Readings:
K&F: 4.5, 12.2, 12.3, 12.4

# Kalman Filters
# Switching Kalman Filter

Graphical Models – 10708
Carlos Guestrin
Carnegie Mellon University

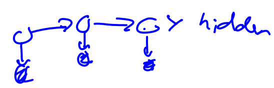
November 20th, 2006

1

## 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**

discrete

continuous
Gaussian

2

# The Kalman Filter

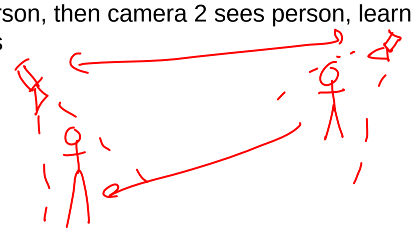
- An HMM with Gaussian distributions
- Has been around for at least 50 years
- Possibly the most used graphical model ever
- It's what
  - does your cruise control
  - tracks missiles
  - controls robots
  - …
- And it's so simple…
  - Possibly explaining why it's so used
- Many interesting models build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

3

# Example of KF – SLAT
## Simultaneous Localization and Tracking

[Funiak, Guestrin, Paskin, Sukthankar '06]

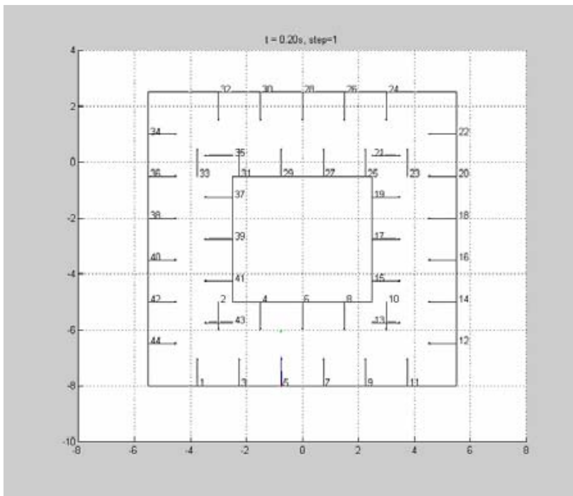
- Place some cameras around an environment, don't know where they are
- Could measure all locations, but requires lots of grad. student (Stano) time
- Intuition:
  - A person walks around
  - If camera 1 sees person, then camera 2 sees person, learn about relative positions of cameras



4

# Example of KF – SLAT
## Simultaneous Localization and Tracking

[Funiak, Guestrin, Paskin, Sukthankar '06]




$X_t \leftarrow$ position

$C_i \leftarrow$ cameras location

$\vdots$

$P(C_i \mid m_{1 \ldots T})$

$P(X_t, C \mid m_{1 \ldots t})$

2dim

#cams x 3



# Multivariate Gaussian

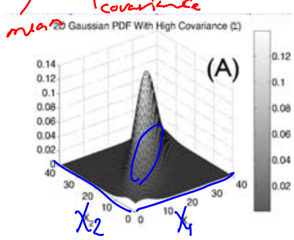
$$\underline{p(X_1, \ldots, \overset{=x}{X_n})} = \underbrace{\frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}}}_{\text{normalizer}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\underset{\text{mean}}{\mu})^T \underset{\text{covariance}}{\Sigma}^{-1}(\mathbf{x}-\mu)\right\}$$

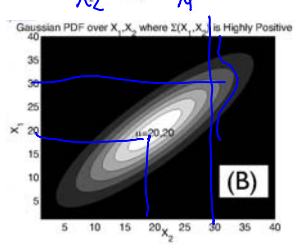
**Mean vector:**

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

**Covariance matrix:**

$$\Sigma = n \begin{pmatrix} & \overset{n}{} & \\ & \sigma_{ij} & \\ & & \end{pmatrix}$$

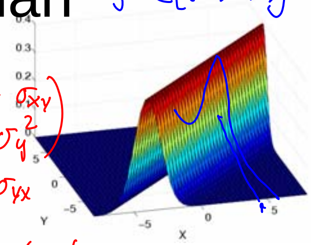


# Conditioning a Gaussian

$\sigma_x^2 = E[(x-\mu_x)^2]$

$\sigma_{xy} = E[(x-\mu_x)(y-\mu_y)]$




- Joint Gaussian: $\mu = \begin{pmatrix}\mu_X \\ \mu_Y\end{pmatrix}$ $\Sigma = \begin{pmatrix}\sigma_X^2 & \sigma_{XY} \\ \sigma_{YX} & \sigma_Y^2\end{pmatrix}$
  - p(X,Y) ~ *N*(μ;Σ)
- Conditional linear Gaussian:
  - p(Y|X) ~ *N*($\mu_{Y|X}$; $\sigma^2_{YX}$)

=20°C

variance

Gaussian

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

$$\underbrace{\mu_{Y|X}}_{\text{posterior mean}} = \underbrace{\mu_Y}_{\text{prior mean}} + \frac{\sigma_{YX}}{\sigma_X^2}(x-\mu_x)$$

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

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

$\mu_{Y|X}$ above average

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

doesn't depend on observed $x$

prior variance

non-negative

observation make you more sure



# Gaussian is a "Linear Model"

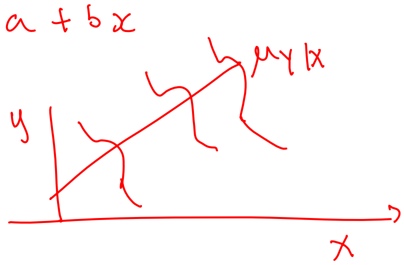
$$\mu_{Y|X} = \mu_Y + \frac{\sigma_{YX}}{\sigma_X^2}(x-\mu_x)$$

$$\sigma^2_{Y|X} = \sigma^2_Y - \underbrace{\frac{\sigma^2_{YX}}{\sigma^2_X}}_{\text{doesn't depend on X}}$$

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

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

$$= a + bx$$



# Conditioning a Gaussian

$\Sigma_{XX}$ cov. matrix for X
$\Sigma_{YY}$ " " Y
$\Sigma_{XY}$ is covar bet X, Y


- Joint Gaussian:
  - p(X,Y) ~ $N(\mu;\Sigma)$

$$\mu = \begin{pmatrix} \mu_X \\ \mu_Y \end{pmatrix} \begin{matrix} \}n \\ \}m \end{matrix} \qquad \Sigma = \begin{array}{c|cc} & n & m \\ \hline n & \Sigma_{XX} & \Sigma_{XY} \\ m & \Sigma_{YX} & \Sigma_{YY} \end{array}$$

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

Conditional dist:

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

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

$$\Sigma_{XY} = \begin{array}{c|c} & m, j \\ \hline n & \begin{pmatrix} \vdots \; \cdots \; \sigma_{x_i y_j} \end{pmatrix} \\ i & \end{array}$$

$$\sigma_{x_i y_j} = E[(X_i - \mu_{x_i})(Y_j - \mu_{y_j})]$$

marginal: $Y \sim N(\mu_Y, \Sigma_{YY})$



# Conditional Linear Gaussian (CLG) – general case

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

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

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

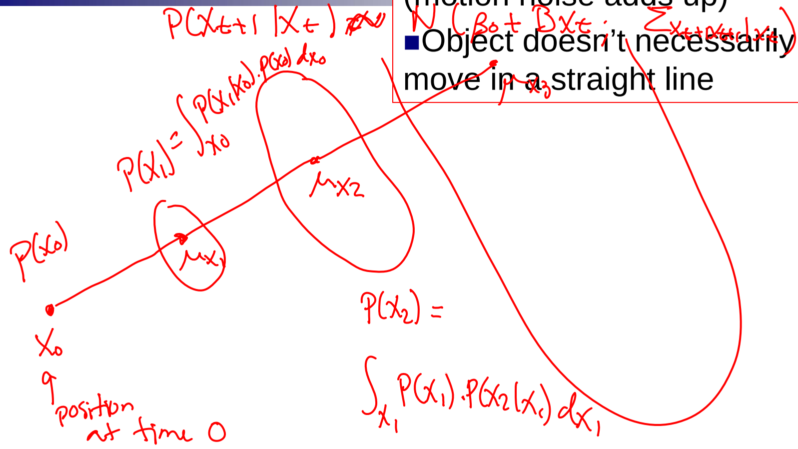
$$\mu_{Y|X} = \mu_Y + \Sigma_{YX}\Sigma_{XX}^{-1}(X - \mu_X) = \beta_0 + BX$$

$$\beta_0 = \mu_Y - \Sigma_{YX}\Sigma_{XX}^{-1}\mu_X$$

$$B = \Sigma_{YX}\Sigma_{XX}^{-1}$$

# Understanding a linear Gaussian – the 2d case

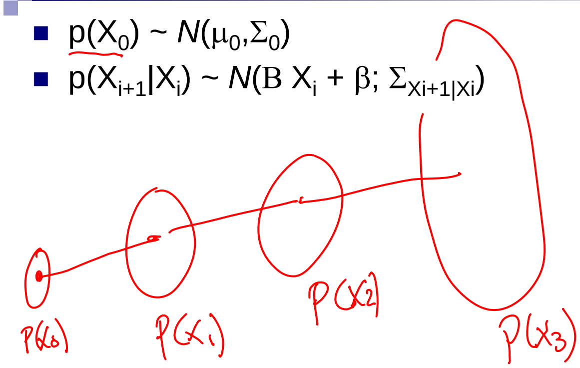
- Variance increases over time (motion noise adds up)
- Object doesn't necessarily move in a straight line






# Tracking with a Gaussian 1

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



# Tracking with Gaussians 2 – Making observations

- We have $p(X_i)$ $\sim N(\mu_{X_i}; \Sigma_{X_iX_i})$
- Detector observes $O_i=o_i$
- Want to compute $p(X_i|O_i=o_i)$
- Use Bayes rule:

$$\underbrace{P(X_i|O_i=o_i)}_{\text{posterior}} \propto \underbrace{P(X_i)}_{\text{prior}} \cdot \underbrace{P(O_i=o_i|X_i)}_{\text{likelihood}}$$

- Require a CLG observation model
  - $p(O_i|X_i) \sim N(\underbrace{W X_i}_{\text{linear function of X}} + \underbrace{v}_{\text{gaussian noise}}; \Sigma_{O_i|X_i})$

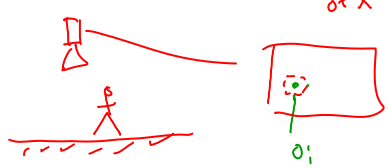
$T_i$ true temp

$M_i$ sensor

$M_i = a T_i + b + \varepsilon$

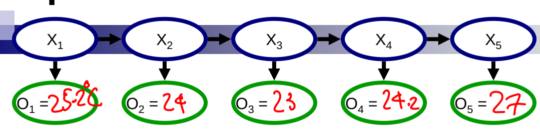
$a$, $b$: calibration params

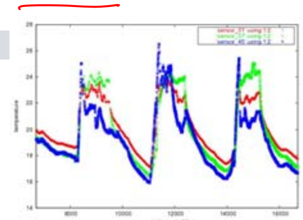
$\varepsilon \sim N(0; \sigma^2)$




13

# Operations in Kalman filter



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

$P(X_t|o_{1..t-1})$ → condition → $P(X_t|o_{1..t})$ → prediction → $P(X_{t+1},X_t|o_{1..t})$ → roll up → $P(X_{t+1}|o_{1..t})$

- I'll describe one implementation of KF, there are others
  - Information filter

14

## Exponential family representation of Gaussian: Canonical Form

$e^{a+b} = e^a \cdot e^b$

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

$$\exp\{-\tfrac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\} = \exp\left[-\tfrac{1}{2}\left[\underbrace{x^T\Sigma^{-1}x}_{\text{quadratic in } x} - 2\underbrace{\mu^T\Sigma^{-1}x}_{\text{linear in } x} + \underbrace{\mu^T\Sigma^{-1}\mu}_{\text{constant (doesn't dep. on } x)}\right]\right]$$

$$P(x) = \frac{1}{Z} e^{-\frac{1}{2}[x^T\Sigma^{-1}x - 2\mu^T\Sigma^{-1}x]}$$

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

$[\mu^T\Sigma^{-1}]^T = n\left(\ \right)$



## Canonical form

log lin model $\div \frac{1}{Z} e^{\sum w_i f_i(x)}$

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

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

vector, $x^T$, matrix, $x$

Param $\theta(\eta, \Lambda)$

a log linear model !!

$e^{\sum_i \eta_i x_i - \frac{1}{2}\sum_{ij} \lambda_{ij} x_i x_j}$ $f(x)$

- Standard form and canonical forms are related:

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

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

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

## Conditioning in canonical form

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

$\eta_1$ = n-vector
$\Lambda_1$ = n×n matrix
$\Lambda_2 = (n+m)\times(n+m)$
$\eta_2$ = n+m -vector

- First multiply: $p(\overset{n}{A}, \overset{m}{B}) = p(A)p(B \mid A)$

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

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

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

$\begin{array}{c} n \\ m \end{array}\begin{pmatrix} \eta_1 \\ 0 \end{pmatrix} + \begin{array}{c} n \\ m \end{array}\begin{pmatrix} \eta_2 \end{pmatrix}$ $\quad \begin{array}{c} n \\ m \end{array}\begin{pmatrix} \Lambda_1 & 0 \\ 0 & 0 \end{pmatrix}$ $\quad \begin{array}{c} n \\ m \end{array}\begin{pmatrix} \Lambda_2 \end{pmatrix}$ $\quad \eta_2 = \begin{array}{c} n \\ m \end{array}\begin{pmatrix} \eta_2 \end{pmatrix}$

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

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

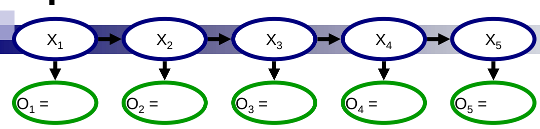
$$\Lambda_{AA|B=y} = \Lambda_{AA}$$

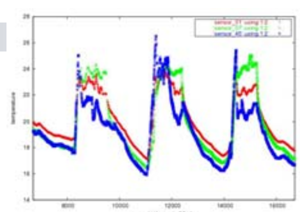
linear function

$P(A,B) : \begin{array}{c} n_A \\ B \end{array}\begin{pmatrix} \boxed{\eta_A} \\ \eta_B \end{pmatrix} \quad \Lambda = \begin{pmatrix} \boxed{\Lambda_{AA}} & \Lambda_{AB} \\ \Lambda_{BA} & \Lambda_{BB} \end{pmatrix}$

17

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

18

# Prediction & roll-up in canonical form

$N(\eta_{X_t}, \Lambda_{X_t X_t})$

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

  sum $\eta$'s & $\Lambda$'s

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

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

19

# Announcements

- Lectures the ~~rest of~~ rest the semester:

  no class this week

  - **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

  ⊗ no classes week of 12/4

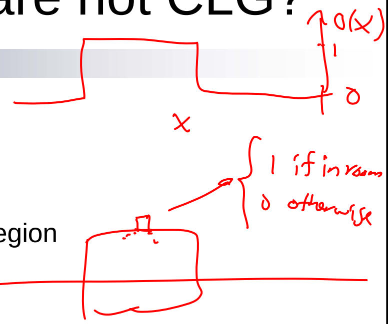
- Deadlines & Presentations:
  - Project Poster Presentations: Dec. $1^{st}$ 3-6pm (NSH Atrium)
    - popular vote for best poster
  - Project write up: Dec. $8^{th}$ by 2pm by email
    - 8 pages – limit will be **strictly enforced**
  - Final: Out Dec. $1^{st}$, Due Dec. $15^{th}$ by 2pm (**strict deadline**)



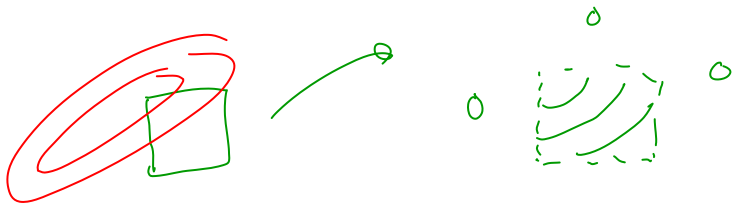
20

# What if observations are not CLG?

- Often observations are not CLG
  - CLG if $O_i = \mathrm{B}\,X_i + \beta_o + \varepsilon$ $\leftarrow N(0, \sigma_2)$
- Consider a motion detector
  - $O_i$ = 1 if person is likely to be in the region



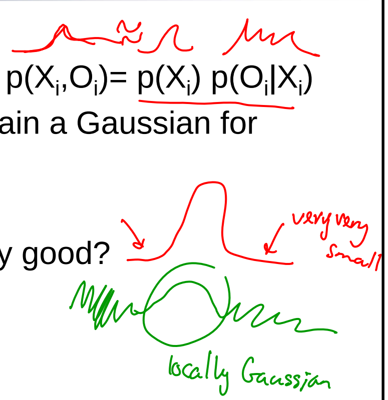

  - Posterior is not Gaussian





# 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



# Linearization as integration

$N(\mu_{x_i}, \Sigma_{x_i})$

$\mu$: $E[X_i]$ $E[O_i]$

- Gaussian approximation of $p(X_i, O_i) = p(X_i)\, p(O_i|X_i)$

$\approx$ Gaussian $\mu, \Sigma$

- Need to compute moments

  $E[X_i] = \mu_{x_i}$

  - $E[O_i] = \int_{o_i} o_i\, p(o_i)\, do_i = \iint_{x_i\, o_i} o_i\, p(x_i)\, p(o_i|x_i)\, do_i\, dx_i$
  - $E[O_i^2]$
  - $E[O_i X_i] = \int_{x_i} \int_{o_i} o_i x_i\, \underbrace{p(o_i|x_i)}_{\text{arbitrary}}\, \underbrace{p(x_i)}_{\text{Gaussian}}\, do_i\, dx_i$

- Note: Integral is product of a Gaussian with an arbitrary function



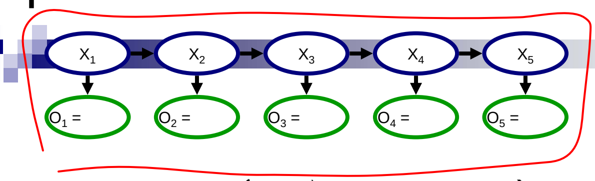
# Linearization as numerical integration

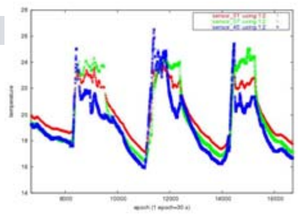
- **Product of a Gaussian with arbitrary function**

- 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**

## Operations in non-linear Kalman filter



non-linear KF

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



## 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

## What if the person chooses different motion models?

- With probability θ, move more or less straight
- With probability 1-θ, do the "moonwalk"

27

## The moonwalk


28

## What if the person chooses different motion models?

- With probability $\theta$, move more or less straight
- With probability $1-\theta$, do the “moonwalk”

29

## Switching Kalman filter

- At each time step, choose one of *k* motion models:
  - You never know which one!
- $p(X_{i+1}|X_i,Z_{i+1})$
  - CLG indexed by $Z_i$
  - $p(X_{i+1}|X_i,Z_{i+1}=j) \sim N(\beta^j_0 + B^j X_i; \Sigma^j_{X_{i+1}|X_i})$

30

# Inference in switching KF – one step

- Suppose
  - $p(X_0)$ is Gaussian
  - $Z_1$ takes one of two values
  - $p(X_1|X_o,Z_1)$ is CLG
- Marginalize $X_0$
- Marginalize $Z_1$
- Obtain mixture of two Gaussians!

31

# Multi-step inference

- Suppose
  - $p(X_i)$ is a mixture of *m* Gaussians
  - $Z_{i+1}$ takes one of two values
  - $p(X_{i+1}|X_i,Z_{i+1})$ is CLG
- Marginalize $X_i$
- Marginalize $Z_i$
- Obtain mixture of 2*m* Gaussians!
  - Number of Gaussians grows exponentially!!!

32

## Visualizing growth in number of Gaussians

33

## Computational complexity of inference in switching Kalman filters

- Switching Kalman Filter with (only) 2 motion models
- Query:
- Problem is NP-hard!!! [Lerner & Parr `01]
  - Why "!!!"?
  - Graphical model is a tree:
    - Inference efficient if all are discrete
    - Inference efficient if all are Gaussian
    - But not with hybrid model (combination of discrete and continuous)

34

# Bounding number of Gaussians

- $P(X_i)$ has $2^m$ Gaussians, but…
- usually, most are bumps have low probability and overlap:

- **Intuitive approximate inference**:
  - Generate *k.m* Gaussians
  - Approximate with *m* Gaussians

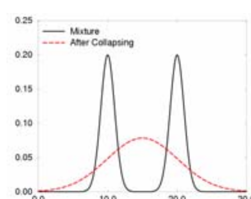
35

# Collapsing Gaussians – Single Gaussian from a mixture

- Given mixture *P* $<w_i;N(\mu_i,\Sigma_i)>$
- Obtain approximation $Q \sim N(\mu,\Sigma)$ as:

$$\mu = \sum_i w_i \mu_i$$

$$\Sigma = \sum_i w_i \Sigma_i + \sum_i w_i (\mu_i - \mu)(\mu_i - \mu)^T$$




- **Theorem**:
  - *P* and *Q* have same first and second moments
  - **KL projection:** *Q* is single Gaussian with lowest KL divergence from *P*

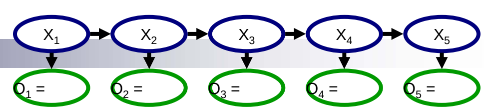
36

# Collapsing mixture of Gaussians into smaller mixture of Gaussians

- Hard problem!
  - Akin to clustering problem…
- Several heuristics exist
  - *c.f.*, K&F book

37

# Operations in non-linear switching Kalman filter




- Compute mixture of Gaussians for $p(X_t \mid O_{1:t} = o_{1:t})$
- Start with $p(X_0)$
- At each time step *t*:
  - For each of the *m* Gaussians in $p(X_i|o_{1:i})$:
    - **Condition** on observation (use **numerical integration**)
    - **Prediction** (Multiply transition model, use **numerical integration**)
      - Obtain *k* Gaussians
    - **Roll-up** (marginalize previous time step)
  - **Project** *k.m* Gaussians into *m'* Gaussians $p(X_i|o_{1:i+1})$

38

## Assumed density filtering

- Examples of very important **assumed density filtering**:
  - Non-linear KF
  - Approximate inference in switching KF
- General picture:
  - Select an **assumed density**
    - e.g., single Gaussian, mixture of *m* Gaussians, …
  - After conditioning, prediction, or roll-up, **distribution no-longer representable with assumed density**
    - e.g., non-linear, mixture of *k.m* Gaussians,…
  - **Project** back into assumed density
    - e.g., numerical integration, collapsing,…

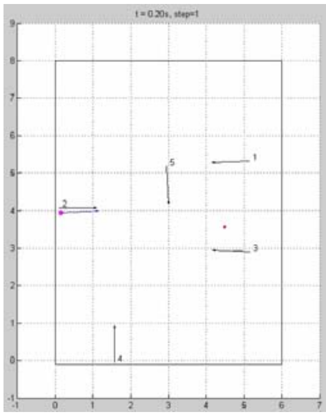


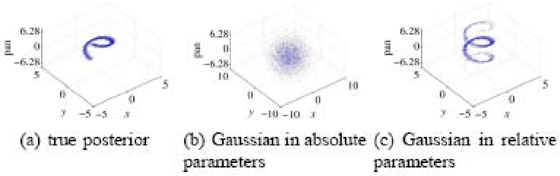
## When non-linear KF is not good enough

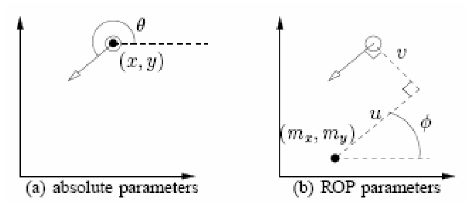
- Sometimes, distribution in non-linear KF is not approximated well as a single Gaussian
  - e.g., a banana-like distribution
- Assumed density filtering:
  - Solution 1: **reparameterize problem** and solve as a **single Gaussian**
  - Solution 2: more typically, **approximate as a mixture of Gaussians**

## Reparameterized KF for SLAT

[Funiak, Guestrin, Paskin, Sukthankar '05]

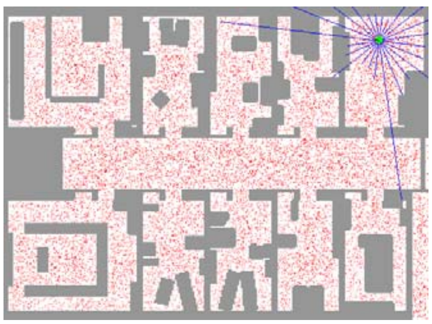







## When a single Gaussian ain't good enough



[Fox et al.]

- Sometimes, smart parameterization is not enough
  - Distribution has multiple hypothesis
- Possible solutions
  - Sampling – particle filtering
  - Mixture of Gaussians
  - …
- Quick overview of one such solution…

## Approximating non-linear KF with mixture of Gaussians

- Robot example:

- $P(X_i)$ is a Gaussian, $P(X_{i+1})$ is a banana
- Approximate $P(X_{i+1})$ as a mixture of *m* Gaussians
  - e.g., using discretization, sampling,…
- Problem:
  - $P(X_{i+1})$ as a mixture of *m* Gaussians
  - $P(X_{i+2})$ is *m* bananas
- One solution:
  - Apply collapsing algorithm to project *m* bananas in *m'* Gaussians

43

## What you need to know

- **Switching Kalman filter**
  - Hybrid model – discrete and continuous vars.
  - Represent belief as mixture of Gaussians
  - Number of mixture components grows exponentially in time
  - Approximate each time step with fewer components
- **Assumed density filtering**
  - Fundamental abstraction of most algorithms for dynamical systems
  - Assume representation for density
  - Every time density not representable, project into representation

44