Time series, HMMs, Kalman Filters

Machine Learning – 10701/15781
Carlos Guestrin
Carnegie Mellon University

March 28th, 2005
Adventures of our BN hero

- Compact representation for probability distributions
- Fast inference
- Fast learning

But... Who are the most popular kids? 1. Naïve Bayes

2 and 3. Hidden Markov models (HMMs) Kalman Filters
Handwriting recognition

Character recognition, e.g., kernel SVMs
Example of a hidden Markov model (HMM)
Understanding the HMM Semantics

\[ X_1 = \{a,\ldots,z\} \rightarrow X_2 = \{a,\ldots,z\} \rightarrow X_3 = \{a,\ldots,z\} \rightarrow X_4 = \{a,\ldots,z\} \rightarrow X_5 = \{a,\ldots,z\} \]

\[ O_1 = \text{ dib } \rightarrow O_2 = \text{ f } \rightarrow O_3 = \text{ a } \rightarrow O_4 = \text{ e } \rightarrow O_5 = \text{ e } \]
HMMs semantics: Details

Just 3 distributions:

\[ P(X_1) \]

\[ P(X_i \mid X_{i-1}) \]

\[ P(O_i \mid X_i) \]
HMMs semantics: Joint distribution

\[ P(X_1, \ldots, X_n \mid o_1, \ldots, o_n) = P(X_1^n \mid o_1^n) \]
\[ \propto P(X_1)P(o_1 \mid X_1) \prod_{i=2}^{n} P(X_i \mid X_{i-1})P(o_i \mid X_i) \]
Learning HMMs from fully observable data is easy.

Learn 3 distributions:

\[ P(X_1) \]

\[ P(O_i \mid X_i) \]

\[ P(X_i \mid X_{i-1}) \]
Possible inference tasks in an HMM

Marginal probability of a hidden variable:

Viterbi decoding – most likely trajectory for hidden vars:
Using variable elimination to compute $P(X_i | o_{1:n})$

Variable elimination order?

Example:
What if I want to compute $P(X_i|o_{1:n})$ for each $i$?

Variable elimination for each $i$?

Variable elimination for each $i$, what’s the complexity?
Reusing computation

Compute:

\[ P( X_i \mid o_1..n ) \]
The forwards-backwards algorithm

\[ P(X_i \mid o_1 \ldots n) \]

- Initialization: \( \alpha_1(X_1) = P(X_1)P(o_1 \mid X_1) \)

- For \( i = 2 \) to \( n \)
  - Generate a forwards factor by eliminating \( X_{i-1} \)
    \[
    \alpha_i(X_i) = \sum_{x_{i-1}} P(o_i \mid X_i)P(X_i \mid X_{i-1} = x_{i-1})\alpha_{i-1}(x_{i-1})
    \]

- Initialization: \( \beta_n(X_n) = 1 \)

- For \( i = n-1 \) to \( 1 \)
  - Generate a backwards factor by eliminating \( X_{i+1} \)
    \[
    \beta_i(X_i) = \sum_{x_{i+1}} P(o_{i+1} \mid x_{i+1})P(x_{i+1} \mid X_i)\beta_{i+1}(x_{i+1})
    \]

- \( \forall \ i \), probability is: \( P(X_i \mid o_1 \ldots n) = \alpha_i(X_i)\beta_i(X_i) \)
Most likely explanation

Compute:

Variable elimination order?

Example:
The Viterbi algorithm

- Initialization: $\alpha_1(X_1) = P(X_1)P(o_1 \mid X_1)$
- For $i = 2$ to $n$
  - Generate a forwards factor by eliminating $X_{i-1}$
  
  $$\alpha_i(X_i) = \max_{x_{i-1}} P(o_i \mid X_i)P(X_i \mid X_{i-1} = x_{i-1})\alpha_{i-1}(x_{i-1})$$

- Computing best explanation: $x_n^* = \operatorname{argmax}_x \alpha_n(x_n)$
- For $i = n-1$ to $1$
  - Use $\operatorname{argmax}$ to get explanation:
  
  $$x_i^* = \operatorname{argmax}_x P(x_{i+1}^* \mid x_i)\alpha_i(x_i)$$
What about continuous variables?

- In general, very hard!
  - Must represent complex distributions

- A special case is very doable
  - When everything is Gaussian
  - Called a Kalman filter
  - One of the most used algorithms in the history of probabilities!
Time series data example: Temperatures from sensor network
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
    \[ p(X_t \mid o_{1:t}) \propto p(X_t \mid o_{1:t-1})p(o_t \mid X_t) \]
  - **Roll-up** (marginalize previous time step)
    \[ p(X_{t+1} \mid o_{1:t}) = \int_{X_t} P(X_{t+1} \mid x_t)p(x_t \mid o_{1:t})dx_t \]
Detour: Understanding Multivariate Gaussians

Observe attributes
Example: Observe $X_1 = 18$

$P(X_2 | X_1 = 18)$
Characterizing a multivariate Gaussian

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

Mean vector:

Covariance matrix:
Conditional Gaussians

- Conditional probabilities
  - $P(Y|X)$

[3D Graph showing conditional probabilities]
Kalman filter with Gaussians

$P(X_1)$
$P(O_i \mid X_i)$
$P(X_i \mid X_{i-1})$

- Equivalent to a linear system
Detour 2: Canonical form

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

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

- Standard form and canonical forms are related:
  \[ \mu = \Lambda^{-1} \eta \]
  \[ \Sigma = \Lambda^{-1} \]

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

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

- First multiply: \( p(A, B) = p(A)p(B \mid A) \)
  \[ p(A) : \eta_1, \Lambda_1 \]
  \[ p(B \mid A) : \eta_2, \Lambda_2 \]
  \[ p(A, B) : \eta_3 = \eta_1 + \eta_2, \Lambda_3 = \Lambda_1 + \Lambda_2 \]

- Then, condition on value \( B = y \) \( p(A \mid B = y) \)
  \[ \eta_{A\mid B=y} = \eta_A - \Lambda_{AB \cdot y} \]
  \[ \Lambda_{A\cdot A\mid B=y} = \Lambda_{AA} \]
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
    $$p(X_t \mid o_{1:t}) \propto p(X_t \mid o_{1:t-1})p(o_t \mid X_t)$$
  - **Roll-up** (marginalize previous time step)
    $$p(X_{t+1} \mid o_{1:t}) = \int_{X_t} P(X_{t+1} \mid x_t)p(x_t \mid o_{1:t})dx_t$$
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 \( X_t \):  \[ p(A) = \int_B P(A, b)\,db \]

\[
\eta_A^m = \eta_A - \Lambda_{AB} \Lambda_{BB}^{-1} \eta_B \\
\Lambda_{AA}^m = \Lambda_{AA} - \Lambda_{AB} \Lambda_{BB}^{-1} \Lambda_{BA}
\]
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
    \[
    p(X_t \mid o_{1:t}) \propto p(X_t \mid o_{1:t-1})p(o_t \mid X_t)
    \]
  - **Roll-up** (marginalize previous time step)
    \[
    p(X_{t+1} \mid o_{1:t}) = \int_{X_t} P(X_{t+1} \mid x_t)p(x_t \mid o_{1:t})dx_t
    \]
Learning a Kalman filter

- Must learn: \( P(X_1) \)

\[
P(O_i \mid X_i) = \frac{P(O_i, X_i)}{P(O_i)}
\]

\[
P(X_i \mid X_{i-1}) = \frac{P(X_i, X_{i-1})}{P(X_{i-1})}
\]

- Learn joint, and use division rule:

\[
p(A) : \ \eta_1, \ \Lambda_1
\]

\[
p(A, B) : \ \eta_2, \ \Lambda_2
\]

\[
p(B \mid A) = \frac{p(A, B)}{p(A)} : \ \eta_3 = \eta_2 - \eta_1, \ \Lambda_3 = \Lambda_2 - \Lambda_1
\]
Maximum likelihood learning of a multivariate Gaussian

\[ \mu = \Lambda^{-1} \eta \]
\[ \Sigma = \Lambda^{-1} \]

- Data: \( \langle x_{1}^{(j)}, \ldots, x_{n}^{(j)} \rangle \)
- Means are just empirical means:
  \[ \hat{\mu}_{i} = \frac{\sum_{j=1}^{m} x_{i}^{(j)}}{m} \]
- Empirical covariances:
  \[ \hat{\Sigma}_{ik} = \frac{\sum_{j=1}^{m} (x_{i}^{(j)} - \hat{\mu}_{i}) (x_{k}^{(j)} - \hat{\mu}_{k})}{m} \]
What you need to know

- Hidden Markov models (HMMs)
  - Very useful, very powerful!
  - Speech, OCR,…
  - Parameter sharing, only learn 3 distributions
  - Trick reduces inference from $O(n^2)$ to $O(n)$
  - Special case of BN

- Kalman filter
  - Continuous vars version of HMMs
  - Assumes Gaussian distributions
  - Equivalent to linear system
  - Simple matrix operations for computations