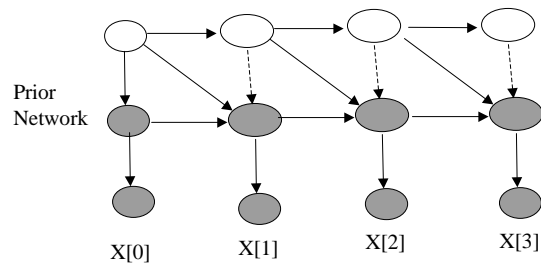
Important special case.

DBN Review:



DBN Review, cont'd

Unrolled Network



Special Networks

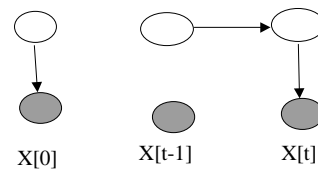
HMM:

Discrete MM hidden state

Continuous or discrete observations

Linear Dynamic System

Linear gaussian state and observation dist'ns



Learning in discrete DBNs

[Friedman, Murphy, and Russell; UAI-98]

Input:

N_{seq} sequences, l th sequence is of length N_l

Output:

Transition network

Prior network

Notation

Variable i , time slice t , sequence l : $x_i^l[t]$

Dirichlet parameter for prior network:

$$\theta_{i,j,k}^{(0)} = P\{X_i[0]=k \mid Pa_i = j\}$$

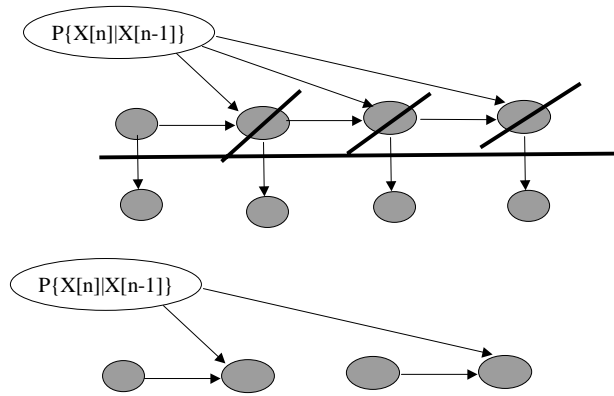
Dirichlet parameter for transition network:

$$\theta_{i,j,k}^{\rightarrow} = P\{X_i[t]=k \mid (Pa_i[t])=j\}$$

Sufficient statistics: $N_{i,j,k}^{(0)} = \sum I[X_i[0]=k, Pa_i = j; X^l]$

$$N_{i,j,k}^{\rightarrow} = \sum_l \sum_t I[X_i[t]=k, Pa_i = j; X^l]$$

Expected answer for complete data



Likelihood decomposes

Answer is identical to normal BN learning.

Probability

$$P\{D | G, \Theta\} = \prod_i \prod_j \prod_k (\theta_{i,j,k}^{(0)})^{N_{i,j,k}^{(0)}} \times \prod_i \prod_j \prod_k (\theta_{i,j,k}^{\rightarrow})^{N_{i,j,k}^{\rightarrow}}$$

Log likelihood

$$L(G : D) = \sum_i \sum_j \sum_k N_{i,j,k}^{(0)} \log \theta_{i,j,k}^{(0)} + \sum_i \sum_j \sum_k N_{i,j,k}^{\rightarrow} \log \theta_{i,j,k}^{\rightarrow}$$

ML estimate

$$\hat{\theta}_{i,j,k}^{(0)} = \frac{N_{i,j,k}^{(0)}}{\sum_k N_{i,j,k}^{(0)}} \quad \hat{\theta}_{i,j,k}^{\rightarrow} = \frac{N_{i,j,k}^{\rightarrow}}{\sum_k N_{i,j,k}^{\rightarrow}}$$

BIC

$$BIC(G : D) = BIC_0 + BIC_{\rightarrow}$$

$$BIC_0 = \sum_i \sum_j \sum_k N_{i,j,k}^{(0)} \log \theta_{i,j,k}^{(0)} + \frac{\log N_{seq}}{2} \#G_0$$

$$BIC_{\rightarrow} = \sum_i \sum_j \sum_k N_{i,j,k}^{\rightarrow} \log \theta_{i,j,k}^{\rightarrow} + \frac{\log N}{2} \#G_{\rightarrow}$$

BDe

Same as normal BDe:

Network and count for prior network

Network and count for transition network

$$\alpha_{i,j,k}^{(0)} = \hat{N}^{(0)} \times P_{B_0} \{X_i[0] = k \mid Pa_i[0] = j\}$$

$$\alpha_{i,j,k}^{\rightarrow} = \hat{N}^{\rightarrow} \times P_{B_0} \{X_i[t] = k \mid (Pa_i[t]) = j\}$$

For both MDL and BDe, best net for B_0 is independent of the best net for B_{\rightarrow}

Learning with incomplete data

This is just the same old algorithm...

SEM:

- Use the EM algorithm to complete the data.
- Determine the expected sufficient statistics.
- Select the ML or MAP score for the expected sufficient statistics.
- Search over structures for $(B^{(0)}, B^{\rightarrow})$

If HMM,

- This is the Baum-Welch algorithm

BATmobile

Generated data that would be seen by a camera overlooking a highway.

- 3500 vehicles simulated for 40-70 time steps.
- Train on 250 - 1500 vehicles and test on 2000.
- Learned decision tree CPTs using both BIC and BDe

Learned Models

reldist: distance to car in front.

relspeed: relative speed to car in front

leftblocked
(rightblocked): car is to the left (right).

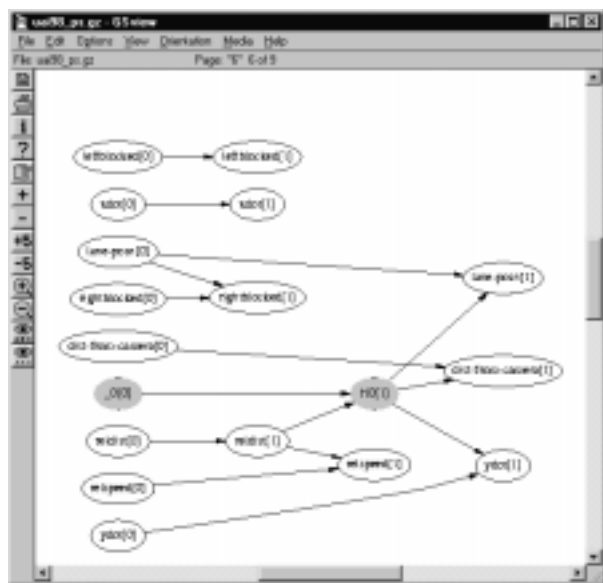
xdot: left-right velocity

ydot: forward velocity or forward acceleration (not relative)



Hidden State

Hidden node has the interpretation "avoid the car in front"

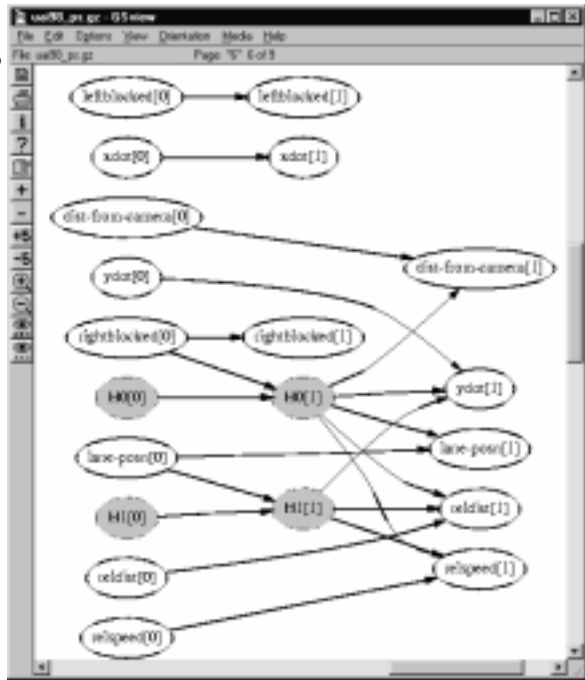


Two hidden nodes

Possible:

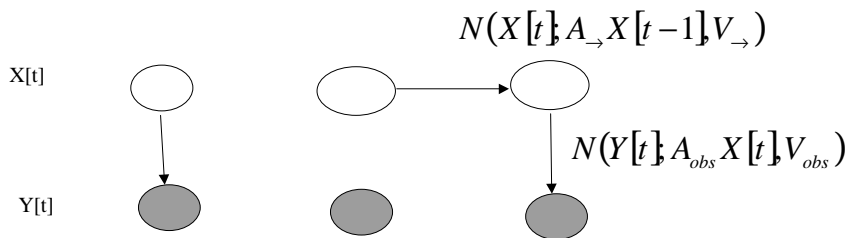
distance and relative speed policy as a function of lane.

lane and speed policy as a function of the car on the right.



Linear Dynamic Systems

[Roweis and Ghahramani, 1997]



Learning

[Shumway + Stoffer, 82; Ghahramani + Hinton, 96]

$$\alpha = \sum_t Y[t]Y^T[t]$$

E-Step:

Run b-net inference algorithm to estimate the mean and variance for each X: $\hat{m}[t], \hat{V}[t]$

M-Step: $\delta = \sum_{t=0, n} Y[t]\hat{X}^T[t]$

$$\gamma = \sum_{t=1, n-1} \gamma_t \qquad \gamma_t = \hat{X}[t]\hat{X}^T[t] + \hat{V}[t]$$

$$\beta = \sum_{t=0}^n \hat{X}[t]\hat{X}^T[t-1] + \hat{V}[t, t-1]$$

$$A_{obs} = \delta(\gamma + \gamma_0 + \gamma_n)^{-1} \qquad A_{\rightarrow} = \beta(\gamma + \gamma_0)^{-1} \qquad X_0 = \hat{X}[0]$$

$$V_{obs} = \frac{\alpha - A_{obs}\delta^T}{N} \qquad V_{\rightarrow} = \frac{(\gamma + \gamma_n - A_{\rightarrow}\beta^T)}{N-1} \qquad V_0 = \hat{V}[0]$$

$$J[t-1] = V[t-1 | D_{t-1}] A_{\rightarrow}^T V^{-1}[t | D_{t-1}]$$

$$\hat{V}[t, t-1] = \hat{V}[t] J[t-1] + J[t-1] (\hat{V}[t] - \hat{V}[t-1]) J^T[t-1]$$

