

Readings:

K&F: 16.2

Jordan Chapter 20

K&F: 4.5, 12.2, 12.3

EM for BNs Kalman Filters Gaussian BNs

Graphical Models – 10708

Carlos Guestrin

Carnegie Mellon University

November 17th, 2006

1

Thus far, fully supervised learning

- We have assumed fully supervised learning:

$$\langle F=t, A=f, S=t, H=t, R=f \rangle$$

⋮

- Many real problems have missing data:

$$\langle F=?, A=?, S=t, H=t, R=f \rangle$$

⋮

The general learning problem with missing data

- Marginal likelihood – \mathbf{x} is observed, \mathbf{z} is missing:

$$\ell(\theta : \mathcal{D}) = \log \prod_{j=1}^m P(\mathbf{x}_j | \theta)$$

$$= \sum_{j=1}^m \log P(\mathbf{x}_j | \theta)$$

$$= \sum_{j=1}^m \log \sum_{\mathbf{z}} P(\mathbf{x}_j, \mathbf{z} | \theta)$$

Sum out (average) over vals of \mathbf{z}

10.708 – ©Carlos Guestrin 2006

3

E-step

- \mathbf{x} is observed, \mathbf{z} is missing
- Compute probability of missing data given current choice of θ
 - $Q(\mathbf{z} | \mathbf{x}_j)$ for each \mathbf{x}_j
 - e.g., probability computed during classification step
 - corresponds to “classification step” in K-means

$$Q^{(t+1)}(\mathbf{z} | \mathbf{x}_j) = P(\mathbf{z} | \mathbf{x}_j, \theta^{(t)})$$

estimate of missing values

$P(A=t | S=t, H=t, R=f, \theta^{(t)})$

10.708 – ©Carlos Guestrin 2006

4

Jensen's inequality

log = *chain rule*
 $1 = \log_e 2 = \log_e (1+1) \quad ? \quad \log(1+b) = 0$

$$\ell(\theta : \mathcal{D}) = \sum_{j=1}^m \log \sum_{\mathbf{z}} P(\mathbf{z} | \mathbf{x}_j) P(\mathbf{x}_j | \theta)$$

■ **Theorem:** $\log \sum_{\mathbf{z}} P(\mathbf{z}) f(\mathbf{z}) \geq \sum_{\mathbf{z}} P(\mathbf{z}) \log f(\mathbf{z})$

Applying Jensen's inequality

■ Use: $\log \sum_{\mathbf{z}} P(\mathbf{z}) f(\mathbf{z}) \geq \sum_{\mathbf{z}} P(\mathbf{z}) \log f(\mathbf{z})$

log a = log a - log b *Est: Q^{(t+1)}*

$$\ell(\theta^{(t)} : \mathcal{D}) = \sum_{j=1}^m \log \sum_{\mathbf{z}} Q^{(t+1)}(\mathbf{z} | \mathbf{x}_j) \frac{P(\mathbf{z}, \mathbf{x}_j | \theta^{(t)})}{Q^{(t+1)}(\mathbf{z} | \mathbf{x}_j)}$$

$$\geq \sum_{j=1}^m \sum_{\mathbf{z}} Q^{(t+1)}(\mathbf{z} | \mathbf{x}_j) \log \frac{P(\mathbf{z}, \mathbf{x}_j | \theta^{(t)})}{Q^{(t+1)}(\mathbf{z} | \mathbf{x}_j)}$$

$$= \underbrace{\sum_{j=1}^m \sum_{\mathbf{z}} Q^{(t+1)}(\mathbf{z} | \mathbf{x}_j) \log P(\mathbf{z}, \mathbf{x}_j | \theta^{(t)})}_{\text{very similar to MLE for BN}} - \underbrace{\sum_{j=1}^m \sum_{\mathbf{z}} Q^{(t+1)}(\mathbf{z} | \mathbf{x}_j) \log Q^{(t+1)}(\mathbf{z} | \mathbf{x}_j)}_{= m \cdot H_{Q^{(t+1)}}(\mathbf{z} | \mathbf{x})}$$

The M-step maximizes lower bound on weighted data

- Lower bound from Jensen's:

$$\ell(\theta^{(t)} : \mathcal{D}) \geq \sum_{j=1}^m \sum_{\mathbf{z}} Q^{(t+1)}(\mathbf{z} | \mathbf{x}_j) \log P(\mathbf{z}, \mathbf{x}_j | \theta^{(t)}) + H(Q^{(t+1)})$$

Handwritten notes:
 E step: estimate $Q^{(t+1)}(\mathbf{z} | \mathbf{x})$
 M step: fix $Q^{(t+1)}$, max θ
 if data were observed $\sum_{j=1}^m \log P(\mathbf{z}_j, \mathbf{x}_j | \theta)$ (max in M step)
 $H(Q^{(t+1)})$ (constant in M step)

- Corresponds to weighted dataset:

- $\langle \mathbf{x}_1, \mathbf{z}=1 \rangle$ with weight $Q^{(t+1)}(\mathbf{z}=1 | \mathbf{x}_1)$
- $\langle \mathbf{x}_1, \mathbf{z}=2 \rangle$ with weight $Q^{(t+1)}(\mathbf{z}=2 | \mathbf{x}_1)$
- $\langle \mathbf{x}_1, \mathbf{z}=3 \rangle$ with weight $Q^{(t+1)}(\mathbf{z}=3 | \mathbf{x}_1)$
- $\langle \mathbf{x}_2, \mathbf{z}=1 \rangle$ with weight $Q^{(t+1)}(\mathbf{z}=1 | \mathbf{x}_2)$
- $\langle \mathbf{x}_2, \mathbf{z}=2 \rangle$ with weight $Q^{(t+1)}(\mathbf{z}=2 | \mathbf{x}_2)$
- $\langle \mathbf{x}_2, \mathbf{z}=3 \rangle$ with weight $Q^{(t+1)}(\mathbf{z}=3 | \mathbf{x}_2)$
- ...

10-708 - ©Carlos Guestrin 2006

7

The M-step

$$\ell(\theta^{(t)} : \mathcal{D}) \geq \sum_{j=1}^m \sum_{\mathbf{z}} Q^{(t+1)}(\mathbf{z} | \mathbf{x}_j) \log P(\mathbf{z}, \mathbf{x}_j | \theta^{(t)}) + H(Q^{(t+1)})$$

- Maximization step:

$$\theta^{(t+1)} \leftarrow \arg \max_{\theta} \sum_{j=1}^m \sum_{\mathbf{z}} Q^{(t+1)}(\mathbf{z} | \mathbf{x}_j) \log P(\mathbf{z}, \mathbf{x}_j | \theta)$$

Handwritten notes:
 weighted MLE
 $\arg \max_{\theta} m \sum_{\mathbf{x}} \hat{p}(\mathbf{x}) \sum_{\mathbf{z}} Q^{(t+1)}(\mathbf{z} | \mathbf{x}) \log P(\mathbf{z}, \mathbf{x} | \theta)$

- Use expected counts instead of counts:

- If learning requires Count(x,z)
- Use $E_{Q^{(t+1)}}[\text{Count}(\mathbf{x}, \mathbf{z})]$

10-708 - ©Carlos Guestrin 2006

8

Convergence of EM

- Define potential function $F(\theta, Q)$:

$$\ell(\theta : \mathcal{D}) \geq F(\theta, Q) = \sum_{j=1}^m \sum_{\mathbf{z}} Q(\mathbf{z} | \mathbf{x}_j) \log \frac{P(\mathbf{z}, \mathbf{x}_j | \theta)}{Q(\mathbf{z} | \mathbf{x}_j)}$$

↑ maximize in M step
E step: max over Q

- EM corresponds to coordinate ascent on F
 - Thus, maximizes lower bound on marginal log likelihood
 - As seen in Machine Learning class last semester

10.708 - ©Carlos Guestrin 2006

9

Announcements

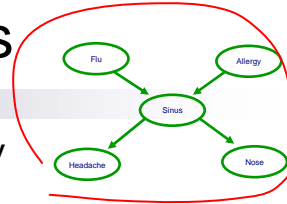
- Lectures the rest of the semester:
 - **Special time: Monday Nov 20 - 5:30-7pm, Wean 4615A:** Gaussian GMs & Kalman Filters
 - **Special time: Monday Nov 27 - 5:30-7pm, Wean 4615A:** Dynamic BNs
 - Wed. 11/30, regular class time: Causality (Richard Scheines)
 - Friday 12/1, regular class time: Finish Dynamic BNs & Overview of Advanced Topics
- Deadlines & Presentations:
 - Project Poster Presentations: Dec. 1st 3-6pm (NSH Atrium) *popular vote for best poster*
 - Project write up: Dec. 8th by 2pm by email
 - 8 pages – limit will be strictly enforced
 - Final: Out Dec. 1st, Due Dec. 15th by 2pm (strict deadline)

10.708 - ©Carlos Guestrin 2006

10

Data likelihood for BNs

- Given structure, log likelihood of fully observed data:



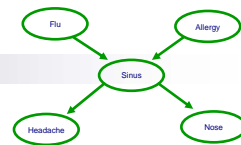
$$\log P(\mathcal{D} \mid \theta_{\mathcal{G}}, \mathcal{G})$$

$$= \sum_{j \in \mathcal{D}} \log P(F^j) P(A^j) P(S^j | F^j, A^j) \cdot P(h^j | s^j) \cdot P(n^j | s^j)$$

$$= \sum_i (\log(f) + \log(a) + \log(p(s | f, a)) + \dots)$$

Marginal likelihood

- What if S is hidden?



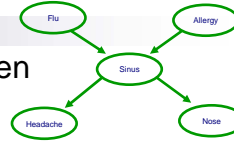
$$\log P(\mathcal{D} \mid \theta_{\mathcal{G}}, \mathcal{G}) = \sum_i \log \left[\sum_s p(f^i) \cdot P(a^i) \cdot P(s | a^i, f^i) P(h^i | s) P(n^i | s) \right]$$

doesn't decompose !!

Log likelihood for BNs with hidden data

- Marginal likelihood – **O** is observed, **H** is hidden

$$\begin{aligned} \ell(\theta : \mathcal{D}) &= \sum_{j=1}^m \log P(\mathbf{o}^{(j)} | \theta) \\ &= \sum_{j=1}^m \log \sum_{\mathbf{h}} P(\mathbf{h}, \mathbf{o}^{(j)} | \theta) \end{aligned}$$



E-step for BNs

- E-step computes probability of hidden vars **h** given **o**

$$Q^{(t+1)}(\mathbf{h} | \mathbf{o}) = P(\mathbf{h} | \mathbf{o}, \theta^{(t)})$$

compute w. inference

- Corresponds to inference in BN



The M-step for BNs

- Maximization step:

$$\theta^{(t+1)} \leftarrow \arg \max_{\theta} \sum_{\mathbf{x}} Q^{(t+1)}(\mathbf{h} | \mathbf{o}) \log P(\mathbf{h}, \mathbf{o} | \theta)$$

$$= \arg \max_{\theta} \sum_{i \in \mathcal{I}} \sum_{\mathbf{x}_i} Q^{(t+1)}(x_i, \mathbf{p}_{x_i} | \theta) \log P(x_i | \mathbf{p}_{x_i})$$

- Use expected counts instead of counts:
 - If learning requires $\text{Count}(\mathbf{h}, \mathbf{o})$
 - Use $E_{Q^{(t+1)}}[\text{Count}(\mathbf{h}, \mathbf{o})]$

M-step for each CPT

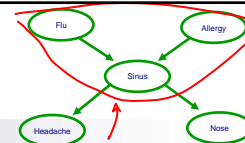
- M-step decomposes per CPT

- Standard MLE:

$$P(X_i = x_i | \text{Pa}_{X_i} = \mathbf{z}) = \frac{\text{Count}(X_i = x_i, \text{Pa}_{X_i} = \mathbf{z})}{\text{Count}(\text{Pa}_{X_i} = \mathbf{z})}$$

- M-step uses expected counts:

$$P(X_i = x_i | \text{Pa}_{X_i} = \mathbf{z}) = \frac{\text{ExCount}(X_i = x_i, \text{Pa}_{X_i} = \mathbf{z})}{\text{ExCount}(\text{Pa}_{X_i} = \mathbf{z})}$$



$P(s | f, a)$

when all data is observed: F, A hidden

$\text{ExCount}[S=s, F=f, A=a]$

$s=t, F=t, A=f$

$$= \sum_{i=1}^m \mathbb{1}(S^i=t) x_i$$

$Q^{(t+1)}(F=t, A=f | \mathbf{o}^{(i)})$

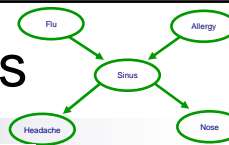
$\mathbf{o}^{(i)}$

$Q^{(t+1)}(F=t, A=f | \mathbf{o}^{(i)})$

$= P(F=t, A=f | \mathbf{o}^{(i)}, \theta^{(t)})$

use VE, Loopy, ...

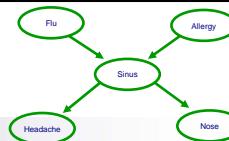
Computing expected counts



$$P(X_i = x_i | \mathbf{Pa}_{X_i} = \mathbf{z}) = \frac{\text{ExCount}(X_i = x_i, \mathbf{Pa}_{X_i} = \mathbf{z})}{\text{ExCount}(\mathbf{Pa}_{X_i} = \mathbf{z})}$$

- M-step requires expected counts:
 - For a set of vars \mathbf{A} , must compute $\text{ExCount}(\mathbf{A}=\mathbf{a})$
 - Some of \mathbf{A} in example j will be observed
 - denote by $\mathbf{A}_O = \mathbf{a}_O^{(j)}$
 - Some of \mathbf{A} will be hidden
 - denote by \mathbf{A}_H
- Use inference (E-step computes expected counts):
 - $\text{ExCount}^{(t+1)}(\mathbf{A}_O = \mathbf{a}_O, \mathbf{A}_H = \mathbf{a}_H) \leftarrow P(\mathbf{A}_H = \mathbf{a}_H | \mathbf{A}_O = \mathbf{a}_O, \theta^{(t)})$
 $\sum_{i=1}^m \mathbb{I}(\mathbf{A}_O^{(i)} = \mathbf{a}_O) \cdot P(\mathbf{A}_H = \mathbf{a}_H | \mathbf{A}_O^{(i)}, \theta^{(t)})$

Data need not be hidden in the same way

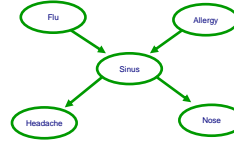


- When data is fully observed
 - A data point is $\langle x_1 = t, x_2 = f, x_3 = t \rangle$
 $\langle x_1 = f, x_2 = f, x_3 = t \rangle$
- When data is partially observed
 - A data point is $\langle x_1 = ?, x_2 = f, x_3 = ? \rangle$
 $\langle x_1 = f, x_2 = ?, x_3 = ? \rangle$
- But unobserved variables can be different for different data points
 - e.g.,
- Same framework, just change definition of expected counts
 - $\text{ExCount}^{(t+1)}(\mathbf{A}_O = \mathbf{a}_O, \mathbf{A}_H = \mathbf{a}_H) \leftarrow P(\mathbf{A}_H = \mathbf{a}_H | \mathbf{A}_O = \mathbf{a}_O, \theta^{(t)})$
 $\sum_{i=1}^m \mathbb{I}(\mathbf{A}_O^{(i)} = \mathbf{a}_O) P(\mathbf{A}_H^{(i)} = \mathbf{a}_H | \mathbf{A}_O^{(i)}, \theta^{(t)})$
*in data point j
Hj are hidden
Oj observed*

Learning structure with missing data

[K&F 16.6]

- Known BN structure: Use expected counts, learning algorithm doesn't change
- Unknown BN structure:
 - Can use expected counts and score model as when we talked about structure learning
 - But, very slow...
 - e.g., greedy algorithm would need to redo inference for every edge we test...
- (Much Faster) **Structure-EM**: Iterate:
 - compute expected counts
 - do a some structure search (e.g., many greedy steps)
 - repeat
- **Theorem**: Converges to local optima of marginal log-likelihood
 - details in the book



10.708 - ©Carlos Guestrin 2006

19

What you need to know about learning with missing data

- EM for Bayes Nets
- E-step: inference computes expected counts
 - Only need expected counts over X_i and \mathbf{Pa}_{x_i}
- M-step: expected counts used to estimate parameters
- Which variables are hidden can change per datapoint
 - Also, use labeled and unlabeled data → some data points are complete, some include hidden variables
- Structure-EM:
 - iterate between computing expected counts & many structure search steps

MNs, EM
Simple...
See reading
 $\max P(X, Y)$

CRF
X ← observed
Y ← hidden
class
pixels
 $\max P(Y | X, \theta)$

EM
 $\max P(\theta)$

CRF
model
doesn't
have
 $P(\theta)$

10.708 - ©Carlos Guestrin 2006

20

Adventures of our BN hero

- Compact representation for probability distributions
- Fast inference
- Fast learning
- Approximate inference
- But... Who are the most popular kids?

1. Naïve Bayes
2 and 3. Hidden Markov models (HMMs)
Kalman Filters

Handwritten notes:
→ hidden
↑ discrete
↑ continuous Gaussian

21

The Kalman Filter

- An HMM with Gaussian distributions
- Has been around for at least 50 years
- Possibly the most used graphical model ever
- It's what
 - does your cruise control
 - tracks missiles
 - controls robots
 - ...
- And it's so simple...
 - Possibly explaining why it's so used
- Many interesting models build on it...
 - An example of a Gaussian BN (more on this later)

22

Example of KF – SLAT

Simultaneous Localization and Tracking

[Funiak, Guestrin, Paskin, Sukthankar '06]

- Place some cameras around an environment, don't know where they are
- Could measure all locations, but requires lots of grad. student (Stano) time
- Intuition:
 - A person walks around
 - If camera 1 sees person, then camera 2 sees person, learn about relative positions of cameras

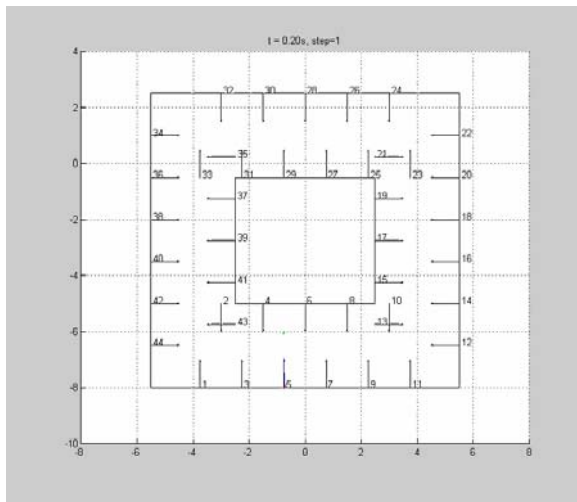


23

Example of KF – SLAT

Simultaneous Localization and Tracking

[Funiak, Guestrin, Paskin, Sukthankar '06]



$X_t \leftarrow$ position
 $C_i \leftarrow$ cameras location
 \vdots
 $P(C_i | m_{1:t})$
 $P(X_t, C | m_{1:t})$
 \uparrow 2dim \uparrow #cams x 3

24

Multivariate Gaussian

$$p(\underbrace{X_1, \dots, X_n}_{=x}) = \frac{1}{\underbrace{(2\pi)^{n/2} |\Sigma|^{1/2}}_{\text{normalizer}}} \exp \left\{ -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu}) \right\}$$

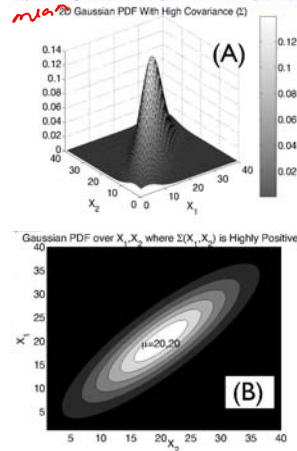
↑ mean
↑ covariance

Mean vector:

$$\boldsymbol{\mu} = \begin{pmatrix} \mu_1 \\ \vdots \\ \mu_n \end{pmatrix}$$

Covariance matrix:

$$\Sigma = \begin{pmatrix} \sigma_{11} & & \\ & \ddots & \\ & & \sigma_{nn} \end{pmatrix}$$



Conditioning a Gaussian

Joint Gaussian:

$$p(X, Y) \sim N(\boldsymbol{\mu}; \Sigma)$$

Conditional linear Gaussian:

$$p(Y|X) \sim N(\mu_{Y|X}; \sigma_{Y|X}^2)$$

$$\mu_{Y|X} = \mu_Y + \frac{\sigma_{YX}}{\sigma_X^2} (x - \mu_X)$$

↑ Gaussian
↑ prior mean
↑ variance

$$\sigma_{Y|X}^2 = \sigma_Y^2 - \frac{\sigma_{YX}^2}{\sigma_X^2}$$

↑ prior variance
↑ non-negative

observation make you more sure

$$\boldsymbol{\mu} = \begin{pmatrix} \mu_X \\ \mu_Y \end{pmatrix} \quad \Sigma = \begin{pmatrix} \sigma_X^2 & \sigma_{XY} \\ \sigma_{YX} & \sigma_Y^2 \end{pmatrix}$$

$$\sigma_{XY} = \sigma_{YX}$$

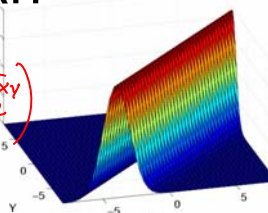
x above mean

& $\sigma_{XY} > 0 \Rightarrow$ positively correlated

$\mu_{Y|X}$ about average

$\mu_{Y|X}$ is a linear function of x

doesn't depend on observed x



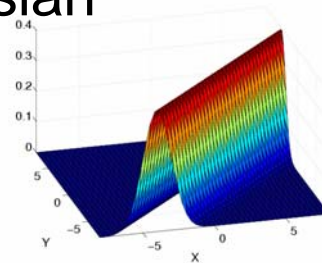
Gaussian is a “Linear Model”

- $\mu_{Y|X} = \mu_Y + \frac{\sigma_{YX}}{\sigma_X^2}(x - \mu_x)$
- Conditional linear Gaussian:
 - $p(Y|X) \sim N(\beta_0 + \beta X; \sigma^2)$
 - $\sigma_{Y|X}^2 = \sigma_Y^2 - \frac{\sigma_{YX}^2}{\sigma_X^2}$

27

Conditioning a Gaussian

- Joint Gaussian:
 - $p(X, Y) \sim N(\mu; \Sigma)$
- Conditional linear Gaussian:
 - $p(Y|X) \sim N(\mu_{Y|X}; \Sigma_{Y|X})$



$$\mu_{Y|X} = \mu_Y + \Sigma_{YX} \Sigma_{XX}^{-1} (x - \mu_x)$$

$$\Sigma_{Y|X} = \Sigma_{YY} - \Sigma_{YX} \Sigma_{XX}^{-1} \Sigma_{XY}$$

28

Conditional Linear Gaussian (CLG) – general case

- Conditional linear Gaussian:

- $p(Y|X) \sim N(\beta_0 + BX; \Sigma_{YY|X})$

$$\mu_{Y|X} = \mu_Y + \Sigma_{YX} \Sigma_{XX}^{-1} (x - \mu_x)$$

$$\Sigma_{YY|X} = \Sigma_{YY} - \Sigma_{YX} \Sigma_{XX}^{-1} \Sigma_{XY}$$

29

Understanding a linear Gaussian – the 2d case

- Variance increases over time
(motion noise adds up)
- Object doesn't necessarily
move in a straight line

30

Tracking with a Gaussian 1

- $p(X_0) \sim N(\mu_0, \Sigma_0)$
- $p(X_{i+1}|X_i) \sim N(B X_i + \beta; \Sigma_{X_{i+1}|X_i})$

31

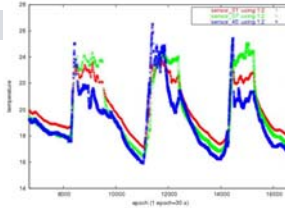
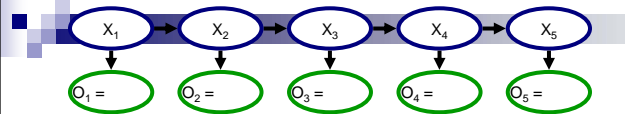
Tracking with Gaussians 2 – Making observations

- We have $p(X_i)$
- Detector observes $O_i=o_i$
- Want to compute $p(X_i|O_i=o_i)$
- Use Bayes rule:

- Require a CLG observation model
 - $p(O_i|X_i) \sim N(W X_i + v; \Sigma_{O_i|X_i})$

32

Operations in Kalman filter



- Compute $p(X_t | O_{1:t} = o_{1:t})$
- Start with $p(X_0)$
- At each time step t
 - **Condition** on observation
 $p(X_t | o_{1:t}) \propto p(X_t | o_{1:t-1})p(o_t | X_t)$
 - **Prediction** (Multiply transition model)
 $p(X_{t+1}, X_t | o_{1:t}) = p(X_{t+1} | X_t)p(X_t | o_{1:t})$
 - **Roll-up** (marginalize previous time step)
 $p(X_{t+1} | o_{1:t}) = \int_{X_t} p(X_{t+1}, x_t | o_{1:t}) dx_t$
- I'll describe one implementation of KF, there are others
 - Information filter

33

Exponential family representation of Gaussian: Canonical Form

$$p(X_1, \dots, X_n) = \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} \exp \left\{ -\frac{1}{2} (\mathbf{x} - \mu)^T \Sigma^{-1} (\mathbf{x} - \mu) \right\}$$

34

Canonical form

$$\begin{aligned}
 p(X_1, \dots, X_n) &= \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} \exp \left\{ -\frac{1}{2} (\mathbf{x} - \mu)^T \Sigma^{-1} (\mathbf{x} - \mu) \right\} \\
 &= K \exp \left\{ \eta^T \mathbf{x} - \frac{1}{2} \mathbf{x}^T \Lambda \mathbf{x} \right\}
 \end{aligned}$$

- Standard form and canonical forms are related:

$$\mu = \Lambda^{-1} \eta$$

$$\Sigma = \Lambda^{-1}$$

- Conditioning is easy in canonical form
- Marginalization easy in standard form

35

Conditioning in canonical form

$$p(X_t | o_{1:t}) \propto p(X_t | o_{1:t-1}) p(o_t | X_t)$$

- First multiply: $p(A, B) = p(A) p(B | A)$

$$p(A) : \eta_1, \Lambda_1$$

$$p(B | A) : \eta_2, \Lambda_2$$

$$p(A, B) : \eta_3 = \eta_1 + \eta_2, \Lambda_3 = \Lambda_1 + \Lambda_2$$

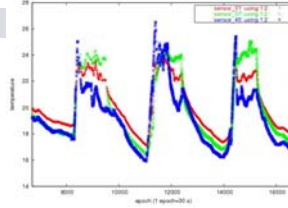
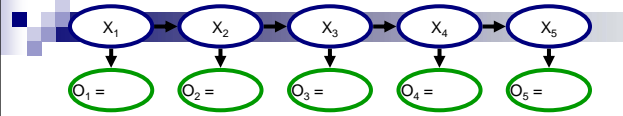
- Then, condition on value $B = y$ $p(A | B = y)$

$$\eta_{A|B=y} = \eta_A - \Lambda_{AB} \cdot y$$

$$\Lambda_{AA|B=y} = \Lambda_{AA}$$

36

Operations in Kalman filter



- Compute $p(X_t | O_{1:t} = o_{1:t})$
- Start with $p(X_0)$
- At each time step t
 - **Condition** on observation

$$p(X_t | o_{1:t}) \propto p(X_t | o_{1:t-1})p(o_t | X_t)$$
 - **Prediction** (Multiply transition model)

$$p(X_{t+1}, X_t | o_{1:t}) = p(X_{t+1} | X_t)p(X_t | o_{1:t})$$
 - **Roll-up** (marginalize previous time step)

$$p(X_{t+1} | o_{1:t}) = \int_{X_t} p(X_{t+1}, x_t | o_{1:t})dx_t$$

37

Prediction & roll-up in canonical form

$$p(X_{t+1} | o_{1:t}) = \int_{X_t} p(X_{t+1} | x_t)p(x_t | o_{1:t})dx_t$$

- First multiply: $p(A, B) = p(A)p(B | A)$

- Then, marginalize X_t : $p(A) = \int_B p(A, b)db$

$$\eta_A^m = \eta_A - \Lambda_{AB}\Lambda_{BB}^{-1}\eta_B$$

$$\Lambda_{AA}^m = \Lambda_{AA} - \Lambda_{AB}\Lambda_{BB}^{-1}\Lambda_{BA}$$

38

What if observations are not CLG?

- Often observations are not CLG
 - CLG if $O_i = B X_i + \beta_o + \varepsilon$
- Consider a motion detector
 - $O_i = 1$ if person is likely to be in the region

 - Posterior is not Gaussian

39

Linearization: incorporating non-linear evidence

- $p(O_i|X_i)$ not CLG, but...
- Find a Gaussian approximation of $p(X_i, O_i) = p(X_i) p(O_i|X_i)$
- Instantiate evidence $O_i = o_i$ and obtain a Gaussian for $p(X_i|O_i = o_i)$

- Why do we hope this would be any good?
 - Locally, Gaussian may be OK

40

Linearization as integration

- Gaussian approximation of $p(X_i, O_i) = p(X_i) p(O_i|X_i)$
- Need to compute moments
 - $E[O_i]$
 - $E[O_i^2]$
 - $E[O_i X_i]$
- Note: Integral is product of a Gaussian with an arbitrary function

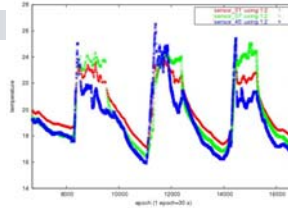
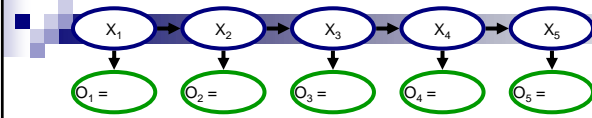
41

Linearization as numerical integration

- **Product of a Gaussian with arbitrary function**
- Effective numerical integration with **Gaussian quadrature** method
 - Approximate integral as **weighted sum over integration points**
 - Gaussian quadrature defines location of points and weights
- Exact if arbitrary function is **polynomial of bounded degree**
- **Number of integration points exponential** in number of dimensions d
- **Exact monomials** requires exponentially fewer points
 - For **$2d+1$ points**, this method is equivalent to effective **Unscented Kalman filter**
 - **Generalizes to many more points**

42

Operations in non-linear Kalman filter



- Compute $p(X_t | O_{1:t} = o_{1:t})$
- Start with $p(X_0)$
- At each time step t
 - **Condition** on observation (use **numerical integration**)

$$p(X_t | o_{1:t}) \propto p(X_t | o_{1:t-1})p(o_t | X_t)$$
 - **Prediction** (Multiply transition model, use **numerical integration**)

$$p(X_{t+1}, X_t | o_{1:t}) = p(X_{t+1} | X_t)p(X_t | o_{1:t})$$
 - **Roll-up** (marginalize previous time step)

$$p(X_{t+1} | o_{1:t}) = \int_{X_t} p(X_{t+1}, x_t | o_{1:t}) dx_t$$

43

What you need to know about Kalman Filters

- **Kalman filter**
 - Probably most used BN
 - Assumes Gaussian distributions
 - Equivalent to linear system
 - Simple matrix operations for computations
- **Non-linear Kalman filter**
 - Usually, observation or motion model not CLG
 - Use numerical integration to find Gaussian approximation

44