Artificial Intelligence

Dynamic Models

Marc Toussaint
University of Stuttgart
Winter 2016/17

Motivation:

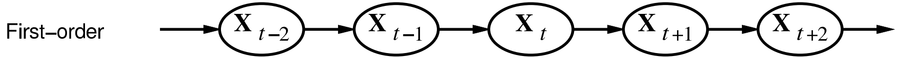
This lecture covors a special case of graphical models for dynamic processes, where the graph is roughly a chain. Such models are called Markov processes, or hidden Markov model when the random variable of the dynamic process is not observable. These models are a cornerstone of time series analysis, as well as for temporal models for language, for instance. A special case of inference in the continuous case is the Kalman filter, which can be use to tracking objects or the state of controlled system.

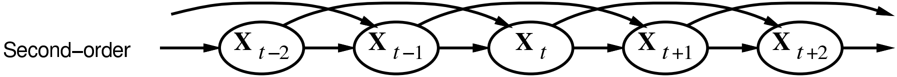
Markov processes (Markov chains)

Markov assumption: X_t depends on *bounded* subset of $X_{0:t-1}$

First-order Markov process: $P(X_t \mid X_{0:t-1}) = P(X_t \mid X_{t-1})$

Second-order Markov process: $P(X_t \mid X_{0:t-1}) = P(X_t \mid X_{t-2}, X_{t-1})$





Sensor Markov assumption: $P(Y_t \mid X_{0:t}, Y_{0:t-1}) = P(Y_t \mid X_t)$

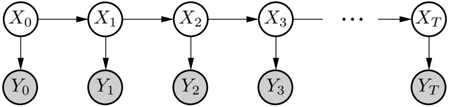
Stationary process: transition model $P(X_t \mid X_{t-1})$ and sensor model $P(Y_t \mid X_t)$ fixed for all t

Hidden Markov Models

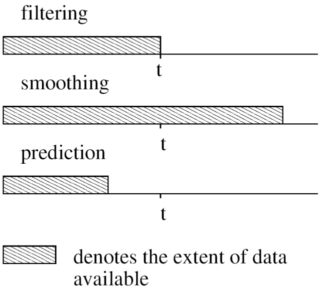
- We assume we have
  - observed (discrete or continuous) variables Y_t in each time slice
  - a discrete latent variable X_t in each time slice
  - some observation model $P(Y_t \mid X_t; \theta)$
  - some transition model $P(X_t \mid X_{t\text{-}1}; \theta)$

- A **Hidden Markov Model (HMM)** is defined as the joint distribution

$$P(X_{0:T}, Y_{0:T}) = P(X_0) \cdot \prod_{t=1}^{T} P(X_t|X_{t\text{-}1}) \cdot \prod_{t=0}^{T} P(Y_t|X_t) \ .$$



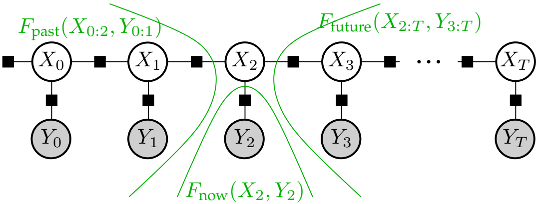
Different inference problems in Markov Models



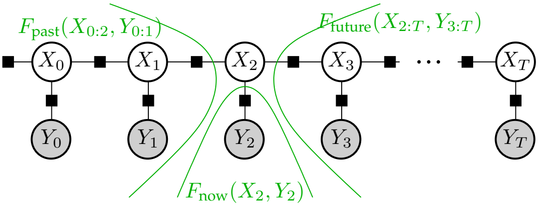
- $P(x_t \,|\, y_{0:T})$ marginal posterior
- $P(x_t \,|\, y_{0:t})$ **filtering**
- $P(x_t \,|\, y_{0:a})$, $t > a$ prediction
- $P(x_t \,|\, y_{0:b})$, $t < b$ **smoothing**
- $P(y_{0:T})$ likelihood calculation

- **Viterbi** alignment: Find sequence $x^*_{0:T}$ that maximizes $P(x_{0:T} \,|\, y_{0:T})$
  (This is done using max-product, instead of sum-product message passing.)

Inference in an HMM – a tree!



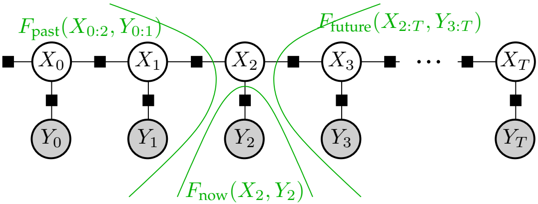
Inference in an HMM – a tree!



- The marginal posterior $P(X_t \mid Y_{1:T})$ is the product of three messages

$$P(X_t \mid Y_{1:T}) \propto P(X_t, Y_{1:T}) = \underbrace{\mu_{\text{past}}}_{\alpha}(X_t)\ \underbrace{\mu_{\text{now}}}_{\varrho}(X_t)\ \underbrace{\mu_{\text{future}}}_{\beta}(X_t)$$

Inference in an HMM – a tree!



- The marginal posterior $P(X_t \,|\, Y_{1:T})$ is the product of three messages

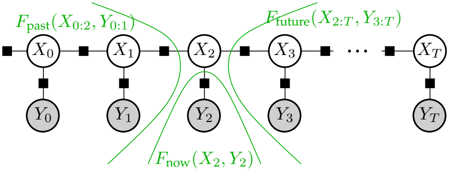
$$P(X_t \,|\, Y_{1:T}) \propto P(X_t, Y_{1:T}) = \underbrace{\mu_{\text{past}}}_{\alpha}(X_t)\ \underbrace{\mu_{\text{now}}}_{\varrho}(X_t)\ \underbrace{\mu_{\text{future}}}_{\beta}(X_t)$$

- For all $a < t$ and $b > t$
  – X_a conditionally independent from X_b given X_t
  – Y_a conditionally independent from Y_b given X_t

“The future is independent of the past given the present”

Markov property

Inference in HMMs



Applying the general message passing equations:

forward msg. $$\mu_{X_{t\text{-}1}\to X_t}(x_t) =: \alpha_t(x_t) = \sum_{x_{t\text{-}1}} P(x_t|x_{t\text{-}1})\ \alpha_{t\text{-}1}(x_{t\text{-}1})\ \varrho_{t\text{-}1}(x_{t\text{-}1})$$

$$\alpha_0(x_0) = P(x_0)$$

backward msg. $$\mu_{X_{t+1}\to X_t}(x_t) =: \beta_t(x_t) = \sum_{x_{t+1}} P(x_{t+1}|x_t)\ \beta_{t+1}(x_{t+1})\ \varrho_{t+1}(x_{t+1})$$

$$\beta_T(x_0) = 1$$

observation msg. $$\mu_{Y_t\to X_t}(x_t) =: \varrho_t(x_t) = P(y_t \mid x_t)$$

posterior marginal $$q(x_t) \propto \alpha_t(x_t)\ \varrho_t(x_t)\ \beta_t(x_t)$$

posterior marginal $$q(x_t, x_{t+1}) \propto \alpha_t(x_t)\ \varrho_t(x_t)\ P(x_{t+1}|x_t)\ \varrho_{t+1}(x_{t+1})\ \beta_{t+1}(x_{t+1})$$

Inference in HMMs – implementation notes

- The message passing equations can be implemented by reinterpreting them as matrix equations: Let $\boldsymbol{\alpha}_t, \boldsymbol{\beta}_t, \boldsymbol{\varrho}_t$ be the vectors corresponding to the probability tables $\alpha_t(x_t), \beta_t(x_t), \varrho_t(x_t)$; and let $\boldsymbol{P}$ be the matrix with enties $P(x_t \,|\, x_{t\text{-}1})$. Then

  ---

  1: $\boldsymbol{\alpha}_0 = \boldsymbol{\pi},\ \boldsymbol{\beta}_T = 1$
  2: $\texttt{for}_{t=1:T\text{-}1}:\ \boldsymbol{\alpha}_t = \boldsymbol{P}\,(\boldsymbol{\alpha}_{t\text{-}1} \circ \boldsymbol{\varrho}_{t\text{-}1})$
  3: $\texttt{for}_{t=T\text{-}1:0}:\ \boldsymbol{\beta}_t = \boldsymbol{P}^\top\,(\boldsymbol{\beta}_{t+1} \circ \boldsymbol{\varrho}_{t+1})$
  4: $\texttt{for}_{t=0:T}:\ \boldsymbol{q}_t = \boldsymbol{\alpha}_t \circ \boldsymbol{\varrho}_t \circ \boldsymbol{\beta}_t$
  5: $\texttt{for}_{t=0:T\text{-}1}:\ \boldsymbol{Q}_t = \boldsymbol{P} \circ [(\boldsymbol{\beta}_{t+1} \circ \boldsymbol{\varrho}_{t+1})\,(\boldsymbol{\alpha}_t \circ \boldsymbol{\varrho}_t)^\top]$

  ---

  where $\circ$ is the *element-wise product!* Here, $\boldsymbol{q}_t$ is the vector with entries $q(x_t)$, and $\boldsymbol{Q}_t$ the matrix with entries $q(x_{t+1}, x_t)$. Note that the equation for $\boldsymbol{Q}_t$ describes $Q_t(x', x) = P(x'|x)[(\beta_{t+1}(x')\varrho_{t+1}(x'))(\alpha_t(x)\varrho_t(x))]$.

Inference in HMMs: classical derivation*

Given our knowledge of Belief propagation, inference in HMMs is simple. For reference, here is a more classical derivation:

$$
\begin{aligned}
P(x_t \,|\, y_{0:T}) &= \frac{P(y_{0:T} \,|\, x_t)\; P(x_t)}{P(y_{0:T})} \\
&= \frac{P(y_{0:t} \,|\, x_t)\; P(y_{t+1:T} \,|\, x_t)\; P(x_t)}{P(y_{0:T})} \\
&= \frac{P(y_{0:t}, x_t)\; P(y_{t+1:T} \,|\, x_t)}{P(y_{0:T})} \\
&= \frac{\alpha_t(x_t)\; \beta_t(x_t)}{P(y_{0:T})}
\end{aligned}
$$

$$
\begin{aligned}
\alpha_t(x_t) &:= P(y_{0:t}, x_t) = P(y_t|x_t)\; P(y_{0:t\text{-}1}, x_t) \\
&= P(y_t|x_t)\; \sum_{x_{t\text{-}1}} P(x_t \,|\, x_{t\text{-}1})\; \alpha_{t\text{-}1}(x_{t\text{-}1}) \\
\beta_t(x_t) &:= P(y_{t+1:T} \,|\, x_t) = \sum_{x_{t+1}} P(y_{t+1:T} \,|\, x_{t+1})\; P(x_{t+1} \,|\, x_t) \\
&= \sum_{x_{t+1}} \Big[\beta_{t+1}(x_{t+1})\; P(y_{t+1}|x_{t+1})\Big]\; P(x_{t+1} \,|\, x_t)
\end{aligned}
$$

Note: α_t here is the same as $\alpha_t \circ \varrho_t$ on all other slides!

HMM remarks

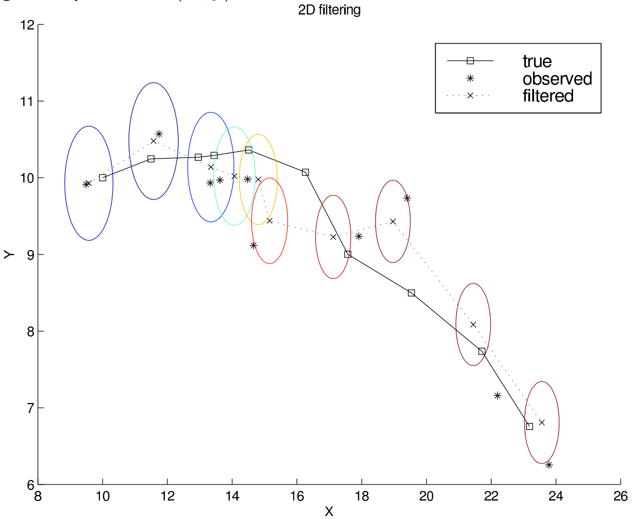
- The computation of forward and backward messages along the Markov chain is also called **forward-backward algorithm**
- Sometimes, computing forward and backward messages (in disrete or continuous context) is also called **Bayesian filtering/smoothing**
- The EM algorithm to learn the HMM parameters is also called **Baum-Welch algorithm**
- If the latent variable x_t is **continuous** $x_t \in \mathbb{R}^d$ instead of discrete, then such a Markov model is also called **state space model**.
- If the continuous transitions and observations are linear Gaussian

$$P(x_{t+1}|x_t) = \mathcal{N}(x_{t+1} \,|\, Ax_t + a, Q) \;, \quad P(y_t|x_t) = \mathcal{N}(y_t \,|\, Cx_t + c, W)$$

  then the forward and backward messages α_t and β_t are also Gaussian.
  $\to$ forward filtering is also called **Kalman filtering**
  $\to$ smoothing is also called **Kalman smoothing**

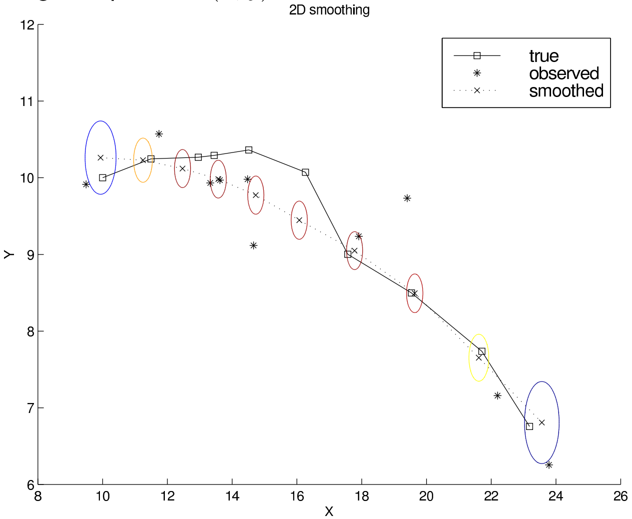
Kalman Filter example

- filtering of a position $(x, y) \in \mathbb{R}^2$:



Kalman Filter example

- smoothing of a position $(x, y) \in \mathbb{R}^2$:



HMM example: Learning Bach

- A machine "listens" (reads notes of) Bach pieces over and over again $\rightarrow$ It's supposed to learn how to write Bach pieces itself (or at least harmonize them).

- *Harmonizing Chorales in the Style of J S Bach* Moray Allan & Chris Williams (NIPS 2004)

- use an HMM
  – observed sequence $Y_{0:T}$ Soprano melody
  – latent sequence $X_{0:T}$ chord & and harmony:

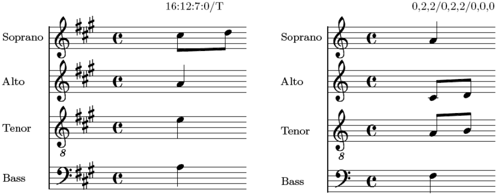


Figure 1: Hidden state representations (a) for harmonisation, (b) for ornamentation.

HMM example: Learning Bach

- results: `http://www.anc.inf.ed.ac.uk/demos/hmmbach/`

Figure 2: Most likely harmonisation under our model of chorale K4, BWV 48

- See also work by Gerhard Widmer
  `http://www.cp.jku.at/people/widmer/`

Dynamic Bayesian Networks

– Arbitrary BNs in each time slide
– Special case: MDPs, speech, etc