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:

- Many real problems have missing data:

The general learning problem with missing data

- Marginal likelihood – **x** is observed, **z** is missing:

$$\begin{aligned}\ell(\theta : \mathcal{D}) &= \log \prod_{j=1}^{m} P(\mathbf{x}_j \mid \theta) \\ &= \sum_{j=1}^{m} \log P(\mathbf{x}_j \mid \theta) \\ &= \sum_{j=1}^{m} \log \sum_{\mathbf{z}} P(\mathbf{x}_j, \mathbf{z} \mid \theta)\end{aligned}$$

E-step

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

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

Jensen's inequality

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

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

Applying Jensen's inequality

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

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

The M-step maximizes lower bound on weighted data

- Lower bound from Jensen's:

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

- Corresponds to weighted dataset:
  - $\langle \mathbf{x}_1, \mathbf{z}=1 \rangle$ with weight $Q^{(t+1)}(\mathbf{z}=1 \mid \mathbf{x}_1)$
  - $\langle \mathbf{x}_1, \mathbf{z}=2 \rangle$ with weight $Q^{(t+1)}(\mathbf{z}=2 \mid \mathbf{x}_1)$
  - $\langle \mathbf{x}_1, \mathbf{z}=3 \rangle$ with weight $Q^{(t+1)}(\mathbf{z}=3 \mid \mathbf{x}_1)$
  - $\langle \mathbf{x}_2, \mathbf{z}=1 \rangle$ with weight $Q^{(t+1)}(\mathbf{z}=1 \mid \mathbf{x}_2)$
  - $\langle \mathbf{x}_2, \mathbf{z}=2 \rangle$ with weight $Q^{(t+1)}(\mathbf{z}=2 \mid \mathbf{x}_2)$
  - $\langle \mathbf{x}_2, \mathbf{z}=3 \rangle$ with weight $Q^{(t+1)}(\mathbf{z}=3 \mid \mathbf{x}_2)$
  - …

The M-step

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

- Maximization step:

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

- Use expected counts instead of counts:
  - If learning requires Count(**x**,**z**)
  - Use $E_{Q(t+1)}$[Count(**x**,**z**)]

Convergence of EM

- Define potential function F(θ,Q):

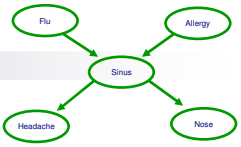
$$\ell(\theta : \mathcal{D}) \;\geq\; F(\theta, Q) = \sum_{j=1}^{m} \sum_{\mathbf{z}} Q(\mathbf{z} \mid \mathbf{x}_j) \log \frac{P(\mathbf{z}, \mathbf{x}_j \mid \theta)}{Q(\mathbf{z} \mid \mathbf{x}_j)}$$

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

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)
  - 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)

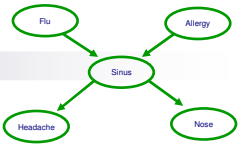
Data likelihood for BNs



- Given structure, log likelihood of fully observed data:

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

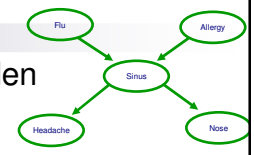
Marginal likelihood



- What if S is hidden?

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

Log likelihood for BNs with hidden data

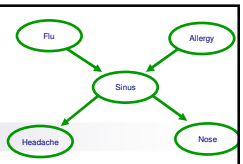


- Marginal likelihood – **O** is observed, **H** is hidden

$$\ell(\theta : \mathcal{D}) = \sum_{j=1}^{m} \log P(\mathbf{o}^{(j)} \mid \theta)$$

$$= \sum_{j=1}^{m} \log \sum_{\mathbf{h}} P(\mathbf{h}, \mathbf{o}^{(j)} \mid \theta)$$

E-step for BNs

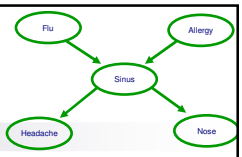


- E-step computes probability of hidden vars **h** given **o**

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

- Corresponds to inference in BN

The M-step for BNs

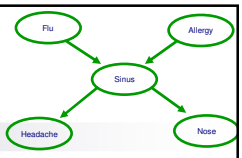


- Maximization step:

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

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

M-step for each CPT



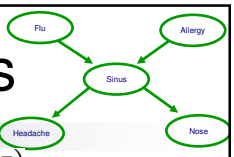
- M-step decomposes per CPT
  - Standard MLE:

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

  - M-step uses expected counts:

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

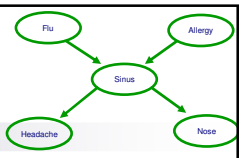
Computing expected counts



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

- M-step requires expected counts:
  - For a set of vars **A**, must compute ExCount(**A**=**a**)
  - Some of **A** in example *j* will be observed
    - denote by $\mathbf{A_O} = \mathbf{a_O}^{(j)}$
  - Some of **A** will be hidden
    - denote by $\mathbf{A_H}$
- Use inference (E-step computes expected counts):
  - $\text{ExCount}^{(t+1)}(\mathbf{A_O} = \mathbf{a_O}^{(j)}, \mathbf{A_H} = \mathbf{a_H}) \leftarrow P(\mathbf{A_H} = \mathbf{a_H} \mid \mathbf{A_O} = \mathbf{a_O}^{(j)}, \boldsymbol{\theta}^{(t)})$

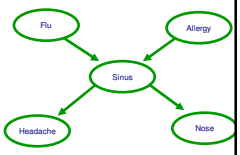
Data need not be hidden in the same way



- When data is fully observed
  - A data point is
- When data is partially observed
  - A data point is
- But unobserved variables can be different for different data points
  - e.g.,
- Same framework, just change definition of expected counts
  - $\text{ExCount}^{(t+1)}(\mathbf{A_O} = \mathbf{a_O}^{(j)}, \mathbf{A_H} = \mathbf{a_H}) \leftarrow P(\mathbf{A_H} = \mathbf{a_H} \mid \mathbf{A_O} = \mathbf{a_O}^{(j)}, \boldsymbol{\theta}^{(t)})$

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

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 $\mathbf{Pa}_{xi}$
- M-step: expected counts used to estimate parameters
- Which variables are hidden can change per datapoint
  - Also, use labeled and unlabeled data $\rightarrow$ some data points are complete, some include hidden variables
- Structure-EM:
  - iterate between computing expected counts & many structure search steps

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

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

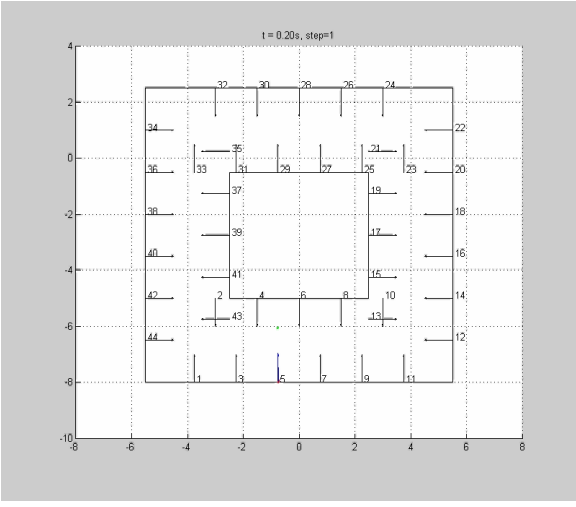
[Funiak, Guestrin, Paskin, Sukthankar '06]

- 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]



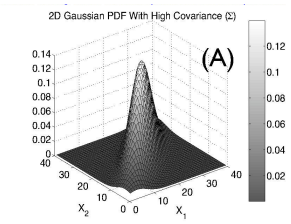
24

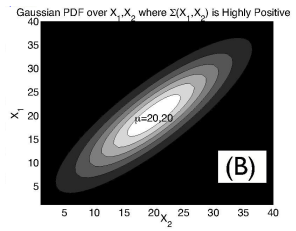
Multivariate Gaussian

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

Mean vector:

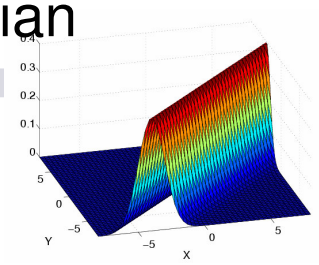
Covariance matrix:





Conditioning a Gaussian

- Joint Gaussian:
  - p(X,Y) ~ *N*(μ;Σ)
- Conditional linear Gaussian:
  - p(Y|X) ~ *N*($\mu_{Y|X}$; σ^2)



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

$$\sigma^2_{Y|X} = \sigma^2_Y - \frac{\sigma^2_{YX}}{\sigma^2_X}$$

Gaussian is a "Linear Model"

- Conditional linear Gaussian:
  - p(Y|X) ~ $N(\beta_0+\beta X; \sigma^2)$

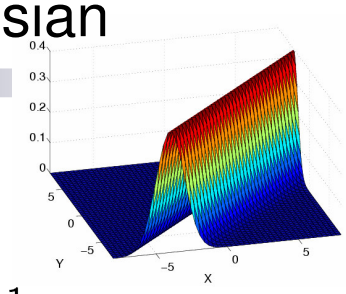
$$\mu_{Y|X} = \mu_Y + \frac{\sigma_{YX}}{\sigma_X^2}(x-\mu_x)$$

$$\sigma^2_{Y|X} = \sigma^2_Y - \frac{\sigma^2_{YX}}{\sigma^2_X}$$

27

Conditioning a Gaussian

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



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

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

28

Conditional Linear Gaussian (CLG) – general case

- Conditional linear Gaussian:
  - p(Y|X) ~ $N(\beta_0+BX; \Sigma_{YY|X})$

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

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

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

31

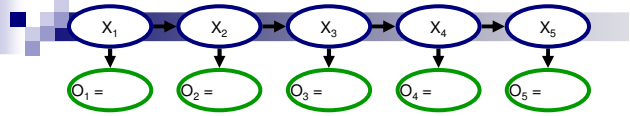
Tracking with Gaussians 2 – Making observations

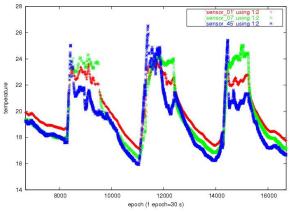
- We have $p(X_i)$
- Detector observes $O_i=o_i$
- Want to compute $p(X_i|O_i=o_i)$
- Use Bayes rule:

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

32

Operations in Kalman filter





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

33

Exponential family representation of Gaussian: Canonical Form

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

34

Canonical form

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

- 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

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

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

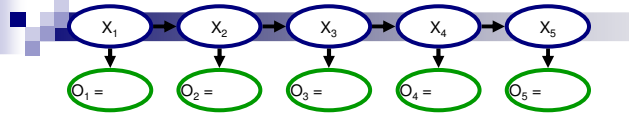
$p(A): \quad \eta_1, \ \Lambda_1$
$p(B \mid A): \quad \eta_2, \ \Lambda_2$
$p(A, B): \quad \eta_3 = \eta_1 + \eta_2, \ \Lambda_3 = \Lambda_1 + \Lambda_2$

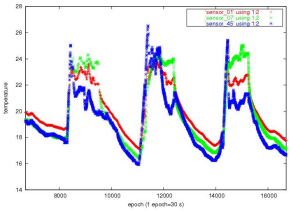
- Then, condition on value B = y $\quad p(A \mid B = y)$

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

36

Operations in Kalman filter





- Compute $p(X_t \mid O_{1:t} = o_{1:t})$
- Start with $p(X_0)$
- At each time step *t*:
  - **Condition** on observation
    $$p(X_t \mid o_{1:t}) \propto p(X_t \mid o_{1:t-1}) p(o_t \mid X_t)$$
  - **Prediction** (Multiply transition model)
    $$p(X_{t+1}, X_t \mid o_{1:t}) = p(X_{t+1} \mid X_t) p(X_t \mid o_{1:t})$$
  - **Roll-up** (marginalize previous time step)
    $$p(X_{t+1} \mid o_{1:t}) = \int_{X_t} p(X_{t+1}, x_t \mid o_{1:t}) dx_t$$

37

Prediction & roll-up in canonical form

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

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

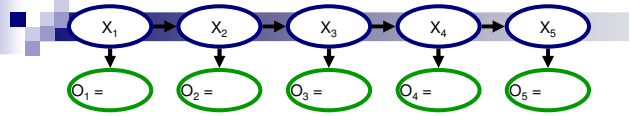
41

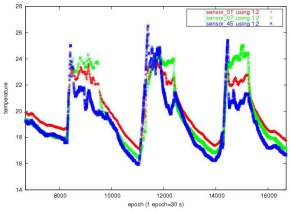
Linearization as numerical integration

- **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 **2*d*+1 points**, this method is equivalent to effective **Unscented Kalman filter**
  - **Generalizes to many more points**

42

Operations in non-linear Kalman filter





- Compute $p(X_t \mid O_{1:t} = o_{1:t})$
- Start with $p(X_0)$
- At each time step *t*:
  - **Condition** on observation (use **numerical integration**)
    $$p(X_t \mid o_{1:t}) \propto p(X_t \mid o_{1:t-1}) p(o_t \mid X_t)$$
  - **Prediction** (Multiply transition model, use **numerical integration**)
    $$p(X_{t+1}, X_t \mid o_{1:t}) = p(X_{t+1} \mid X_t) p(X_t \mid o_{1:t})$$
  - **Roll-up** (marginalize previous time step)
    $$p(X_{t+1} \mid o_{1:t}) = \int_{X_t} p(X_{t+1}, x_t \mid o_{1:t}) dx_t$$

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