Readings:
K&F: 4.5, 12.2, 12.3, 12.4, 18.1, 18.2, 18.3, 18.4

Switching Kalman Filter
Dynamic Bayesian Networks

Graphical Models – 10708
Carlos Guestrin
Carnegie Mellon University

November 27th, 2006

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 $1-\theta$, do the “moonwalk”

3

The moonwalk

4

What if the person chooses different motion models?

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

5

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
  - $p(X_{i+1}|X_i,Z_{i+1}=j) \sim N(\beta^j_0 + B^j X_i; \Sigma^j_{X_{i+1}|X_i})$

6

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!

7

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!!!

8

Visualizing growth in number of Gaussians

9

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)

10

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

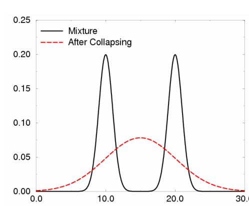
11

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*

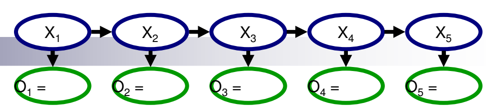
12

Collapsing mixture of Gaussians into smaller mixture of Gaussians

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

13

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 $p(X_i|o_{1:i+1})$

14

Announcements

- Lectures the rest of the semester:
  - Wed. 11/30, regular class time: Causality (Richard Scheines)
  - **Last Class**: 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**)
    - **no late days on final!**

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,…

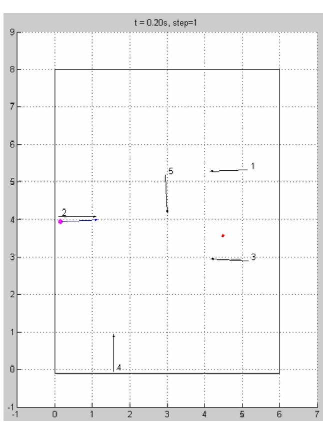
When non-linear KF is not good enough

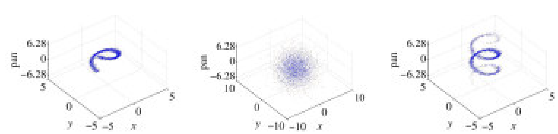
- Sometimes, distribution in non-linear KF is not approximated well as a single Gaussian
  - e.g., a banana-like distribution
- Assumed density filtering:
  - Solution 1: **reparameterize problem** and solve as a **single Gaussian**
  - Solution 2: more typically, **approximate as a mixture of Gaussians**

17

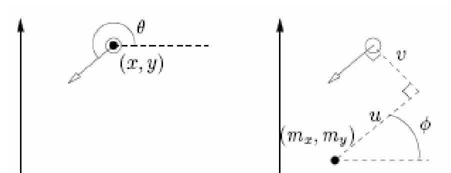
Reparameterized KF for SLAT

[Funiak, Guestrin, Paskin, Sukthankar '05]





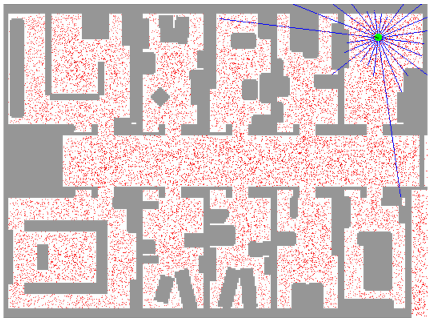
(a) true posterior (b) Gaussian in absolute parameters (c) Gaussian in relative parameters



(a) absolute parameters (b) ROP parameters

18

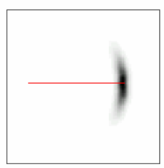
When a single Gaussian ain't good enough



[Fox et al.]

- Sometimes, smart parameterization is not enough
  - Distribution has multiple hypothesis
- Possible solutions
  - Sampling – particle filtering
  - Mixture of Gaussians
  - …
- See book for details…

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

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

21

More than just a switching KF

- Switching KF selects among *k* motion models
- Discrete variable can depend on past
  - Markov model over hidden variable
- What if *k* is really large?
  - Generalize HMMs to large number of variables

22

Dynamic Bayesian network (DBN)

- HMM defined by
  - Transition model $P(X^{(t+1)}|X^{(t)})$
  - Observation model $P(O^{(t)}|X^{(t)})$
  - Starting state distribution $P(X^{(0)})$
- DBN – Use Bayes net to represent each of these compactly
  - Starting state distribution $P(X^{(0)})$ is a BN
  - (silly) e.g, performance in grad. school DBN
    - Vars: **H**appiness, **P**roductivity, Hira**B**lility, **F**ame
    - Observations: Pape**R**, **S**chmooze

23

Transition Model: Two Time-slice Bayes Net (2-TBN)

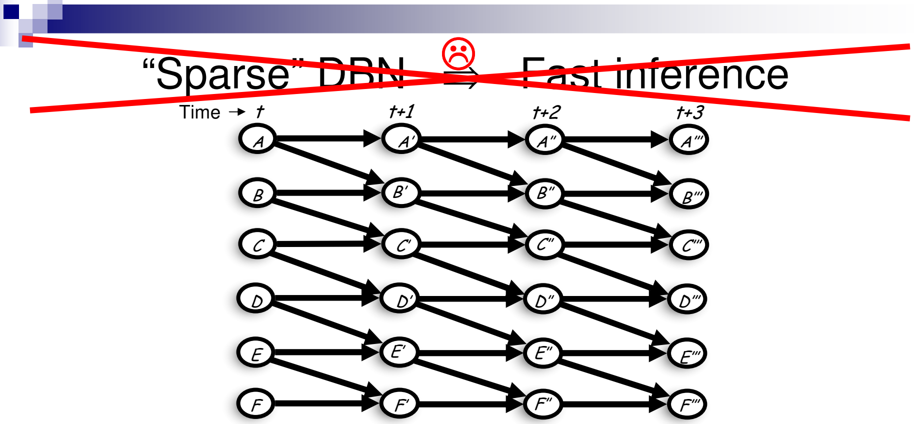
- Process over vars. **X**
- 2-TBN: represents transition and observation models $P(\mathbf{X}^{(t+1)},\mathbf{O}^{(t+1)}|\mathbf{X}^{(t)})$
  - $\mathbf{X}^{(t)}$ are *interface variables* (don't represent distribution over these variables)
  - As with BN, exponential reduction in representation complexity

24

Unrolled DBN

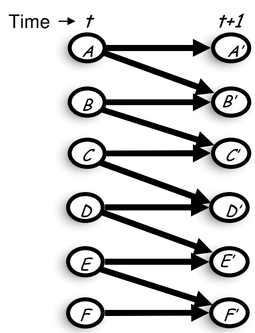
- Start with $P(X^{(0)})$
- For each time step, add vars as defined by 2-TBN

"Sparse" DBN and fast inference



Even after one time step!!

What happens when we marginalize out time t?



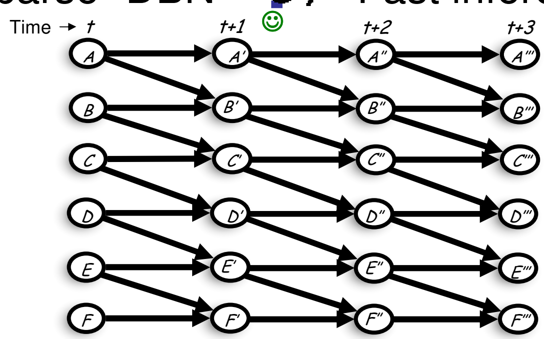
27

"Sparse" DBN and fast inference 2

Structured representation of belief often yields good approximate

Almost!

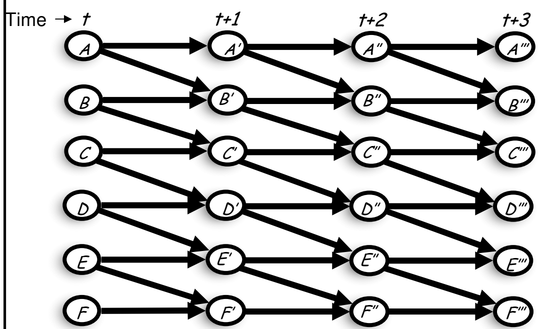
"Sparse" DBN ⇝ ? → Fast inference ☺



28

BK Algorithm for approximate DBN inference [Boyen, Koller '98]

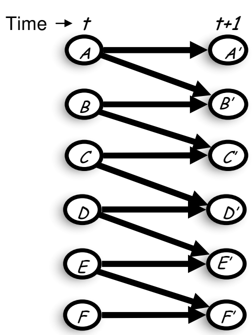
- Assumed density filtering:
  - Choose a factored representation $\hat{P}$ for the belief state
  - Every time step, belief not representable with $\hat{P}$, project into representation



29

A simple example of BK: Fully-Factorized Distribution

- Assumed density:
  - Fully factorized



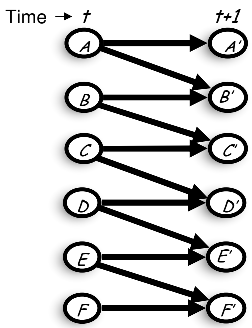
True $P(X^{(t+1)})$:

Assumed Density for $\hat{P}(X^{(t+1)})$:

30

Computing Fully-Factorized Distribution at time t+1

- Assumed density:
  - Fully factorized

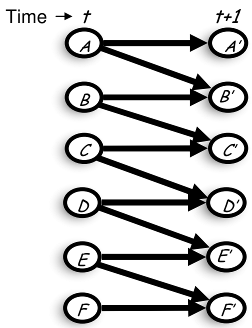


Assumed Density for $\hat{P}(X^{(t+1)})$:

Computing for $\hat{P}(X_i^{(t+1)})$:

General case for BK: Junction Tree Represents Distribution

- Assumed density:
  - Fully factorized

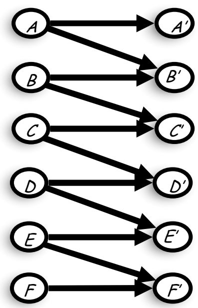


True $P(X^{(t+1)})$:

Assumed Density for $\hat{P}(X^{(t+1)})$:

Computing factored belief state in the next time step

- Introduce observations in current time step
  - Use J-tree to calibrate time *t* beliefs
- Compute *t+1* belief, project into approximate belief state
  - marginalize into desired factors
  - corresponds to KL projection
- Equivalent to computing marginals over factors directly
  - For each factor in *t+1* step belief
    - Use variable elimination



Error accumulation

- Each time step, projection introduces error
- Will error add up?
  - causing unbounded approximation error as $t \to \infty$

Contraction in Markov process

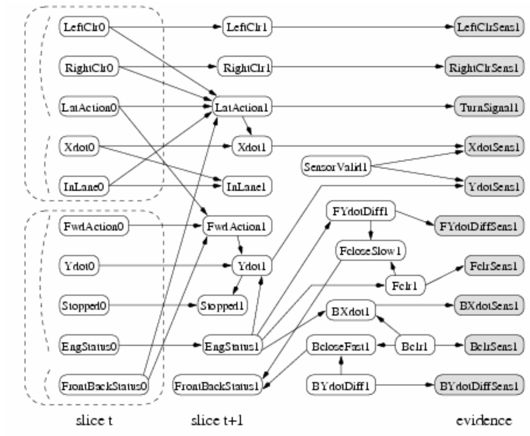
35

BK Theorem

- Error does not grow unboundedly!

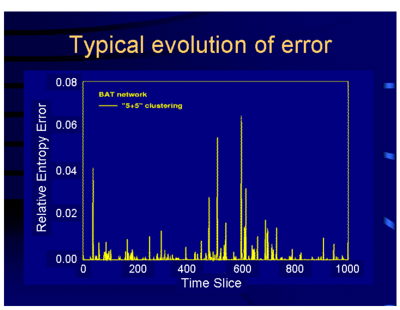
36

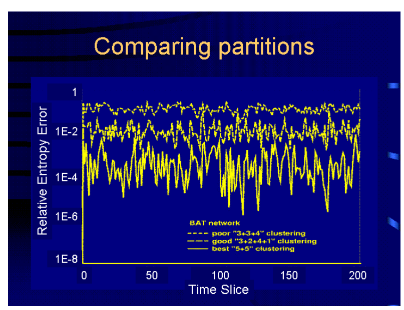
Example – BAT network [Forbes et al.]



37

BK results [Boyen, Koller ’98]

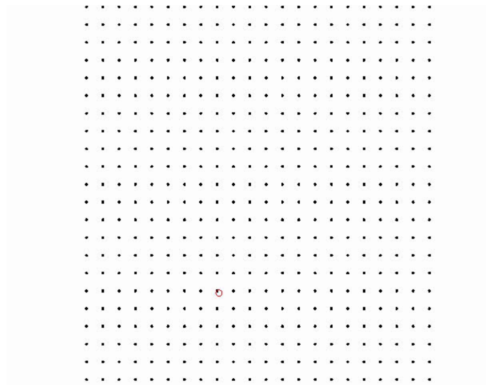




38

Thin Junction Tree Filters [Paskin '03]

- BK assumes fixed approximation clusters
- TJTF adapts clusters over time
  - attempt to minimize projection error



Hybrid DBN (many continuous and discrete variables)

- DBN with large number of discrete and continuous variables
- # of mixture of Gaussian components blows up in one time step!
- Need many smart tricks…
  - e.g., see Lerner Thesis

Figure 10.1: The prototype RWGS system

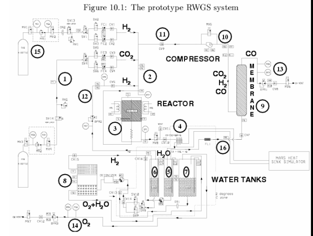


Figure 10.2: The RWGS schematic

Reverse Water Gas Shift System (RWGS) [Lerner et al. '02]

DBN summary

- **DBNs**
  - factored representation of HMMs/Kalman filters
  - sparse representation does not lead to efficient inference
- **Assumed density filtering**
  - BK – factored belief state representation is assumed density
  - Contraction guarantees that error does blow up (but could still be large)
  - Thin junction tree filter adapts assumed density over time
  - Extensions for hybrid DBNs