State Space Models

Probabilistic Graphical Models (10-708)

Lecture 13, Oct 31, 2007

Eric Xing

Reading: J-Chap. 13,14

A road map to more complex
dynamic models

discrete Y

discrete X

Mixture model
e.g., mixture of multinomials

HMM
(for discrete sequential data, e.g., text)

Factorial HMM

continuous Y

continuous X

Mixture model
e.g., mixture of Gaussians

HMM
(for continuous sequential data,
e.g., speech signal)

State space model

Factor analysis

Switching SSM

Eric Xing
State space models (SSM):

- A sequential FA or a continuous state HMM

\[
\begin{align*}
X_t &
\rightarrow X_{t-1} \rightarrow X_{t-2} \rightarrow \cdots \rightarrow X_1 \rightarrow X_0 \\
Y_t &
\rightarrow Y_{t-1} \rightarrow Y_{t-2} \rightarrow \cdots \rightarrow Y_1 \\
\end{align*}
\]

\[
x_t = Ax_{t-1} + Gw_t \\
y_t = Cx_{t-1} + v_t \\
w_t \sim N(0; Q), \quad v_t \sim N(0; R) \\
x_0 \sim N(0; \Sigma_0),
\]

This is a linear dynamic system.

- In general,

\[
x_t = f(x_{t-1}) + Gw_t \\
y_t = g(x_{t-1}) + v_t \\
\]

where \( f \) is an (arbitrary) dynamic model, and \( g \) is an (arbitrary) observation model.

LDS for 2D tracking

- Dynamics: new position = old position + \( \Delta \times \text{velocity} + \text{noise} \)

(constant velocity model, Gaussian noise)

\[
\begin{align*}
\begin{pmatrix}
\dot{x}^1_t \\
\dot{x}^2_t \\
\dot{x}^3_t \\
\dot{x}^4_t \\
\end{pmatrix}
= 
\begin{pmatrix}
1 & 0 & \Delta & 0 \\
0 & 1 & 0 & \Delta \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
\end{pmatrix}
\begin{pmatrix}
x^1_{t-1} \\
x^2_{t-1} \\
x^3_{t-1} \\
x^4_{t-1} \\
\end{pmatrix}
\]

+ noise

- Observation: project out first two components (we observe Cartesian position of object - linear!)

\[
\begin{pmatrix}
y^1_t \\
y^2_t \\
\end{pmatrix}
= 
\begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
\end{pmatrix}
\begin{pmatrix}
x^1_t \\
x^2_t \\
x^3_t \\
x^4_t \\
\end{pmatrix}
\]

+ noise
The inference problem 1

- Filtering $\Rightarrow$ given $y_1, \ldots, y_t$, estimate $x_t$: $P(x_t | y_{1t})$
  - The Kalman filter is a way to perform exact online inference (sequential Bayesian updating) in an LDS.
  - It is the Gaussian analog of the forward algorithm for HMMs:
    \[
    p(X_t = i | y_{1t}) = \alpha'_i \propto p(y_t | x_t = i) \sum_j p(x_t = i | x_{t-1} = j) \alpha'_{j,t-1}
    \]

The inference problem 2

- Smoothing $\Rightarrow$ given $y_1, \ldots, y_T$, estimate $x_t (t < T)$
  - The Rauch-Tung-Strievel smoother is a way to perform exact off-line inference in an LDS. It is the Gaussian analog of the forwards-backwards (alpha-gamma) algorithm:
    \[
    p(X_t = i | y_{1T}) = \gamma'_i \propto \sum_j \alpha'_i p(x'_i | x_i) \gamma'_{i,i}
    \]
2D tracking

Kalman filtering in the brain?
Since all CPDs are linear Gaussian, the system defines a large multivariate Gaussian.

Hence all marginals are Gaussian.

Hence we can represent the belief state $p(X_t|y_1:t)$ as a Gaussian

- mean $\hat{x}_{t|T} \equiv E(X_t|y_1,...,y_T)$
- covariance $P_{t|T} \equiv E(X_TX_T^T|y_1,...,y_T)$

It is common to work with the inverse covariance (precision) matrix $P_{t|T}^{-1}$; this is called information form.

Kalman filtering is a recursive procedure to update the belief state:

- Predict step: compute $p(X_{t+1}|y_1:t)$ from prior belief $p(X_t|y_1:t)$ and dynamical model $p(X_{t+1}|X_t)$ --- time update

- Update step: compute new belief $p(X_{t+1}|y_1:t+1)$ from prediction $p(X_{t+1}|y_1:t)$, observation $y_{t+1}$ and observation model $p(y_{t+1}|X_{t+1})$ --- measurement update
Predict step

- **Dynamical Model:** \( x_{t+1} = \mathbf{A} x_t + \mathbf{G} w_t, \quad w_t \sim \mathcal{N}(0, \mathbf{Q}) \)
  - One step ahead prediction of state:
    \[
    \hat{x}_{t+1} = \mathbf{E}(X_{t+1} | Y_1, \ldots, Y_t) = \mathbf{A} \hat{x}_t
    \]
    \[
    P_{t+1} = \mathbf{E}(X_{t+1} - \hat{x}_{t+1} \mid Y_1, \ldots, Y_t) = \mathbf{E}(\mathbf{A} X_t + \mathbf{G} w_t - \hat{x}_{t+1} \mid Y_1, \ldots, Y_t)
    = \mathbf{A} \mathbf{P}_t \mathbf{A}^T + \mathbf{G} \mathbf{Q} \mathbf{G}^T
    \]

- **Observation model:** \( y_t = \mathbf{C} x_t + v_t, \quad v_t \sim \mathcal{N}(0, \mathbf{R}) \)
  - One step ahead prediction of observation:
    \[
    E(Y_{t+1} | Y_1, \ldots, Y_t) = \mathbf{E}(\mathbf{C} X_{t+1} + \mathbf{V}_{t+1} | Y_1, \ldots, Y_t) = \mathbf{C} \hat{x}_{t+1}
    \]
    \[
    E(Y_{t+1} - \hat{y}_{t+1} \mid Y_1, \ldots, Y_t) = \mathbf{C} P_{t+1} \mathbf{C}^T + \mathbf{R}
    \]
    \[
    E(Y_{t+1} - \hat{y}_{t+1} \mid X_{t+1} - \hat{x}_{t+1}) = \mathbf{C} P_{t+1} \mathbf{C}^T + \mathbf{R}
    \]

Update step

- Summarizing results from previous slide, we have \( p(X_{t+1}, Y_{t+1} | Y_{1:t}) \sim \mathcal{N}(m_{t+1}, V_{t+1}) \), where
  \[
  m_{t+1} = \begin{pmatrix} \hat{x}_{t+1} \\ \mathbf{C} \hat{x}_{t+1} \end{pmatrix}, \quad V_{t+1} = \begin{pmatrix} P_{t+2} & P_{t+1} \mathbf{C}^T \\ \mathbf{C} P_{t+1} \mathbf{C}^T + \mathbf{R} & \mathbf{C} P_{t+1} \mathbf{C}^T + \mathbf{R} \end{pmatrix}
  \]

- Remember the formulas for conditional Gaussian distributions:
  \[
  p(x_1 | x_2, \mu, \Sigma) = \mathcal{N} \left( x_1 \mid \mu_1, \Sigma_{11} \right)
  \]
  \[
  p(x_2 | x_1, m_2^m, V_2^m) = \mathcal{N} \left( x_2 \mid m_2^m, V_2^m \right)
  \]
  \[
  m_2^m = \mu_2, \quad V_2^m = \Sigma_{22}
  \]
  \[
  p(x_1 | x_2, m_1^m, V_1^m) = \mathcal{N} \left( x_1 \mid m_1^m, V_1^m \right)
  \]
  \[
  m_1^m = \mu_1 + \Sigma_{12} \Sigma_{22}^{-1} (x_2 - \mu_2), \quad V_1^m = \Sigma_{11} - \Sigma_{12} \Sigma_{22}^{-1} \Sigma_{21}
  \]
Kalman Filter

- Measurement updates:
  \[ \hat{x}_{t+1|t} = \hat{x}_{t|t} + K_{t+1} (y_{t+1} - C \hat{x}_{t+1|t}) \]
  \[ P_{t+1|t} = P_{t+1|t} - KCP_{t+1|t} \]
  where \( K_{t+1} \) is the Kalman gain matrix
  \[ K_{t+1} = P_{t+1|t} C^T (CP_{t+1|t} C^T + R)^{-1} \]

- Time updates:
  \[ \hat{x}_{t+1} = A \hat{x}_{t|t} \]
  \[ P_{t+1} = AP_{t|t} A^T + GQG^T \]
  \( K_t \) can be pre-computed (since it is independent of the data).

Example of KF in 1D

- Consider noisy observations of a 1D particle doing a random walk:
  \[ x_{t+1} = x_t + w, \quad w \sim \mathcal{N}(0, \sigma_w) \quad z_t = x_t + v, \quad v \sim \mathcal{N}(0, \sigma_v) \]

- KF equations:
  \[ P_{t+1} = AP_{t|t} A^T + GQG^T = \sigma_x + \sigma_x \quad \hat{x}_{t+1} = A \hat{x}_{t|t} = \hat{x}_{t|t} \]
  \[ K_{t+1} = P_{t+1} C^T (CP_{t+1} C^T + R)^{-1} = (\sigma_x + \sigma_x)(\sigma_x + \sigma_x + \sigma_x) \]
  \[ \hat{x}_{t+1} = \hat{x}_{t|t} + K_{t+1} (z_{t+1} - C \hat{x}_{t+1|t}) = \frac{\sigma_x + \sigma_x z_{t+1} + \sigma_x \hat{x}_{t|t}}{\sigma_x + \sigma_x + \sigma_x} \]
  \[ P_{t+1} = P_{t+1} - KCP_{t+1} = \frac{\sigma_x + \sigma_x \sigma_x}{\sigma_x + \sigma_x + \sigma_x} \]
KF intuition

- The KF update of the mean is
  \[ \hat{x}_{t+1|t} = \hat{x}_{t|t} + K_{t+1}(z_{t+1} - C\hat{x}_{t+1|t}) \]
  \[ = \frac{(\sigma_x + \sigma_z)z_t + \sigma_{\hat{x}_{t+1|t}}}{\sigma_x + \sigma_z + \sigma_z} \]

  - the term \((z_{t+1} - C\hat{x}_{t+1|t})\) is called the innovation

- New belief is convex combination of updates from prior and observation, weighted by Kalman Gain matrix:
  \[ K_{t+1} = P_{t+1|t}C(CP_{t+1|t}C^T + R)^{-1} \]

- If the observation is unreliable, \(\sigma_z\) (i.e., \(R\)) is large so \(K_{t+1}\) is small, so we pay more attention to the prediction.

- If the old prior is unreliable (large \(\sigma_x\)) or the process is very unpredictable (large \(\sigma_x\)), we pay more attention to the observation.

KF, RLS and LMS

- The KF update of the mean is
  \[ \hat{x}_{t+1|t} = A\hat{x}_{t|t} + K_{t+1}(y_{t+1} - C\hat{x}_{t+1|t}) \]

- Consider the special case where the hidden state is a constant, \(x_t = \theta\), but the “observation matrix” \(C\) is a time-varying vector, \(C = x_t^T\).
  - Hence the observation model at each time slide, \(y_t = x_t^T\theta + \nu_t\), is a linear regression

- We can estimate recursively using the Kalman filter:
  \[ \hat{\theta}_{t+1} = \hat{\theta}_t + P_{t+1|t}^{-1}(y_{t+1} - x_{t+1}^T\hat{\theta}_t)x_t \]
  This is called the recursive least squares (RLS) algorithm.

- We can approximate \(P_{t+1|t}^{-1} = \eta_{t+1}\) by a scalar constant. This is called the least mean squares (LMS) algorithm.

- We can adapt \(\eta_t\) online using stochastic approximation theory.
Complexity of one KF step

- Let $X_t \in \mathbb{R}^{N_x}$ and $Y_t \in \mathbb{R}^{N_y}$.
- Computing $P_{t+1|t} = A P_{t|t} A + G Q G^T$ takes $O(N_x^2)$ time, assuming dense $P$ and dense $A$.
- Computing $K_{t+1} = P_{t+1|t} C (C P_{t+1|t} C^T + R)^{-1}$ takes $O(N_y^3)$ time.
- So overall time is, in general, max $\{N_x^2, N_y^3\}$

The inference problem 2

- Smoothing $\rightarrow$ given $y_1, \ldots, y_T$, estimate $x_i (t<T)$
  - The Rauch-Tung-Strievel smoother is a way to perform exact off-line inference in an LDS. It is the Gaussian analog of the forwards-backwards (alpha-gamma) algorithm:

$$p(X_t = i | y_{1:T}) = \gamma'_i \propto \sum_j \alpha'_j p(X_{t+1} | X_t) \gamma'_{j+1}$$
RTS smoother derivation

- Smoothing \( \rightarrow \) given \( y_1, \ldots, y_T \), estimate \( P(x_t|y_{1:T}) \) \((t<T)\)
  - Step 1: joint distribution of \( x_t \) and \( x_{t+1} \) conditioned on \( y_{1:t} \)
  - Use \( x_{t+1} = Ax_t + Gw_t; w_t \sim \mathcal{N}(0; Q); \) \( \hat{x}_{t+1|t} = A\hat{x}_{t|t} \)

RTS smoother derivation

- Following the results from previous slide, we need to derive \( p(x_{t+1}, x_t|y_{1:t}) \sim \mathcal{N}(m, V) \), where
  \[
  m = \begin{pmatrix} \hat{x}_{t|t} \\ \hat{x}_{t+1|t} \end{pmatrix}, \quad V = \begin{pmatrix} P_{t|t} & P_{t|t}A^T \\ AP_{t|t} & P_{t+1|t} \end{pmatrix},
  \]
  - all the quantities here are available after a forward KF pass
  - Remember the formulas for conditional Gaussian distributions:
    \[
    p(x_t|x_{t+1}, y_{0:t}) = \mathcal{N}(\mu_t, \Sigma_t = L_t(L_t^T + \Sigma_{t+1|t})^{-1}L_t^T)
    \]
    \[
    E[x_t|x_{t+1}, y_{0:t}] = \hat{x}_{t|t} + L_t(x_{t+1} - \hat{x}_{t+1|t})
    \]
    \[
    \text{Var}[x_t|x_{t+1}, y_{0:t}] = P_{t|t} - L_tP_{t+1|t}L_t^T
    \]
    \[
    L_t = P_{t|t}A^TP_{t+1|t}^T
    \]
## RTS smoother derivation

\[
E[x_t|x_{t+1},y_{0:t}] = \hat{x}_{t|t} + L_t(x_{t+1} - \hat{x}_{t+1|t})
\]

\[
\text{Var}[x_t|x_{t+1},y_{0:t}] = P_{t|t} - L_t P_{t+1|t} L_t^T
\]

- **Step 2:** compute \( \hat{x}_t = E[x_t|y_{0:t}] \) using results above
  - Use \( E[x_t|x_{t+1},y_{0:T}] = E[x_t|x_{t+1},y_{0:t}] \)
  - Use \( E[X|Z] = E[E(X|Y,Z)|Z] \)

## RTS derivation

- Repeat the same process for Variance
  - Refer to Jordan chapter 15

- The RTS smoother results:

\[
\hat{X}_{t|T} = \hat{X}_{t|t} + L_t (\hat{X}_{t+1|T} - \hat{X}_{t+1|t})
\]

\[
P_{t|T} = P_{t|t} + L_t (P_{t+1|T} - P_{t+1|t}) L_t^T
\]
Learning SSMs

- Complete log likelihood
\[
\ell(\theta, D) = \sum \log p(x_t, y_t) = \sum \log p(x_t) + \sum \log p(x_{t+1} | x_t) + \sum \sum \log p(y_{t+1} | x_{t+1})
\]
\[
= f(x_1; \Sigma_0) + f^2(x_t, X_t; x_{t-1}, X_{t-1}; y_t; \forall t; \forall t; A, Q, G) + f^2(x_t, X_t; x_{t-1}, X_{t-1}; \forall t; \forall t; C, R)
\]

- EM
  - E-step: compute \( \langle x_t, x_{t-1}^\prime, \langle x_t, x_{t-1}^\prime, \langle x_t \rangle | x_{t-1} \rangle \rangle \)

  these quantities can be inferred via KF and RTS filters, etc.,

  \( \theta, g, \langle x_t, x_{t-1}^\prime = \text{var}(x_t, x_{t-1}^\prime) + E(x_t)^2 = \Sigma_{0t} + x_{t-1}^2 \)

  - M-step: MLE using

    \[
    \ell^2(\theta, D) = f(x_1; \Sigma_0) + f^2(x_t, X_t; x_{t-1}, X_{t-1}; y_t; \forall t; \forall t; A, Q, G) + f^2(x_t, X_t; x_{t-1}, X_{t-1}; \forall t; \forall t; C, R)
    \]

    c.f., M-step in factor analysis

Nonlinear systems

- In robotics and other problems, the motion model and the observation model are often nonlinear:

  \[
  x_t = f(x_{t-1}) + w_t, \quad y_t = g(x_t) + v_t
  \]

- An optimal closed form solution to the filtering problem is no longer possible.

- The nonlinear functions \( f \) and \( g \) are sometimes represented by neural networks (multi-layer perceptrons or radial basis function networks).

- The parameters of \( f \) and \( g \) may be learned offline using EM, where we do gradient descent (back propagation) in the M step, c.f. learning a MRF/CRF with hidden nodes.

- Or we may learn the parameters online by adding them to the state space: \( x_t = (x_t, \theta) \). This makes the problem even more nonlinear.
Extended Kalman Filter (EKF)

- The basic idea of the EKF is to linearize $f$ and $g$ using a second order Taylor expansion, and then apply the standard KF.
  - i.e., we approximate a stationary nonlinear system with a non-stationary linear system:
    \[
    \begin{align*}
    x_t &= f(\hat{x}_{t-1}) + A_{x_{t-1}} (x_{t-1} - \hat{x}_{t-1}) + w_t \\
    y_t &= g(\hat{x}_{t-1}) + C_{x_{t-1}} (x_{t-1} - \hat{x}_{t-1}) + v_t
    \end{align*}
    \]
    where $\hat{x}_{t-1} = f(\hat{x}_{t-1})$ and $A_x = \frac{\partial f}{\partial x}|_{\hat{x}}$ and $C_x = \frac{\partial g}{\partial x}|_{\hat{x}}$

- The noise covariance ($Q$ and $R$) is not changed, i.e., the additional error due to linearization is not modeled.

Complex Graphical Models

Probabilistic Graphical Models (10-708)

Lecture 14, Nov 7, 2007

Eric Xing

Reading: posted papers
A road map to more complex dynamic models

- **Discrete**
  - Mixture model (e.g., mixture of multinormals)
  - HMM (for discrete sequential data, e.g., text)
  - Factorial HMM

- **Continuous**
  - Mixture model (e.g., mixture of Gaussians)
  - HMM (for continuous sequential data, e.g., speech signal)
  - State space model
  - Factor analysis
  - Switching SSM

NLP and Data Mining

We want:

- Semantic-based search
- Infer topics and categorize documents
- Multimedia inference
- Automatic translation
- Predict how topics evolve
- ...
The Vector Space Model

- Represent each document by a high-dimensional vector in the space of words

Latent Semantic Indexing

- LSA does not define a properly normalized probability distribution of observed and latent entities
- Does not support probabilistic reasoning under uncertainty and data fusion
Latent Semantic Structure

Distribution over words
\[ P(w) = \sum_{\ell} P(w, \ell) \]

Inferring latent structure
\[ P(\ell \mid w) = \frac{P(w \mid \ell)P(\ell)}{P(w)} \]

Prediction
\[ P(w_{n+1} \mid w) = ... \]

Admixture Models

- Objects are bags of elements
- Mixtures are distributions over elements
- Objects have mixing vector \( \theta \)
  - Represents each mixtures' contributions
- Object is generated as follows:
  - Pick a mixture component from \( \theta \)
  - Pick an element from that component
**Topic Models = Admixture Models**

Generating a document

- Draw $\theta$ from the prior

For each word $n$
- Draw $z_n$ from $\text{multinomial}(\theta)$
- Draw $w_n \mid z_n, \{\beta_k\}$ from $\text{multinomial}(\beta_k)$

Which prior to use?

**Choice of Prior**

- **Dirichlet (LDA) (Blei et al. 2003)**
  - Conjugate prior means efficient inference
  - Can only capture variations in each topic’s intensity independently

- **Logistic Normal (CTM=LoNTAM) (Blei & Lafferty 2005, Ahmed & Xing 2006)**
  - Capture the intuition that some topics are highly correlated and can rise up in intensity together
  - Not a conjugate prior implies hard inference
Logistic Normal Vs. Dirichlet

![Graphs showing comparison between Logistic Normal and Dirichlet distributions](image)

Logistic Normal

Dirichlet
**Mixed Membership Model (M³)**

- Mixture versus admixture

A Bayesian mixture model  
A Bayesian admixture model: Mixed membership model

**Latent Dirichlet Allocation: M³ in text mining**

- A document is a bag of words each generated from a randomly selected topic

---

The William Randolph Hearst Foundation will give $5 million to Lincoln Center, Metropolitan Opera Co., New York Philharmonic and Juilliard School. "Our board felt that we had a real opportunity to make a mark on the future of the performing arts with these grants," an important source of support in health, medical research, education and the visual arts," Hearst Foundation President Randolph A. Hearst said Monday in announcing the grants. Lincoln Center's plans will be $500,000 for its new building, which will house opera, theater and music programs. New public buildings. The Metropolitan Opera Co., New York Philharmonic and Juilliard School will receive a total of $500,000 each. The Juilliard School, which aims to perform arts education, will get $500,000. The Hearst Foundation, a leading supporter of the Lincoln Center, will make its usual annual $10,000 donation, too.
Population admixture: $M^3$ in genetics

- The genetic materials of each modern individual are inherited from multiple ancestral populations, each DNA locus may have a different generic origin …

- Ancestral labels may have (e.g., Markovian) dependencies

Inference in Mixed Membership Models

- Mixture versus admixture

\[ p(D) = \sum_{\{z_{nm}\}} \prod_{n} \prod_{m} p(X_{n,m} | \phi_{z_{nm}}, \pi_{z_{nm}}) p(\pi_{z_{nm}} | \alpha) p(\phi | \mathcal{G}) d\pi_{z_{nm}} \cdots d\pi_{z_{nm}} d\phi \]

- Inference is very hard in $M^3$; all hidden variables are coupled and not factorizable!

\[ p(\pi_{z_{nm}} | D) = \sum_{\{z_{nm}\}} \prod_{n} \prod_{m} p(X_{n,m} | \phi_{z_{nm}}, \pi_{z_{nm}}) p(\pi_{z_{nm}} | \alpha) p(\phi | \mathcal{G}) d\pi_{z_{nm}} \cdots d\pi_{z_{nm}} d\phi \]
Approaches to inference

- Exact inference algorithms
  - The elimination algorithm
  - The junction tree algorithms

- Approximate inference techniques
  - Monte Carlo algorithms:
    - Stochastic simulation / sampling methods
    - Markov chain Monte Carlo methods
  - Variational algorithms:
    - Belief propagation
    - Variational inference