

# Artificial Intelligence

Dynamic Models

Marc Toussaint  
University of Stuttgart  
Winter 2018/19

## Motivation:

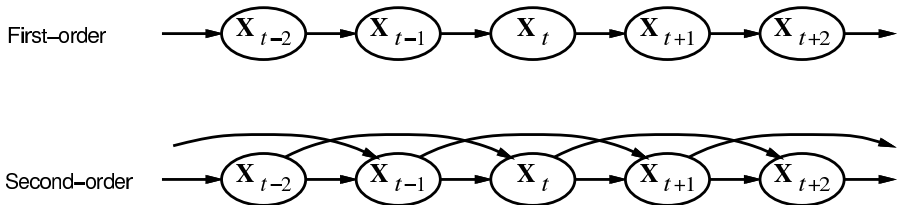
This lecture covers a special case of graphical models for dynamic processes, where the graph is roughly a chain. Such models are called Markov processes, or hidden Markov model when the random variable of the dynamic process is not observable. These models are a cornerstone of time series analysis, 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 used 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:  $P(X_t | X_{0:t-1}) = P(X_t | X_{t-1})$

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



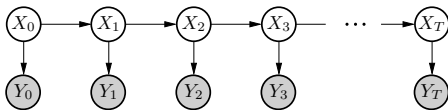
Sensor Markov assumption:  $P(Y_t | X_{0:t}, Y_{0:t-1}) = P(Y_t | X_t)$

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

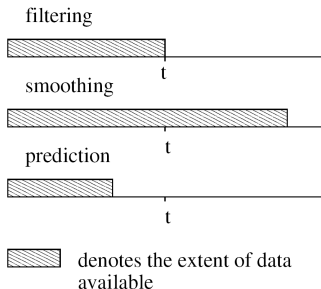
# Hidden Markov Models

- 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  $P(Y_t | X_t; \theta)$
  - some transition model  $P(X_t | X_{t-1}; \theta)$
- A **Hidden Markov Model (HMM)** is defined as the joint distribution

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



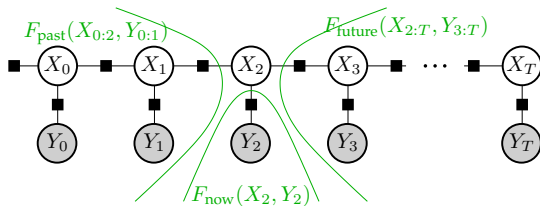
## Different inference problems in Markov Models



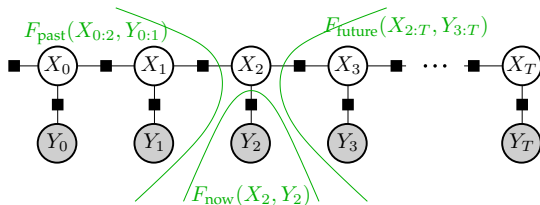
- $P(x_t | y_{0:T})$  marginal posterior
- $P(x_t | y_{0:t})$  **filtering**
- $P(x_t | y_{0:a}), t > a$  prediction
- $P(x_t | y_{0:b}), t < b$  **smoothing**
- $P(y_{0:T})$  likelihood calculation

- **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!



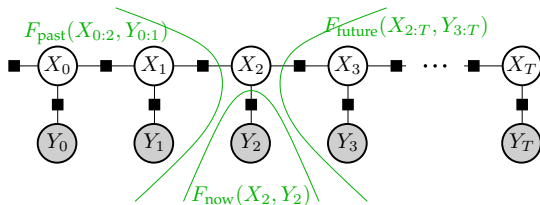
# Inference in an HMM – a tree!



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

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

# Inference in an HMM – a tree!



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

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

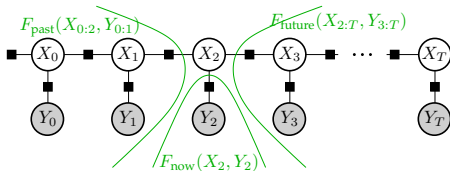
- For all  $a < t$  and  $b > t$ 
  - $X_a$  conditionally independent from  $X_b$  given  $X_t$
  - $Y_a$  conditionally independent from  $Y_b$  given  $X_t$

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

**Markov property**



# Inference in HMMs



Applying the general message passing equations:

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

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

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

$$\beta_T(x_T) = 1$$

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

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  $\alpha_t, \beta_t, \varrho_t$  be the vectors corresponding to the probability tables  $\alpha_t(x_t), \beta_t(x_t), \varrho_t(x_t)$ ; and let  $P$  be the matrix with entries  $P(x_t | x_{t-1})$ . Then

---

1:  $\alpha_0 = \pi, \beta_T = 1$

2:  $\text{for}_{t=1:T-1} : \alpha_t = P (\alpha_{t-1} \circ \varrho_{t-1})$

3:  $\text{for}_{t=T-1:0} : \beta_t = P^\top (\beta_{t+1} \circ \varrho_{t+1})$

4:  $\text{for}_{t=0:T} : \mathbf{q}_t = \alpha_t \circ \varrho_t \circ \beta_t$

5:  $\text{for}_{t=0:T-1} : \mathbf{Q}_t = P \circ [(\beta_{t+1} \circ \varrho_{t+1}) (\alpha_t \circ \varrho_t)^\top]$

---

where  $\circ$  is the *element-wise product*! Here,  $\mathbf{q}_t$  is the vector with entries  $q(x_t)$ , and  $\mathbf{Q}_t$  the matrix with entries  $q(x_{t+1}, x_t)$ . Note that the equation for  $\mathbf{Q}_t$  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:

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

$$\begin{aligned} \alpha_t(x_t) &:= P(y_{0:t}, x_t) = P(y_t | x_t) P(y_{0:t-1}, x_t) \\ &= P(y_t | x_t) \sum_{x_{t-1}} P(x_t | x_{t-1}) \alpha_{t-1}(x_{t-1}) \end{aligned}$$

$$\begin{aligned} \beta_t(x_t) &:= P(y_{t+1:T} | x_t) = \sum_{x_{t+1}} P(y_{t+1:T} | x_{t+1}) P(x_{t+1} | x_t) \\ &= \sum_{x_{t+1}} \left[ \beta_{t+1}(x_{t+1}) P(y_{t+1} | x_{t+1}) \right] P(x_{t+1} | x_t) \end{aligned}$$

Note:  $\alpha_t$  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 discrete 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

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

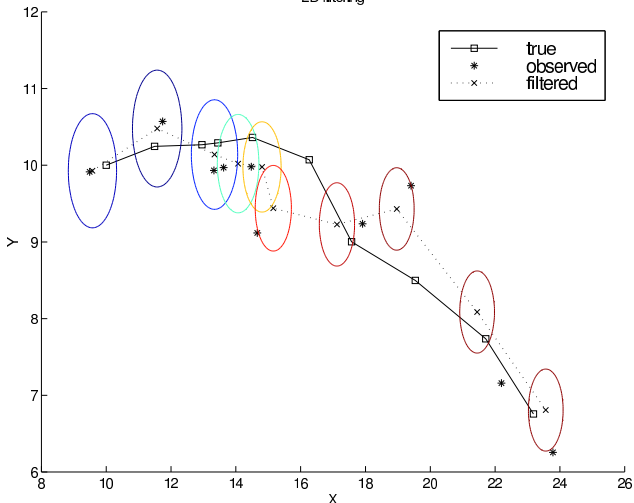
then the forward and backward messages  $\alpha_t$  and  $\beta_t$  are also Gaussian.

→ forward filtering is also called **Kalman filtering**

→ smoothing is also called **Kalman smoothing**

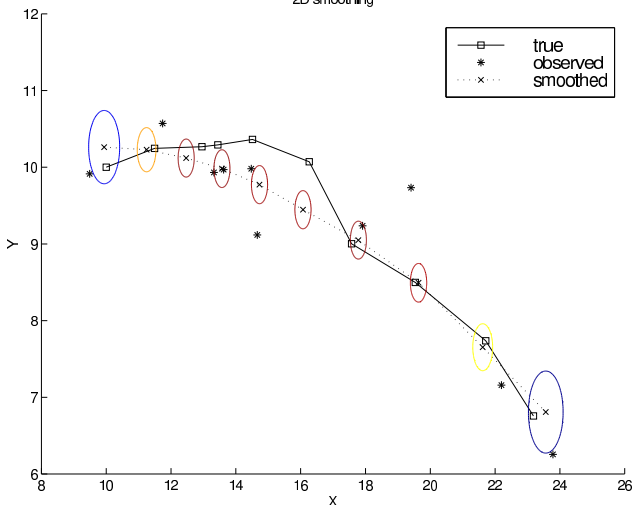
# Kalman Filter example

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



# Kalman Filter example

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



# HMM example: Learning Bach

- A machine “listens” (reads notes of) Bach pieces over and over again  
→ 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 shows two musical staves, (a) and (b), illustrating hidden state representations for harmonisation and ornamentation. Both staves are in G major (one sharp) and 4/4 time. Staff (a) is labeled "16:12:7:0/T" and shows a Soprano melody with a whole note G4 and a half note A4. The other staves (Alto, Tenor, Bass) show a simple harmonic accompaniment with whole notes: Alto (B4), Tenor (D5), and Bass (G3). Staff (b) is labeled "0,2,2/0,2,2/0,0,0" and shows the same Soprano melody. The other staves show a more complex harmonic accompaniment with eighth notes: Alto (B4, A4), Tenor (G4, F#4), and Bass (G3, F#3).

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