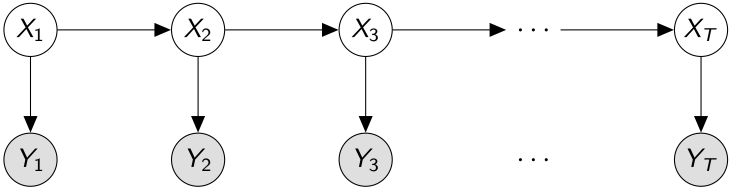
Tutorial 8: Sequential Models

Sebastian Curi

Probabilistic Artificial Inteligence

November 15, 2019

Problem Setting

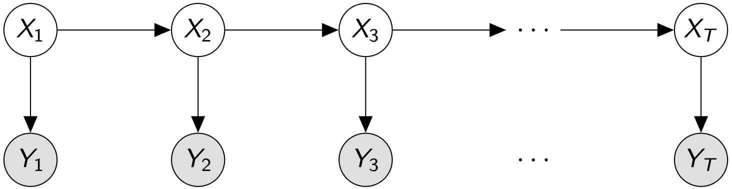


This factorizes as:

$$P(X_{1:T}, Y_{1:T}) = P(X_1) \prod_{t=2}^{T} P(X_t|X_{t-1}) \prod_{t=1}^{T} P(Y_t|X_t)$$

- $P(X_1)$ is called the initial distribution.
- $P(X_t|X_{t-1})$ is called the transition model.
- $P(Y_t|X_t)$ is called the measurement model.

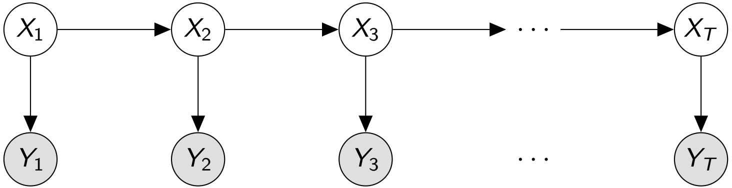
Inference Tasks



- Marginalization (Smoothing): $P(X_t|y_{1:T})$
  - Prediction: $P(X_t|y_{1:t-1})$
  - Filtering: $P(X_t|y_{1:t})$
- Most Probable Explanation (MPE): $\arg\max_{x_{1:T}} P(X_{1:T} = x_{1:T}|y_{1:T})$

Note that MPE is not (in general) the arg max of the smoothed marginal.

Filtering $P(X_t|y_{1:t})$



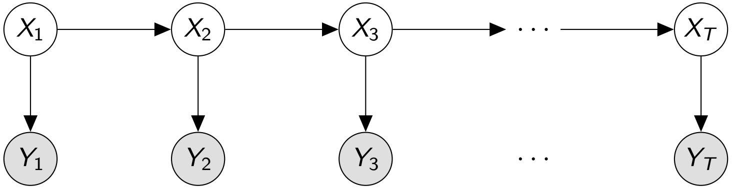
Assume we have predicted $P(X_t|y_{1:t-1})$.

$$P(X_t|y_{1:t}) = \frac{1}{Z} P(X_t|y_{1:t-1}) P(y_t|X_t)$$

$$Z = \sum_{x'} P(X_t = x'|y_{1:t-1}) P(y_t|X_t = x')$$

- When X is discrete, $O(|X|)$.

Prediction $P(X_{t+1}|y_{1:t})$

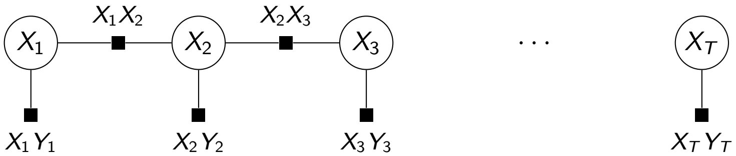


Assume we have filtered $P(X_t|y_{1:t})$

$$\begin{aligned} P(X_{t+1}|y_{1:t}) &= \sum_{x'} P(X_{t+1}, X_t = x'|y_{1:t}) \\ &= \sum_{x'} P(X_{t+1}|X_t = x')P(X_t = x'|y_{1:t}) \end{aligned}$$

- When X is discrete, $O(|X|^2)$.

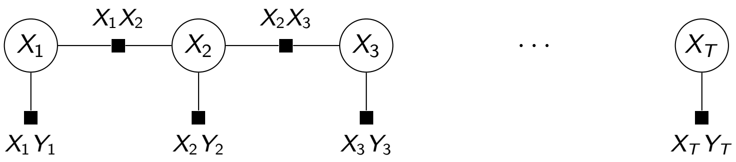
Smoothing $P(X_t|y_{1:T})$: Sum Product Algorithm!



Sum-Product Algorithm converges as Graphical Model is a Polytree

1. Select X_1 as root.
2. Nodes are X_t and Factors are $[X_t, Y_t]$ and $[X_t, X_{t+1}]$.
3. Set $f_{[X_t,Y_t]}(x, y) = P(Y_t = y|X_t = x)[[y = y_t]]$.
4. Set $f_{[X_t,X_{t+1}]}(x, x') = P(X_{t+1} = x'|X_t = x)$.
5. Propagate Messages from root to leaves.
6. Propagate Messages from leaves to root.

Smoothing $P(X_t|y_{1:T})$: Forwards Pass



For $t = 1, \ldots T$

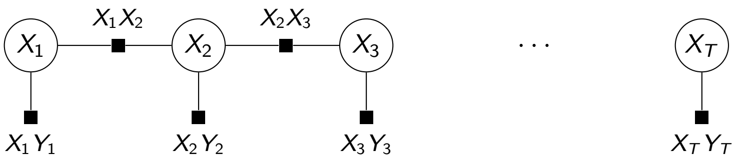
- Message from node X_t to factors $[X_t, X_{t+1}]$ **Filtering!**:

$$\mu_{X_t \to [X_t, X_{t+1}]}(x) = \mu_{[X_{t-1}, X_t] \to X_t}(x)\mu_{[X_t, Y_t] \to X_t}(x)$$
$$\propto P(X_t = x|y_{1:t-1})P(y_t|X_t = x) \propto P(X_t = x|y_{1:t})$$

- Message from factors $[X_t, X_{t+1}]$ to node X_{t+1} **Prediction!**:

$$\mu_{[X_t, X_{t+1}] \to X_{t+1}}(x) = \sum_{x'} f_{[X_t, X_{t+1}]}(x', x)\mu_{X_t \to [X_t, X_{t+1}]}(x')$$
$$= \sum_{x'} P(X_{t+1} = x|X_t = x')P(X_t = x'|y_{1:t}) = P(X_{t+1} = x|y_{1:t})$$

Smoothing $P(X_t|y_{1:T})$: Backwards Pass



For $t = T, \ldots 1$

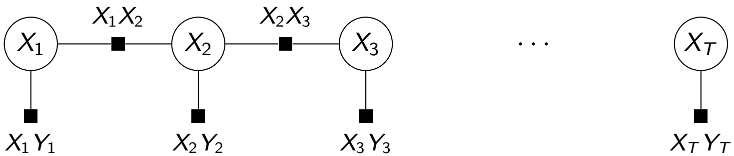
- Message from node X_{t+1} to factors $[X_t, X_{t+1}]$ **Filtering!**:

$$\mu_{X_{t+1}\to[X_t,X_{t+1}]}(x) = \mu_{[X_t,X_{t+1}]\to X_t}(x)\mu_{[X_t,Y_t]\to X_t}(x)$$
$$\propto P(X_t = x|y_{t+1:T})P(y_t|X_t = x) \propto P(X_t = x|y_{t:T})$$

- Message from factors $[X_t, X_{t+1}]$ to node X_t **Prediction!**:

$$\mu_{[X_t,X_{t+1}]\to X_t}(x) = \sum_{x'} f_{[X_t,X_{t+1}]}(x, x')\mu_{X_{t+1}\to[X_t,X_{t+1}]}(x')$$
$$= \sum_{x'} P(X_t = x|X_{t+1} = x')P(X_{t+1} = x'|y_{t+1:T}) = P(X_t = x|y_{t+1:T})$$

Calculating Marginal $P(X_t|y_{1:T})$



$$
\begin{aligned}
P(X_t = x|y_{1:T}) &\propto \mu_{[X_{t-1},X_t]\to X_t}(x)\mu_{[X_t,X_{t+1}]\to X_t}(x)\mu_{[X_t,Y_t]\to X_t}(x) \\
&\propto P(X_t = x|y_{1:t-1})P(X_t = x|y_{t+1:T})P(y_t|X_t = x) \\
&\propto P(X_t = x|y_{1:t})P(X_t = x|y_{t+1:T}) \\
&\propto P(X_t = x|y_{1:t-1})P(X_t = x|y_{t:T})
\end{aligned}
$$

How does the Max-Product Algorithm look like?

Continuous State Spaces

- Prediction $O(|X|^2)$, Filtering $O(|X|)$. In general, intractable.
- Idea 1: Exact Distributions (when Tractable).
- Idea 2: Approximate Distribution with Particles (Monte Carlo).
- Idea 3: Approximate Distribution with Parametric Function (VI).

Particle Approximation

1. Approximate $P(X_t|y_{1:t})$ with K particles $x_t^{(i)}$, with weight $w_t^{(i)}$.

$$P(X_t = x|y_{1:t}) \approx \frac{1}{K}\sum_{i=1}^{K} w_t^{(i)} \delta_{x_t^{(i)}}(x)$$

2. **Prediction**: Propagate each particle, $x'^{(i)} \sim P(X_{t+1}|X_t = x_t^{(i)})$.

$$P(X_{t+1} = x|y_{1:t}) = \int P(X_{t+1} = x|X_t = x')P(X_t = x'|y_{1:t})dx'$$

$$\approx \int P(X_{t+1} = x|X_t = x')\frac{1}{K}\sum_{i=1}^{K} w_t^{(i)} \delta_{x_t^{(i)}}(x')dx'$$

$$= \frac{1}{K}\sum_{i=1}^{K} w_t^{(i)} \int P(X_{t+1} = x|X_t = x')\delta_{x_t^{(i)}}(x')dx'$$

$$= \frac{1}{K}\sum_{i=1}^{K} w_t^{(i)} P(X_{t+1} = x|X_t = x_t^{(i)}) \approx \frac{1}{K}\sum_{i=1}^{K} w_t^{(i)} \delta_{x'^{(i)}}(x)$$

Particle Approximation

3. **Filter**: Re-weight each particle with $w_i \propto P(y_t|X_t = x_i)$

$$
\begin{aligned}
P(X_t = x|y_{1:t}) &\propto P(X_t = x|y_{1:t-1})P(y_t|X_t = x) \\
&\approx \frac{1}{K}\sum_{i=1}^{K} w_t^{(i)} \delta_{x'^{(i)}}(x) P(y_t|X_t = x) \\
&= \frac{1}{K}\sum_{i=1}^{K} P(y_t|X_t = x_{t+1}^{(i)}) w_t^{(i)} \delta_{x'^{(i)}}(x) \\
&= \frac{1}{K}\sum_{i=1}^{K} w_{t+1}^{(i)} \delta_{x_{t+1}^{(i)}}(x)
\end{aligned}
$$

4. **Resampling:** Particle Starvation, Numeric Stability, Multi-Modality
   - $w_{t+1}^{(i)} = P(y_t|X_t = x'^{(i)})w_t^{(i)}$ and $x_{t+1}^{(i)} = x'^{(i)}$
   - $w_{t+1}^{(i)} = 1$ and $x_{t+1}^{(i)} \sim \sum_{i=1}^{K} P(y_t|X_t = x'^{(i)})\delta_{x'^{(i)}}(x)$
5. **Inference** Particle sum-product and max-product algorithms.

Exact Distribution: Linear Gaussian Systems

- Initial Distribution: $x_0 \sim \mathcal{N}(\mu_0, \Sigma_0)$
- Transition: $x_{t+1} = Fx_t + \varepsilon_t$, with $\varepsilon_t \sim \mathcal{N}(0, \Sigma_x)$
- Measurement: $y_t = Hx_t + \eta_t$, with $\eta_t \sim \mathcal{N}(0, \Sigma_y)$.

Predict $P(X_{t+1}|y_{1:t})$ from $P(x_t|y_{1:t}) = \mathcal{N}(\mu_t, \Sigma_t)$:

$$P(X_{t+1}|y_{1:t}) = \mathcal{N}(F\mu_t, F\Sigma_t F^\top + \Sigma_x)$$

Filter $P(X_{t+1}|y_{1:t+1})$ from $P(X_{t+1}|y_{1:t}) = \mathcal{N}(F\mu_t, F\Sigma_t F^\top + \Sigma_x)$:

$$\begin{aligned}
P(X_{t+1}|y_{1:t+1}) = \mathcal{N}(\mu_{t+1}, \Sigma_{t+1})& \\
\mu_{t+1} &= F\mu_t + K_{t+1}(y_t - HF\mu_t) \\
\Sigma_{t+1} &= (I - K_{t+1}H)(F\Sigma_t F^\top + \Sigma_x) \\
K_{t+1} &= (F\Sigma_t F^\top + \Sigma_x)H^\top (H(F\Sigma_t F^\top + \Sigma_x)H^\top + \Sigma_y)^{-1} \\
\text{``} &= \frac{\sigma_t^2 + \sigma_x^2}{\sigma_t^2 + \sigma_x^2 + \sigma_y^2}\text{''}
\end{aligned}$$

Approximate Distribution: Non-Linear Systems

- **Transition**: $x_{t+1} = f(x_t) + \varepsilon_t$, with $\varepsilon_t \sim \mathcal{N}(0, \Sigma_x)$.
- **Challenge**: Even if $P(X_t|y_{1:t})$ is Gaussian, $P(x_{t+1}|y_{1:t})$ is **not**.
- **Approximation**: $P(x_{t+1}|y_{1:t}) \approx Q = \mathcal{N}(\mu, \Sigma)$

Extended Kalman Filter

- Mean: $\mu = f(\mu_t)$.
- Covariance (linear approx): $\Sigma = \hat{F}\Sigma_t\hat{F}^\top + \Sigma_x$, where $\hat{F} = \frac{\partial f(x)}{\partial x}\Big|_{x=x_t}$

Unscented Kalman Filter

- Select (or sample) K points from $P(X_t|y_{1:t}) = \mathcal{N}(\mu_t, \Sigma_t)$.
- Propagate points with dynamics $x_i' \sim f(x_i) + w$.
- Approximate μ and Σ via moment matching.

Assumed Density Filtering

- Minimize $D(P||Q)$ w.r.t. μ and Σ.

True/False Questions

- Kalman Filters are derived from an Assumed Density Filtering because it assumes a Gaussian state.
  False: We are not assuming that the posterior is Gaussian, it really is because of the model.
- The Extended Kalman Filter is an Assumed Density Filter.
  False: The Extended Kalman Filter is not minimizing the KL-Divergence but using a Linear Approximation.
- The Unscented Kalman Filter is an Assumed Density Filter.
  True: Moment Matching minimizes the KL-Divergence between any distribution and a Gaussian.
- The Unscented Kalman Filter is a Particle Filter.
  False Even if the Unscented Kalman Filter uses particles, Particle Filters do not assume any parametric distribution.