Homework 3 out, due Wed Nov 24
  - Questions 1 & 3 already covered
  - MDPs (question 2) coming up Wednesday

Code for project final released later today
Temporal models

So far: “Static” models (no notion of time)
  - Variables don’t change values

In practice:
  - World changes over time
  - Want to “keep track” of change by using probabilistic inference

Basic idea: Create “copies” of variables, one per time step
Dynamical models

- Static model
  

Rain

- Dynamic model

$\text{Rain}_1 \rightarrow \text{Rain}_2 \rightarrow \text{Rain}_3 \rightarrow \ldots$

- Assumes *discrete, unit-length* time steps!
Markov chains

Markov assumption:
- 1st order MC: $X_t \perp X_{t-2} \mid X_{t-1}$
- k-th order MC: $X_t \perp X_{t-k-1} \mid X_{t-k:t-1}$

Stationarity assumption:
- 1st order: $P(X_t \mid X_{t-1})$ independent of $t$
- k-th order: $P(X_t \mid X_{t-k:t-1})$ constant
E.g.: Given that it rains now, how likely is it to rain a week from now?

Sps. I know \( P(X_t) \), \( P(X_t | X_{t-1}) \)

What is \( P(X_t) \)?

\[
P(X_t) = \sum_{X_{1:t-1}} P(X_{1:t}) = \sum_{X_{1:t-1}} \frac{P(X_t | X_{1:t-1}) \cdot P(X_{1:t-1})}{P(X_{t-1}) \cdot P(X_{1:t-2} | X_{t-1})}
\]

\[
= \sum_{X_{t-1}} P(X_t | X_{t-1}) \cdot P(X_{t-1}) \cdot \sum_{X_{1:t-2} | X_{t-1}} P(X_{1:t-2} | X_{t-1})
\]

\[
= \sum_{X_{t-1}} P(X_t | X_{t-1}) \cdot P(X_{t-1}) \cdot \frac{1}{P(X_{t-1})}
\]
Prediction in Markov Chains

\[ P_t = P(X_t) = \sum_{X_{t-1}} P(X_{t-1}) P(X_t | X_{t-1}) \]

\[ P_t = p_{t-1} T \]

\[ p_0 T^t \]

\[ T = \begin{pmatrix} P(1|1) & \ldots & P(k|1) \\ \vdots & \ddots & \vdots \\ P(1|k) & \ldots & P(k|k) \end{pmatrix} \]

"transition matrix"
HMMs / Kalman Filters

- Most famous Bayesian networks:
  - Naïve Bayes model
  - Hidden Markov model
  - Kalman Filter

- Hidden Markov models
  - Speech recognition
  - Sequence analysis in comp. bio

- Kalman Filters control
  - Cruise control in cars
  - GPS navigation devices
  - Tracking missiles..

- Very simple models but very powerful!!
HMMs / Kalman Filters

- $X_1, \ldots, X_T$: Unobserved (hidden) variables
- $Y_1, \ldots, Y_T$: Observations
- **HMMs**: $X_i$ Multinomial, $Y_i$ multinomial (or arbitrary)
- **Kalman Filters**: $X_i$, $Y_i$ Gaussian distributions
HMMs for speech recognition

Words

\[ X_1 \rightarrow X_2 \rightarrow X_3 \rightarrow X_4 \rightarrow X_5 \rightarrow X_6 \]

Phoneme

\[ Y_1 \rightarrow Y_2 \rightarrow Y_3 \rightarrow Y_4 \rightarrow Y_5 \rightarrow Y_6 \]

“He ate the cookies on the couch”
Example: Umbrella world

Rain

<table>
<thead>
<tr>
<th>$R_{t-1}$</th>
<th>$P(R)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\downarrow$</td>
<td>$\uparrow$</td>
</tr>
<tr>
<td>$\downarrow$</td>
<td>$\uparrow$</td>
</tr>
<tr>
<td>$\downarrow$</td>
<td>$\uparrow$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$R_t$</th>
<th>$P(U_t)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t$</td>
<td>0.9</td>
</tr>
<tr>
<td>$f$</td>
<td>0.2</td>
</tr>
</tbody>
</table>

$P(R_t | R_{t-1}) = 0.7$

Rain

Umbrella

Umbrella

Umbrella

Umbrella

Umbrella

Umbrella

Umbrella
Inference tasks

Filtering \[ P(X_T | Y_{1:T}) \]

Prediction \[ P(X_{T+\delta} | Y_{1:T}) \]

Smoothing \[ P(X_t | Y_{1:T}) \] \(1 \leq t \leq T\)

Most probable explanation \[ \arg\max_{X_{1:T}} P(X_{1:T} | Y_{1:T}) \]
Inference in Hidden Markov Models

- **Inference:**
  - In principle, can use variable elimination / belief propagation
  - New variables $X_t, Y_t$ at each time step $\rightarrow$ need to rerun
  - Complexity grows with time!!

- **Bayesian Filtering:**
  - Suppose we already have computed $P(X_t \mid y_{1,...,t})$
  - Want to efficiently (recursively) compute $P(X_{t+1} \mid y_{1,...,t+1})$

\[
P(X_{t+1} (y_{1:t+1}) = \mathbb{I} (y_{t+1} \mid P(X_t | y_{1:t}))
\]
Bayesian filtering

- Start with $P(X_1)$
- At time $t$
  - Assume we have $P(X_t \mid y_{1\ldots t-1})$
  - **Conditioning:** $P(X_t \mid y_{1\ldots t})$
    \[
    P(X_t \mid y_{1\ldots t}) = \frac{1}{2} P(x_t \mid y_{1\ldots t-1}) \cdot P(y_t \mid x_t)
    \]
  - **Prediction:** $P(X_{t+1} \mid y_{1\ldots t})$
    \[
    P(X_{t+1} \mid y_{1\ldots t}) = \sum_{X_t} P(X_{t+1}, x_t \mid y_{1\ldots t})
    \]
    \[
    = \sum_{X_t} P(x_t \mid y_{1\ldots t}) \frac{P(x_{t+1} \mid x_t, y_{1\ldots t})}{P(x_{t+1} \mid x_t)}
    \]
    \[
    For \ k \ states, \ can \ do \ filtering \ in \ O(k^4)
    \]

\[
\text{have} \quad P(x_t)
\]
\[
\text{want} \quad P(x_t \mid x_t)
\]
\[
= \frac{1}{2} P(y_t \mid x_t) \cdot P(y_t)
\]
\[
\text{Bayes' rule}
\]
Understanding Bayesian filtering

![Bayesian filtering diagram](image)

- **True** 0.500 → 0.500 → 0.627
- **False** 0.500 → 0.500 → 0.373
- **Prediction** 0.818 → 0.182
- **Conditioning** 0.883 → 0.117

**Rain** → **Umbrella**
HMM for robot localization

(a) Posterior distribution over robot location after $E_1 = \text{NSW}$

(b) Posterior distribution over robot location after $E_1 = \text{NSW}, \ E_2 = \text{NS}$
Prediction in HMMs

- Have: $P(X_t \mid y_{1:t})$
- Want: $P(X_{t+k} \mid y_{1:t})$

Just leave out conditioning! $X_{t:t+k} \mid y_{1:t}$ is MC!

\[ p_t = P(X_t \mid y_{1:t}) \]
\[ p_{t+k} = p_t \cdot T^k \]

\[ T = \begin{pmatrix}
    P(111) & \cdots & P(1k1) \\
    \vdots & \ddots & \vdots \\
    P(11k) & \cdots & P(k1k)
\end{pmatrix} \]
Smoothing / MPE

- **Smoothing:** \( P(X_t \mid y_{1:T}) \)

- **Most probable explanation:** \( \arg \max_{x_{1:T}} P(x_{1:T} \mid y_{1:T}) \)

- **HMM is polytree Bayesian network!**

- **Can use sum product** (aka forward-backward) for smoothing and **max product** (aka Viterbi algo) for MPE

- **Specialized implementations using matrix algebra**
Kalman filters

- Track objects in *continuous domain* using noisy measurements
  - E.g., birds flying, robots moving, chemical plants, ...

- System described using Gaussian variables
  - E.g., location in X,Y,Z; velocity in X,Y,Z; acceleration in X,Y,Z,...

![Diagram of Kalman filter system]
Bivariate Gaussian distribution

\[
\frac{1}{2\pi \sqrt{|\Sigma|}} \exp \left( -\frac{1}{2} (y - \mu)^T \Sigma^{-1} (y - \mu) \right)
\]

\[
\Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}
\]

\[
\Sigma = \begin{pmatrix} 1 & 0.9 \\ 0.9 & 1 \end{pmatrix}
\]
Multivariate Gaussian distribution

\[
\mathcal{N}(y; \Sigma, \mu) = \frac{1}{(2\pi)^{n/2}\sqrt{|\Sigma|}} \exp \left( -\frac{1}{2} (y - \mu)^T \Sigma^{-1} (y - \mu) \right)
\]

\[
\Sigma = \begin{pmatrix}
\sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1n} \\
\sigma_{12} & \sigma_2^2 & \cdots & \sigma_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
\sigma_{1n} & \sigma_{2n} & \cdots & \sigma_n^2
\end{pmatrix}
\]
Kalman Filters (Gaussian HMMs)

- \( X_1, \ldots, X_T \): Location of object being tracked \( \in \mathbb{R}^d \)
- \( Y_1, \ldots, Y_T \): Observations \( \in \mathbb{R}^{d'} \)
- \( P(X_1) \): Prior belief about location at time 1
- \( P(X_{t+1} | X_t) \): “Motion model”
  - How do I expect my target to move in the environment?
    \[
    X_{t+1} = FX_t + \varepsilon_t \text{ where } \varepsilon_t \in \mathcal{N}(0, \Sigma_x)
    \]
- \( P(Y_t | X_t) \): “Sensor model”
  - What do I observe if target is at location \( X_t \)?
    \[
    Y_t = HX_t + \eta_t \text{ where } \eta_t \in \mathcal{N}(0, \Sigma_y)
    \]
Bayesian filtering in KFs (1D)

- Start with $P(X_1)$
- At time $t$
  - Assume we have $P(X_t \mid y_{1:t-1})$
  - Conditioning: $P(X_t \mid y_{1:t})$

\[
P(X_t \mid y_{1:t}) = \frac{1}{Z} \cdot P(X_t \mid y_{1:t-1}) P(y_t \mid X_t)
\]

- Prediction: $P(X_{t+1} \mid y_{1:t})$
\[
P(X_{t+1} \mid y_{1:t}) = \int P(X_{t+1} \mid X_t) P(X_t \mid y_{1:t}) \, dX_t
\]
Example: Simple random walk

- **Transition / motion model**
  \[ P(x_{t+1} \mid x_t) = \mathcal{N}(x_t, \sigma_x^2) \]
  \[ x_{t+1} = x_t + \epsilon_t \quad , \quad \epsilon_t \sim \mathcal{N}(0, \sigma_x^2) \]

- **Sensor model**
  \[ P(y_t \mid x_t) = \mathcal{N}(x_t, \sigma_y^2) \]
  \[ y_t = x_t + \eta_t \quad , \quad \eta_t \sim \mathcal{N}(0, \sigma_y^2) \]

- **State at time t:**
  \[ P(x_t \mid y_{1:t}) = \mathcal{N}(\mu_t, \sigma_t^2) \]
Example: Bayesian filtering in KFs

\[ \mu_{t+1} = \frac{\sigma_y^2 \mu_t + (\sigma_t^2 + \sigma_x^2) y_{t+1}}{\sigma_t^2 + \sigma_x^2 + \sigma_y^2} \]

\[ \sigma_{t+1}^2 = \frac{(\sigma_t^2 + \sigma_x^2) \sigma_y^2}{\sigma_t^2 + \sigma_x^2 + \sigma_y^2} \]

Suppose: \( P(x_t \mid y_{1:t}) = \mathcal{N}(\mu_t, \sigma_t^2) \)
General Kalman update

- Transition model
  \[ P(x_{t+1} \mid x_t) = \mathcal{N}(x_{t+1}; Fx_t, \Sigma_x) \]

- Sensor model
  \[ P(y_t \mid x_t) = \mathcal{N}(y_t; Hx_t, \Sigma_y) \]

- Kalman Update:
  \[ \mu_{t+1} = F\mu_t + K_{t+1}(y_{t+1} - HF\mu_t) \]
  \[ \Sigma_{t+1} = (I - K_{t+1})(F\Sigma_tF^T + \Sigma_x) \]

- Kalman gain:
  \[ K_{t+1} = (F\Sigma_tF^T + \Sigma_x)H^T(H(F\Sigma_tF^T + \Sigma_x)H^T + \Sigma_y)^{-1} \]

- Can compute \( \Sigma_t \) and \( K_t \) offline
2D tracking example
Kalman smoothing

2D filtering

2D smoothing

(a)  (b)
When KFs fail

KFs assume transition model is **linear**
- Implies that predictive distribution is Gaussian (unimodal)
- Need approximate inference to capture non-linearities!
Factored dynamical models

- So far: HMMs and Kalman filters

What if we have more than one variable at each time step?
- E.g., temperature at different locations, or road conditions in a road network?

⇒ Dynamic Bayesian Networks
Dynamic Bayesian Networks

- At every timestep have a *Bayesian Network*

- Variables at each time step $t$ called a slice $S_t$
- "Temporal" edges connecting $S_{t+1}$ with $S_t"
Flow of influence in DBNs

$A_1 \perp B_1 \checkmark$

$A_2 \perp B_2 \times$

$A_2 \perp C_2 \checkmark$

$A_3 \perp C_3 \times$
Inference in DBNs?

**DBN**

- \( A_1 \) → \( A_2 \)
- \( B_1 \) → \( B_2 \)
- \( C_1 \) → \( C_2 \)
- \( D_1 \) → \( D_2 \)

**Marginals at time 2**

- \( A_2 \)
- \( B_2 \)
- \( C_2 \)
- \( D_2 \)

*Need approximate inference!*
Particle filtering

- Very useful approximate inference technique for dynamical models
  - Nonlinear Kalman filters
  - Dynamic Bayesian networks

- **Basic idea:** Approximate the posterior at each time by samples (particles), which are propagated and reweighted over time