Readings:
K&F: 6.1, 6.2, 6.3, 14.1, 14.2, 14.3, 14.4,

Kalman Filters
Gaussian MNs

Graphical Models – 10708
Carlos Guestrin
Carnegie Mellon University

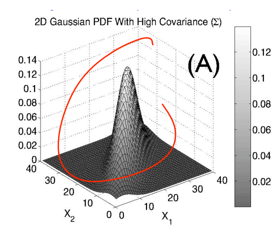
December 1st, 2008

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:

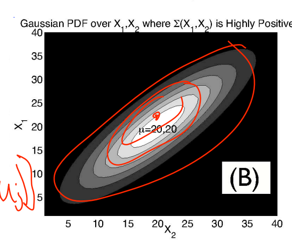
$$\mu = \begin{pmatrix} \mu_1 \\ \vdots \\ \mu_n \end{pmatrix} \qquad \mu_i = E[X_i]$$



Covariance matrix:

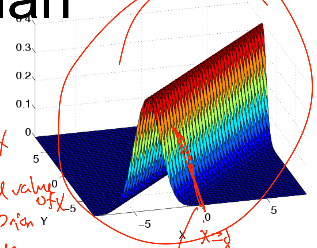
$$\Sigma = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_2^2 & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_3^2 \end{pmatrix} \quad \sigma_{32} = \sigma_{23}$$

$$\sigma_{ij} = E[(X_i - \mu_i)(X_j - \mu_j)]$$



Conditioning a Gaussian

- Joint Gaussian:
  - p(X,Y) ~ $N(\mu;\Sigma)$
- Conditional linear Gaussian:
  - p(Y|X) ~ $N(\mu_{Y|X}; \sigma^2_{Y|X})$ ← gaussian



for each fixed x

x=0

P(Y|X=0)

observed value of x

prior Y mean

prior mean of x

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

↑ prior

$$\underbrace{\sigma^2_{Y|X}}_{\text{posterior variance}} = \underbrace{\sigma^2_Y}_{\text{prior variance}} - \underbrace{\frac{\sigma^2_{YX}}{\sigma^2_X}}_{\geq 0}$$

← posterior variance doesn't depend on observed value!!

$\sigma^2_{Y|X} \leq \sigma^2_Y$ ($\sigma^2_{Y|X} = \sigma^2_Y$ iff $Y \perp X$)

observations always decrease variance

3

Gaussian is a "Linear Model"

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

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

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

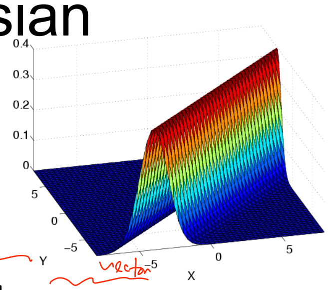
$$\mu_{Y|X} = \underbrace{\mu_Y - \frac{\sigma_{YX}}{\sigma^2_X}\mu_x}_{\beta_0} + \underbrace{\frac{\sigma_{YX}}{\sigma^2_X}}_{\beta} x$$

$$= \beta_0 + \beta x$$

equivalently: $Y = \beta_0 + \beta X + \varepsilon$ ← white Noise $N(0, \sigma^2_{Y|X})$

4

Conditioning a Gaussian



- Joint Gaussian: $\overset{\text{matrix}}{\Sigma} = \begin{pmatrix} \Sigma_{XX} & \Sigma_{XY} \\ \Sigma_{YX} & \Sigma_{YY} \end{pmatrix}$
  - p(X,Y) ~ $N(\mu;\Sigma)$
- Conditional linear Gaussian:
  - p(Y|X) ~ $N(\mu_{Y|X}; \Sigma_{YY|X})$

vector   prior mean vector   vector

$$\mu_{Y|X} = \mu_Y + \Sigma_{YX}\Sigma_{XX}^{-1}(x - \mu_x)$$
$$\underbrace{\Sigma_{YY|X}}_{\text{covariance of the posterior}} = \Sigma_{YY} - \Sigma_{YX}\Sigma_{XX}^{-1}\Sigma_{XY}$$

5

Conditional Linear Gaussian (CLG) – general case

$|X|\,\overset{|X|}{\Sigma_{XX}}$ $|Y|\,\overset{|X|}{\Sigma_{YX}}$ $\overset{1}{\mu_X}$

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

β_0 vector of size $|Y|$ $|Y|\,\overset{|X|}{B}$

$$\mu_{Y|X} = \mu_Y + \Sigma_{YX}\Sigma_{XX}^{-1}(x - \mu_x) = \mu_Y - \Sigma_{YX}\Sigma_{XX}^{-1}\mu_X + \Sigma_{YX}\Sigma_{XX}^{-1}x$$
$$\Sigma_{YY|X} = \Sigma_{YY} - \Sigma_{YX}\Sigma_{XX}^{-1}\Sigma_{XY}$$
$$= \beta_0 + B x$$

$Y = \beta_0 + BX + \varepsilon \leftarrow$ white Noise $N(\vec{0}, \Sigma_{YY|X})$

6

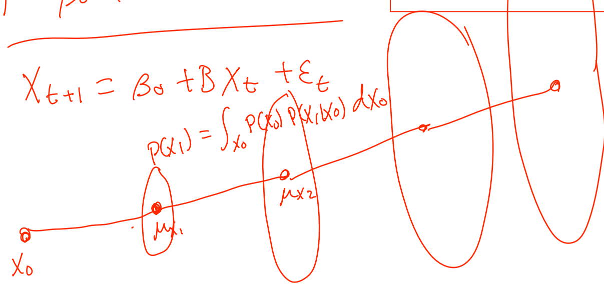
Understanding a linear Gaussian – the 2d case

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

$Y = \beta_0 + BX + \varepsilon$

$X_{t+1} = \beta_0 + B X_t + \varepsilon_t$

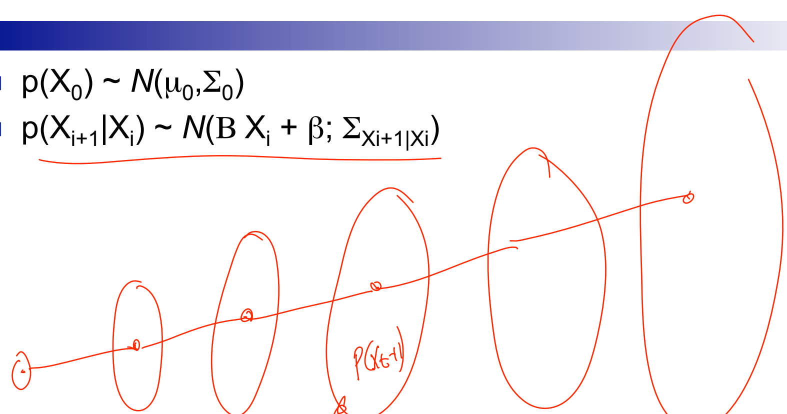
$p(X_1) = \int_{X_0} p(X_0)\, p(X_1 \mid X_0)\, dX_0$



7

Tracking with a Gaussian 1

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



$\int_{X_t} p(X_t)\, p(X_{t+1} \mid X_t)\, dX_t$

8

Tracking with Gaussians 2 – Making observations

Why Gaussians
1. easy
reasonable many apps { 2. central limit thm
3. max ent...

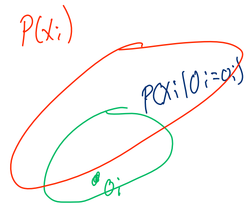
- We have $p(X_i)$ ← prior
- Detector observes $O_i=o_i$ ← observation
- Want to compute $p(X_i|O_i=o_i)$ ← posterior
- Use Bayes rule: $P(X_i|O_i=o_i) \propto P(X_i)\, P(O_i=o_i|X_i)$

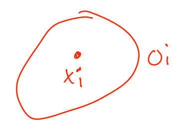
e.g. camera tracking
W → transforms from 3d position X_i to 2d camera obs. O_i

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

if true location is X_i
$O_i = v + W X_i + \varepsilon$ ← $N(0, \Sigma_{O_i|X_i})$
simplest case: unshifted, unbiased, $v = 0$
$W = I$
⇒

$P(X_i)$



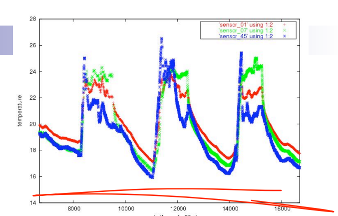


9

Operations in Kalman filter

$X_1 \to X_2 \to X_3 \to X_4 \to X_5$

$O_1=20$, $O_2=21$, $O_3=21$, $O_4=24$, $O_5=27$



- Compute $p(X_t \mid O_{1:t} = o_{1:t})$
- Start with $p(X_0)$
- At each time step t:
  - **Condition** on observation (posterior, prior, likelihood)
    $p(X_t \mid o_{1:t}) \propto p(X_t \mid o_{1:t-1}) p(o_t \mid X_t)$
  - **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

10

Exponential family representation of Gaussian: Canonical Form

$= e^c \cdot e^{f(x)}$

$e^{f(x)+c} \propto e^{f(x)}$

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

$\Lambda \succeq 0$

positive semi definite

$$\propto \exp\left\{-\frac{1}{2} x^T \Sigma^{-1} x + \mu^T \Sigma^{-1} x - \frac{1}{2} \mu^T \Sigma^{-1} \mu\right\}$$

$$\propto \exp\left\{-\frac{1}{2} x^T \Sigma^{-1} x + \mu^T \Sigma^{-1} x\right\}$$

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

$\mu^T \Sigma^{-1} \equiv \eta^T$

$$= \exp\left\{-\frac{1}{2} x^T \Lambda x + \eta^T x\right\}$$

$$= \exp\left\{-\frac{1}{2} \sum_{ij} \lambda_{ij} x_i x_j + \sum_i \eta_i x_i\right\} = \exp\left\{\sum_{ij} \lambda_{ij} f_{ij}(x_i x_j) + \sum_i \eta_i f_i(x_i)\right\}$$

log linear model

features

$f_i(x) = x_i$

$f_{ij}(x) = x_i x_j$

11

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

12

Conditioning in canonical form

$$\underbrace{p(X_t \mid o_{1:t})}_{\text{posterior}} \propto \underbrace{p(X_t \mid o_{1:t-1})}_{\text{prior}}\underbrace{p(o_t \mid X_t)}_{\text{observation}}$$

joint, gaussian, gaussian, CLG

$|A| = k$
$|B| = m$

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

$p(A): \quad \eta_1, \ \Lambda_1 \leftarrow$ $k\ \eta_1$, $k\ \Lambda^k$

$p(B \mid A): \quad \eta_2, \ \Lambda_2 \leftarrow$ $k{+}m\ \eta_2$, $k{+}m\ \Lambda_2^{k+m}$

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

$\Lambda_1 = \begin{pmatrix} \Lambda_{1,0} & 0 \\ 0 & 0 \end{pmatrix}$, $\Lambda_2 = \begin{pmatrix} \cdot & \cdot \\ \cdot & \cdot \end{pmatrix}$

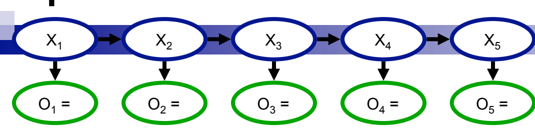
$$\Lambda_3 = \begin{array}{c|cc} & k & m \\ \hline k & \Lambda_{AA} & \Lambda_{AB} \\ m & \Lambda_{BA} & \Lambda_{BB} \end{array} \qquad \eta_3 = \begin{array}{c|c} & 1 \\ \hline k & \eta_A \\ m & \eta_B \end{array}$$

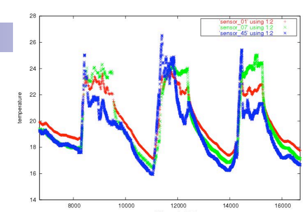
- Then, condition on value B = y $\quad p(A \mid B = y)$

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

13

Operations in Kalman filter





- Compute $\quad p(X_t \mid O_{1:t} = o_{1:t})$
- Start with $p(X_0)$
- At each time step *t*:
  - **Condition** on observation $\leftarrow$ matrix addition / selection of submatrices

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

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

14

Prediction & roll-up in canonical form

transition model CLG — posterior in current timestep ← Gauss

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

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

  same as before

  $\eta = \begin{pmatrix} \eta_A \\ \eta_B \end{pmatrix}$

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

  $\Lambda = \begin{pmatrix} \Lambda_{AA} & \Lambda_{AB} \\ \Lambda_{BA} & \Lambda_{BB} \end{pmatrix}$

marginal

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

marginal

$P(A) = N(\eta^m_A, \Lambda^m_{AA})$

where does $P(X_{t+1} \mid X_t)$, $P(O_t \mid X_t)$ come from? learn from data!! :)

$\frac{P(X_{t+1}, X_t)}{P(X_t)}$ ← $P(X_{t+1} \mid X_t)$ ratio matrix subtraction

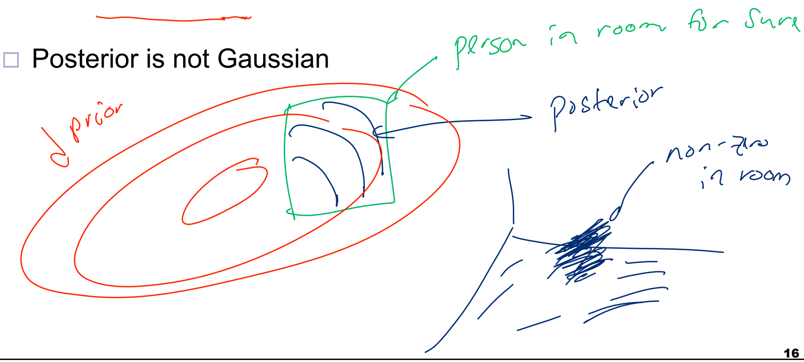
Can also do EM for Kalman filters

15

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

    detector: in room / not in room
  - Posterior is not Gaussian

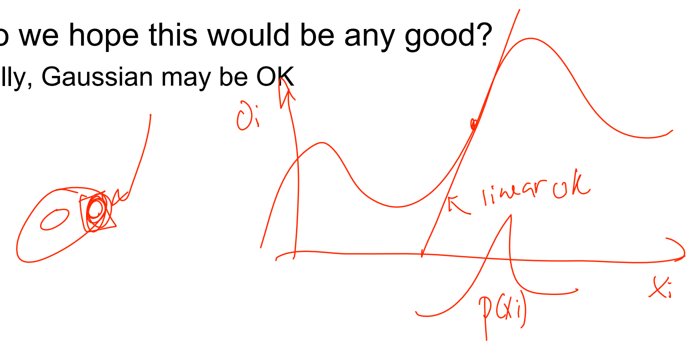


16

Linearization: incorporating non-linear evidence

not for observation model, but for joint } focuses approximation

- $p(O_i|X_i)$ 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



17

Linearization as integration

already know $E[X_i]$, $E[X_i^2]$

- Gaussian approximation of $\underbrace{p(X_i,O_i)}_{\text{mean \& variance}} \overset{\text{gaussian approx}}{=} \underbrace{p(X_i)}_{\text{gaussian}}\, \underbrace{p(O_i|X_i)}_{\text{arbitrary function}}$
- Need to compute moments — plug directly here
  - $E[O_i] = \mu_{o_i} = \int o_i\, p(o_i|x_i)\, p(x_i)\, dx_i\, do_i$
  - $E[O_i^2] = \int o_i^2\, p(o_i|x_i)\, p(x_i)\, dx_i\, do_i$
  - $E[O_i X_i] = \int \underbrace{o_i x_i\, p(o_i|x_i)}_{f(o_i,x_i)}\, \underbrace{p(x_i)}_{\text{Gaussian}}\, dx_i\, do_i$
- Note: Integral is product of a Gaussian with an arbitrary function

18

Linearization as numerical integration

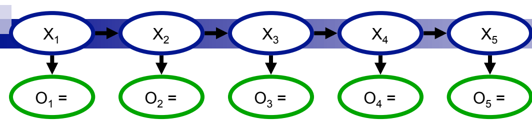
- **Product of a Gaussian with arbitrary function**
- Effective numerical integration with **Gaussian quadrature** method
  - Approximate integral as **weighted sum over integration points**
  - Gaussian quadrature defines location of points and weights
- Exact if arbitrary function is **polynomial of 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**

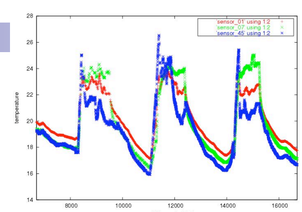
can do this even if $p(O_i|X_i)$ is a black box

extended Kalman filter requires derivative of $p(O_i|X_i)$

19

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)

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

20

Canonical form & Markov Nets

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

$$\propto \exp\left\{-\frac{1}{2}\sum_{ij}\lambda_{ij}x_ix_j + \sum_i \eta_i x_i\right\}$$
$$= \exp\left\{\sum_{ij}\lambda_{ij}\underbrace{f_{ij}(x_i,x_j)}_{\text{edge features}} + \sum_i \eta_i \underbrace{f_i(x_i)}_{\text{node features}}\right\}$$

MN:

graph structure $\leftarrow$ precision matrix $\Lambda \equiv \Sigma^{-1}$ defines graph structure

$\lambda_{ij} = 0$ no edge

What you need to know about Gaussians, Kalman Filters, Gaussian MNs

- **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
- **Gaussian Markov Nets**
  - Sparsity in precision matrix equivalent to graph structure
- **Continuous and discrete (hybrid) model**
  - Much harder, but doable and interesting (see book)