Probability, Markov models and HMMs

Vibhav Gogate
The University of Texas at Dallas

CS 6364

Many slides over the course adapted from either Dan Klein, Luke Zettlemoyer, Stuart Russell and Andrew Moore
Outline

- Probability review
  - Random Variables and Events
  - Joint / Marginal / Conditional Distributions
  - Product Rule, Chain Rule, Bayes’ Rule
  - Probabilistic Inference

- Probabilistic sequence models (and inference)
  - Markov Chains
  - Hidden Markov Models
  - Particle Filters
Probability Review

- **Probability**
  - Random Variables
  - Joint and Marginal Distributions
  - Conditional Distribution
  - Product Rule, Chain Rule, Bayes’ Rule
  - Inference

- You’ll need all this stuff A LOT for the next few weeks, so make sure you go over it now!
Inference in Ghostbusters

- A ghost is in the grid somewhere.
- Sensor readings tell how close a square is to the ghost:
  - On the ghost: red
  - 1 or 2 away: orange
  - 3 or 4 away: yellow
  - 5+ away: green
- Sensors are noisy, but we know $P(\text{Color} | \text{Distance})$.

| $P(\text{red} | 3)$ | $P(\text{orange} | 3)$ | $P(\text{yellow} | 3)$ | $P(\text{green} | 3)$ |
|-------------------|-----------------------|-----------------------|----------------------|
| 0.05              | 0.15                  | 0.5                   | 0.3                  |
Random Variables

- A *random variable* is some aspect of the world about which we (may) have uncertainty
  - \( R \) = Is it raining?
  - \( D \) = How long will it take to drive to work?
  - \( L \) = Where am I?

- We denote random variables with capital letters

- Random variables have domains
  - \( R \) in \{true, false\}
  - \( D \) in \([0, \infty)\)
  - \( L \) in possible locations, maybe \{(0,0), (0,1), …\}
Joint Distributions

- A *joint distribution* over a set of random variables: $X_1, X_2, \ldots X_n$ specifies a real number for each assignment (or *outcome*):

$$P(X_1 = x_1, X_2 = x_2, \ldots X_n = x_n)$$

$$P(x_1, x_2, \ldots x_n)$$

- Size of distribution if $n$ variables with domain sizes $d$?
- Must obey: $P(x_1, x_2, \ldots x_n) \geq 0$

$$\sum_{(x_1, x_2, \ldots x_n)} P(x_1, x_2, \ldots x_n) = 1$$

- A *probabilistic model* is a joint distribution over variables of interest
- For all but the smallest distributions, impractical to write out

<table>
<thead>
<tr>
<th></th>
<th>T</th>
<th>W</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>hot</td>
<td>sun</td>
<td>0.4</td>
<td></td>
</tr>
<tr>
<td>hot</td>
<td>rain</td>
<td>0.1</td>
<td></td>
</tr>
<tr>
<td>cold</td>
<td>sun</td>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td>cold</td>
<td>rain</td>
<td>0.3</td>
<td></td>
</tr>
</tbody>
</table>
Events

- An *outcome* is a joint assignment for all the variables

\[(x_1, x_2, \ldots x_n)\]

- An *event* is a set E of outcomes

\[P(E) = \sum_{(x_1 \ldots x_n) \in E} P(x_1 \ldots x_n)\]

- From a joint distribution, we can calculate the probability of any event

  - Probability that it’s hot AND sunny?
  - Probability that it’s hot?
  - Probability that it’s hot OR sunny?
Marginal Distributions

- Marginal distributions are sub-tables which eliminate variables
- Marginalization (summing out): Combine collapsed rows by adding

\[ P(X_1 = x_1) = \sum_{x_2} P(X_1 = x_1, X_2 = x_2) \]

### P(T, W)

<table>
<thead>
<tr>
<th>T</th>
<th>W</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>hot</td>
<td>sun</td>
<td>0.4</td>
</tr>
<tr>
<td>hot</td>
<td>rain</td>
<td>0.1</td>
</tr>
<tr>
<td>cold</td>
<td>sun</td>
<td>0.2</td>
</tr>
<tr>
<td>cold</td>
<td>rain</td>
<td>0.3</td>
</tr>
</tbody>
</table>

### P(T)

<table>
<thead>
<tr>
<th>T</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>hot</td>
<td>0.5</td>
</tr>
<tr>
<td>cold</td>
<td>0.5</td>
</tr>
</tbody>
</table>

### P(W)

<table>
<thead>
<tr>
<th>W</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>sun</td>
<td>0.6</td>
</tr>
<tr>
<td>rain</td>
<td>0.4</td>
</tr>
</tbody>
</table>
Conditional Distributions

- Conditional distributions are probability distributions over some variables given fixed values of others.

### Conditional Distributions

\[
P(W|T = \text{hot})
\]

<table>
<thead>
<tr>
<th>W</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>sun</td>
<td>0.8</td>
</tr>
<tr>
<td>rain</td>
<td>0.2</td>
</tr>
</tbody>
</table>

\[
P(W|T = \text{cold})
\]

<table>
<thead>
<tr>
<th>W</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>sun</td>
<td>0.4</td>
</tr>
<tr>
<td>rain</td>
<td>0.6</td>
</tr>
</tbody>
</table>

### Joint Distribution

\[
P(T, W)
\]

<table>
<thead>
<tr>
<th>T</th>
<th>W</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>hot</td>
<td>sun</td>
<td>0.4</td>
</tr>
<tr>
<td>hot</td>
<td>rain</td>
<td>0.1</td>
</tr>
<tr>
<td>cold</td>
<td>sun</td>
<td>0.2</td>
</tr>
<tr>
<td>cold</td>
<td>rain</td>
<td>0.3</td>
</tr>
</tbody>
</table>

\[
P(x_1|x_2) = \frac{P(x_1, x_2)}{P(x_2)}
\]
Normalization Trick

A trick to get a whole conditional distribution at once:
- Select the joint probabilities matching the evidence
- Normalize the selection (make it sum to one)

Why does this work? Sum of selection is \( P(\text{evidence})! \) (\( P(r) \), here)

\[
P(x_1 | x_2) = \frac{P(x_1, x_2)}{P(x_2)} = \frac{P(x_1, x_2)}{\sum_{x_1} P(x_1, x_2)}
\]
Probabilistic Inference

- **Probabilistic inference**: compute a desired probability from other known probabilities (e.g. conditional from joint)

- We generally compute conditional probabilities
  - $P(\text{on time} \mid \text{no reported accidents}) = 0.90$
  - These represent the agent’s *beliefs* given the evidence

- Probabilities change with new evidence:
  - $P(\text{on time} \mid \text{no accidents, 5 a.m.}) = 0.95$
  - $P(\text{on time} \mid \text{no accidents, 5 a.m., raining}) = 0.80$
  - Observing new evidence causes *beliefs to be updated*
## Inference by Enumeration

- **P(sun)?**

- **P(sun | winter)?**

- **P(sun | winter, warm)?**

<table>
<thead>
<tr>
<th>S</th>
<th>T</th>
<th>W</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>summer</td>
<td>hot</td>
<td>sun</td>
<td>0.30</td>
</tr>
<tr>
<td>summer</td>
<td>hot</td>
<td>rain</td>
<td>0.05</td>
</tr>
<tr>
<td>summer</td>
<td>cold</td>
<td>sun</td>
<td>0.10</td>
</tr>
<tr>
<td>summer</td>
<td>cold</td>
<td>rain</td>
<td>0.05</td>
</tr>
<tr>
<td>winter</td>
<td>hot</td>
<td>sun</td>
<td>0.10</td>
</tr>
<tr>
<td>winter</td>
<td>hot</td>
<td>rain</td>
<td>0.05</td>
</tr>
<tr>
<td>winter</td>
<td>cold</td>
<td>sun</td>
<td>0.15</td>
</tr>
<tr>
<td>winter</td>
<td>cold</td>
<td>rain</td>
<td>0.20</td>
</tr>
</tbody>
</table>
Inference by Enumeration

- **General case:**
  - Evidence variables: \( E_1 \ldots E_k = e_1 \ldots e_k \)
  - Query* variable: \( Q \)
  - Hidden variables: \( H_1 \ldots H_r \)

- **We want:** \( P(Q|e_1 \ldots e_k) \)
- **First,** select the entries consistent with the evidence
- **Second,** sum out \( H \) to get joint of Query and evidence:

\[
P(Q, e_1 \ldots e_k) = \sum_{h_1 \ldots h_r} P(Q, h_1 \ldots h_r, e_1 \ldots e_k) \]

- **Finally,** normalize the remaining entries to conditionalize

- **Obvious problems:**
  - Worst-case time complexity \( O(d^n) \)
  - Space complexity \( O(d^n) \) to store the joint distribution
The Product Rule

Sometimes have conditional distributions but want the joint

\[ P(x|y) = \frac{P(x, y)}{P(y)} \quad \Longleftrightarrow \quad P(x, y) = P(x|y)P(y) \]

Example:

\[
\begin{array}{c|c|c|c}
P(W) & D & W & P \\
\hline
R & wet & sun & 0.1 \\
& dry & sun & 0.9 \\
& wet & rain & 0.7 \\
& dry & rain & 0.3 \\
\end{array}
\]

\[
\begin{array}{c|c|c|c}
P(D, W) & D & W & P \\
\hline
wet & sun & 0.08 \\
dry & sun & 0.72 \\
wet & rain & 0.14 \\
dry & rain & 0.06 \\
\end{array}
\]
The Chain Rule

- More generally, can always write any joint distribution as an incremental product of conditional distributions

\[
P(x_1, x_2, x_3) = P(x_1)P(x_2|x_1)P(x_3|x_1, x_2)
\]

\[
P(x_1, x_2, \ldots x_n) = \prod_i P(x_i|x_1 \ldots x_{i-1})
\]

- Why is this always true?
Bayes’ Rule

- Two ways to factor a joint distribution over two variables:

\[ P(x, y) = P(x|y)P(y) = P(y|x)P(x) \]

- Dividing, we get:

\[ P(x|y) = \frac{P(y|x)P(x)}{P(y)} \]

- Why is this at all helpful?
  - Lets us build one conditional from its reverse
  - Often one conditional is tricky but the other one is simple
  - Foundation of many systems we’ll see later

- In the running for most important AI equation!
Inference with Bayes’ Rule

- Example: Diagnostic probability from causal probability:

\[ P(\text{Cause}|\text{Effect}) = \frac{P(\text{Effect}|\text{Cause})P(\text{Cause})}{P(\text{Effect})} \]

- Example:
  - m is meningitis, s is stiff neck
    \[
P(s|m) = 0.8 \quad P(m) = 0.0001 \quad P(s) = 0.1
    \]
    \[
P(m|s) = \frac{P(s|m)P(m)}{P(s)} = \frac{0.8 \times 0.0001}{0.1} = 0.0008
    \]
    - Note: posterior probability of meningitis still very small
    - Note: you should still get stiff necks checked out! Why?
Let’s say we have two distributions:

- **Prior distribution** over ghost location: \( P(G) \)
  - Let’s say this is uniform
- **Sensor reading model**: \( P(R \mid G) \)
  - Given: we know what our sensors do
  - \( R = \) reading color measured at \((1,1)\)
  - E.g. \( P(R = \text{yellow} \mid G=(1,1)) = 0.1 \)

We can calculate the **posterior distribution** \( P(G|r) \) over ghost locations given a reading using Bayes’ rule:

\[
P(g \mid r) \propto P(r \mid g)P(g)
\]
Independence

- Two variables are *independent* in a joint distribution if:

\[ P(X, Y) = P(X)P(Y) \]

\[ X \perp Y \]

\[ \forall x, y \ P(x, y) = P(x)P(y) \]

- Says the joint distribution *factors* into a product of two simple ones
- Usually variables aren’t independent!

- Can use independence as a *modeling assumption*
  - Independence can be a simplifying assumption
  - *Empirical* joint distributions: at best “close” to independent
  - What could we assume for \{Weather, Traffic, Cavity\}?  

- Independence is like something from CSPs: what?
Example: Independence?

\[ P_1(T, W) \]

\[
\begin{array}{c|c|c}
T & W & P \\
\hline
\text{hot} & \text{sun} & 0.4 \\
\text{hot} & \text{rain} & 0.1 \\
\text{cold} & \text{sun} & 0.2 \\
\text{cold} & \text{rain} & 0.3 \\
\end{array}
\]

\[
\begin{array}{c|c}
T & P \\
\hline
\text{hot} & 0.5 \\
\text{cold} & 0.5 \\
\end{array}
\]

\[ P(T) \]

\[
\begin{array}{c|c}
W & P \\
\hline
\text{sun} & 0.6 \\
\text{rain} & 0.4 \\
\end{array}
\]

\[
\begin{array}{c|c|c}
T & W & P \\
\hline
\text{hot} & \text{sun} & 0.3 \\
\text{hot} & \text{rain} & 0.2 \\
\text{cold} & \text{sun} & 0.3 \\
\text{cold} & \text{rain} & 0.2 \\
\end{array}
\]

\[ P_2(T, W) = P(T)P(W) \]
Example: Independence

- N fair, independent coin flips:

\[
P(X_1) \begin{array}{|c|c|} \hline H & 0.5 \\ T & 0.5 \\ \hline \end{array} \quad P(X_2) \begin{array}{|c|c|} \hline H & 0.5 \\ T & 0.5 \\ \hline \end{array} \ldots P(X_n) \begin{array}{|c|c|} \hline H & 0.5 \\ T & 0.5 \\ \hline \end{array}
\]
Conditional Independence
Conditional Independence

- $P(\text{Toothache, Cavity, Catch})$

- If I have a cavity, the probability that the probe catches it doesn't depend on whether I have a toothache:
  - $P(+\text{catch} \mid +\text{toothache}, +\text{cavity}) = P(+\text{catch} \mid +\text{cavity})$

- The same independence holds if I don’t have a cavity:
  - $P(+\text{catch} \mid +\text{toothache}, -\text{cavity}) = P(+\text{catch} \mid -\text{cavity})$

- Catch is conditionally independent of Toothache given Cavity:
  - $P(\text{Catch} \mid \text{Toothache, Cavity}) = P(\text{Catch} \mid \text{Cavity})$

- Equivalent statements:
  - $P(\text{Toothache} \mid \text{Catch, Cavity}) = P(\text{Toothache} \mid \text{Cavity})$
  - $P(\text{Toothache, Catch} \mid \text{Cavity}) = P(\text{Toothache} \mid \text{Cavity}) P(\text{Catch} \mid \text{Cavity})$
  - One can be derived from the other easily
Conditional Independence

- Unconditional (absolute) independence very rare (why?)

- *Conditional independence* is our most basic and robust form of knowledge about uncertain environments.

- $X$ is conditionally independent of $Y$ given $Z$ if and only if:

  \[
  \forall x, y, z : P(x, y|z) = P(x|z)P(y|z)
  \]

  or, equivalently, if and only if

  \[
  \forall x, y, z : P(x|z, y) = P(x|z)
  \]
Conditional Independence

- What about this domain:
  - Traffic
  - Umbrella
  - Raining
Conditional Independence

- What about this domain:
  - Fire
  - Smoke
  - Alarm
Probability Recap

- Conditional probability
  \[ P(x|y) = \frac{P(x,y)}{P(y)} \]

- Product rule
  \[ P(x,y) = P(x|y)P(y) \]

- Chain rule
  \[ P(X_1, X_2, \ldots, X_n) = P(X_1)P(X_2|X_1)P(X_3|X_1, X_2)\ldots = \prod_{i=1}^{n} P(X_i|X_1, \ldots, X_{i-1}) \]

- X, Y independent if and only if:
  \[ \forall x, y : P(x,y) = P(x)P(y) \]

- X and Y are conditionally independent given Z if and only if:
  \[ \forall x, y, z : P(x,y|z) = P(x|z)P(y|z) \quad \text{Hence} \quad X \perp Y | Z \]
Markov Models (Markov Chains)

- A Markov model is a chain-structured Bayesian Network (BN)

- A Markov model includes:
  - Random variables $X_t$ for all time steps $t$ (the state)
  - Parameters: called transition probabilities or dynamics, specify how the state evolves over time (also, initial probs)

\[
P(X_1) \quad \text{and} \quad P(X_t|X_{t-1})
\]
Markov Models (Markov Chains)

- A Markov model defines
  - a joint probability distribution:

  \[
P(X_1, \ldots, X_n) = P(X_1) \prod_{t=2}^{N} P(X_t|X_{t-1})
  \]

- One common inference problem:
  - Compute marginals \( P(X_t) \) for all time steps \( t \).
Conditional Independence

- **Basic conditional independence:**
  - Past and future independent of the present
  - Each time step only depends on the previous
  - This is called the (first order) Markov property

- **Note that the chain is just a (growing) BN**
  - We can always use generic BN reasoning on it if we truncate the chain at a fixed length
Example: Markov Chain

- **Weather:**
  - States: \( X = \{\text{rain, sun}\} \)
  - Transitions:

- Initial distribution: 1.0 sun
- What’s the probability distribution after one step?

\[
P(X_2 = \text{sun}) = P(X_2 = \text{sun}|X_1 = \text{sun})P(X_1 = \text{sun}) + P(X_2 = \text{sun}|X_1 = \text{rain})P(X_1 = \text{rain})
\]

\[
0.9 \cdot 1.0 + 0.1 \cdot 0.0 = 0.9
\]

This is a conditional distribution
Markov Chain Inference

- **Question:** probability of being in state x at time t?
- **Slow answer:**
  - Enumerate all sequences of length t which end in s
  - Add up their probabilities

\[
P(X_t = \text{sun}) = \sum_{x_1 \ldots x_{t-1}} P(x_1, \ldots x_{t-1}, \text{sun})
\]

\[
P(X_1 = \text{sun})P(X_2 = \text{sun}|X_1 = \text{sun})P(X_3 = \text{sun}|X_2 = \text{sun})P(X_4 = \text{sun}|X_3 = \text{sun})
\]
\[
P(X_1 = \text{sun})P(X_2 = \text{rain}|X_1 = \text{sun})P(X_3 = \text{sun}|X_2 = \text{rain})P(X_4 = \text{sun}|X_3 = \text{sun})
\]
\[
\vdots
\]
Mini-Forward Algorithm

- Question: What’s $P(X)$ on some day $t$?
  - We don’t need to enumerate every sequence!

\[
P(x_t) = \sum_{x_{t-1}} P(x_t|x_{t-1})P(x_{t-1})
\]

$P(x_1) = \text{known}$

Forward simulation
Example

- From initial observation of sun

\[
\begin{align*}
P(X_1) & \quad P(X_2) \quad P(X_3) & & P(X_\infty) \\
\langle 1.0 \rangle & \quad \langle 0.9 \rangle & \quad \langle 0.82 \rangle & \quad \langle 0.5 \rangle \\
0.0 & \quad 0.1 & \quad 0.18 & \quad 0.5
\end{align*}
\]

- From initial observation of rain

\[
\begin{align*}
P(X_1) & \quad P(X_2) \quad P(X_3) & & P(X_\infty) \\
\langle 0.0 \rangle & \quad \langle 0.1 \rangle & \quad \langle 0.18 \rangle & \quad \langle 0.5 \rangle \\
1.0 & \quad 0.9 & \quad 0.82 & \quad 0.5
\end{align*}
\]
Stationary Distributions

- If we simulate the chain long enough:
  - What happens?
  - Uncertainty accumulates
  - Eventually, we have no idea what the state is!

- Stationary distributions:
  - For most chains, the distribution we end up in is independent of the initial distribution
  - Called the *stationary distribution* of the chain
  - Usually, can only predict a short time out
Video of Demo Ghostbusters
Basic Dynamics
Video of Demo Ghostbusters
Circular Dynamics
Video of Demo Ghostbusters
Whirlpool Dynamics
Pac-man Markov Chain

Pac-man knows the ghost’s initial position, but gets no observations!
Stationary Distributions

- For most chains:
  - Influence of the initial distribution gets less and less over time.
  - The distribution we end up in is independent of the initial distribution

- Stationary distribution:
  - The distribution we end up with is called the **stationary distribution** of the chain.
  - It satisfies
    \[ P_\infty(X) = P_{\infty+1}(X) = \sum_x P(X|x)P_\infty(x) \]
Example: Stationary Distributions

Question: What’s $P(X)$ at time $t = \infty$?

\[ P_\infty(\text{sun}) = P(\text{sun}|\text{sun})P_\infty(\text{sun}) + P(\text{sun}|\text{rain})P_\infty(\text{rain}) \]
\[ P_\infty(\text{rain}) = P(\text{rain}|\text{sun})P_\infty(\text{sun}) + P(\text{rain}|\text{rain})P_\infty(\text{rain}) \]

\[ P_\infty(\text{sun}) = 0.9P_\infty(\text{sun}) + 0.3P_\infty(\text{rain}) \]
\[ P_\infty(\text{rain}) = 0.1P_\infty(\text{sun}) + 0.7P_\infty(\text{rain}) \]

\[ P_\infty(\text{sun}) = 3P_\infty(\text{rain}) \]
\[ P_\infty(\text{rain}) = \frac{1}{3}P_\infty(\text{sun}) \]

Also:
\[ P_\infty(\text{sun}) + P_\infty(\text{rain}) = 1 \]

\[ P_\infty(\text{sun}) = \frac{3}{4} \]
\[ P_\infty(\text{rain}) = \frac{1}{4} \]
Web Link Analysis

- **PageRank over a web graph**
  - Each web page is a state
  - Initial distribution: uniform over pages
  - Transitions:
    - With prob. $c$, uniform jump to a random page (dotted lines, not all shown)
    - With prob. $1-c$, follow a random outlink (solid lines)

- **Stationary distribution**
  - Will spend more time on highly reachable pages
  - E.g. many ways to get to the Acrobat Reader download page
  - Somewhat robust to link spam
  - Google 1.0 returned the set of pages containing all your keywords in decreasing rank, now all search engines use link analysis along with many other factors (rank actually getting less important over time)
Hidden Markov Models

- Markov chains not so useful for most agents
  - Eventually you don’t know anything anymore
  - Need observations to update your beliefs

- Hidden Markov models (HMMs)
  - Underlying Markov chain over states $S$
  - You observe outputs (effects) at each time step
  - POMDPs without actions (or rewards).
  - As a Bayes’ net:
An HMM is defined by:

- Initial distribution: \( P(X_1) \)
- Transitions: \( P(X_t | X_{t-1}) \)
- Emissions: \( P(E | X) \)
Hidden Markov Models

- Defines a joint probability distribution:

\[
P(X_1, \ldots, X_n, E_1, \ldots, E_n) = P(X_{1:n}, E_{1:n}) = P(X_1)P(E_1|X_1) \prod_{t=2}^{N} P(X_t|X_{t-1})P(E_t|X_t)
\]
Ghostbusters HMM

- $P(X_1) = \text{uniform}$
- $P(X'|X) = \text{usually move clockwise, but sometimes move in a random direction or stay in place}$
- $P(E|X) = \text{same sensor model as before: red means close, green means far away.}$

\[
\begin{array}{ccc}
1/9 & 1/9 & 1/9 \\
1/9 & 1/9 & 1/9 \\
1/9 & 1/9 & 1/9 \\
\end{array}
\]

\[
\begin{array}{ccc}
1/6 & 1/6 & 1/2 \\
0 & 1/6 & 0 \\
0 & 0 & 0 \\
\end{array}
\]
Video of Demo Ghostbusters – Circular Dynamics -- HMM
HMM Computations

- **Given**
  - joint $P(X_{1:n}, E_{1:n})$
  - evidence $E_{1:n} = e_{1:n}$

- **Inference problems include:**
  - Filtering, find $P(X_t|e_{1:t})$ for all $t$
  - Smoothing, find $P(X_t|e_{1:n})$ for all $t$
  - Most probable explanation, find
    $$x^*_{1:n} = \arg\max_{x_{1:n}} P(x_{1:n}|e_{1:n})$$
Conditional Independence

- HMMs have two important independence properties:
  - Markov hidden process, future depends on past via the present
  - Current observation independent of all else given current state
Real HMM Examples

- **Speech recognition HMMs:**
  - Observations are acoustic signals (continuous valued)
  - States are specific positions in specific words (so, tens of thousands)

- **Machine translation HMMs:**
  - Observations are words (tens of thousands)
  - States are translation options

- **Robot tracking:**
  - Observations are range readings (continuous)
  - States are positions on a map (continuous)
Filtering / Monitoring

- Filtering, or monitoring, is the task of tracking the distribution $B(X)$ (the belief state) over time.
- We start with $B(X)$ in an initial setting, usually uniform.
- As time passes, or we get observations, we update $B(X)$.
- The Kalman filter was invented in the 60’s and first implemented as a method of trajectory estimation for the Apollo program.
Example: Robot Localization

Sensor model: never more than 1 mistake
Motion model: may not execute action with small prob.
Example: Robot Localization

Prob

0

1

t=1
Example: Robot Localization

t=2
Example: Robot Localization

Prob
0 1

t=3
Example: Robot Localization

t=4
Example: Robot Localization

$\text{Prob} \begin{array}{c} 0 \\ 1 \end{array}$

$t=5$
Inference Recap: Simple Cases

\[ P(X_1|e_1) \]

\[ P(x_1|e_1) = \frac{P(x_1, e_1)}{P(e_1)} \]

\[ \propto_{X_1} P(x_1, e_1) \]

\[ = P(x_1)P(e_1|x_1) \]

\[ P(x_2) = \sum_{x_1} P(x_1, x_2) \]

\[ = \sum_{x_1} P(x_1)P(x_2|x_1) \]
Online Belief Updates

- Every time step, we start with current $P(X \mid \text{evidence})$
- We update for time:
  
  $$P(x_t \mid e_{1:t-1}) = \sum_{x_{t-1}} P(x_{t-1} \mid e_{1:t-1}) \cdot P(x_t \mid x_{t-1})$$

- We update for evidence:
  
  $$P(x_t \mid e_{1:t}) \propto P(x_t \mid e_{1:t-1}) \cdot P(e_t \mid x_t)$$

- The forward algorithm does both at once (and doesn’t normalize)
- Problem: space is $|X|$ and time is $|X|^2$ per time step
Passage of Time

- Assume we have current belief $P(X | \text{evidence to date})$
  \[ B(X_t) = P(X_t | e_{1:t}) \]

- Then, after one time step passes:
  \[ P(X_{t+1} | e_{1:t}) = \sum_{x_t} P(X_{t+1} | x_t) P(x_t | e_{1:t}) \]

- Or, compactly:
  \[ B'(X') = \sum_x P(X' | x) B(x) \]

- Basic idea: beliefs get “pushed” through the transitions
  - With the “B” notation, we have to be careful about what time step $t$ the belief is about, and what evidence it includes
Example: Passage of Time

- As time passes, uncertainty “accumulates”

\[ B'(X') = \sum_{x} P(X'|x)B(x) \]

Transition model: ghosts usually go clockwise
Assume we have current belief \( P(X \mid \text{previous evidence}) \):

\[
B'(X_{t+1}) = P(X_{t+1} \mid e_{1:t})
\]

Then:

\[
P(X_{t+1} \mid e_{1:t+1}) \propto P(e_{t+1} \mid X_{t+1}) P(X_{t+1} \mid e_{1:t})
\]

Or:

\[
B(X_{t+1}) \propto P(e \mid X) B'(X_{t+1})
\]

Basic idea: beliefs reweighted by likelihood of evidence

Unlike passage of time, we have to renormalize
Example: Observation

- As we get observations, beliefs get reweighted, uncertainty “decreases”

\[ B(X) \propto P(e|X)B'(X) \]
The Forward Algorithm

- We need to know: \( B_t(X) = P(X_t | e_{1:t}) \)
- We can derive the following updates

\[
P(x_t | e_{1:t}) \propto_X P(x_t, e_{1:t})
\]

\[
= \sum_{x_{t-1}} P(x_{t-1}, x_t, e_{1:t})
\]

\[
= \sum_{x_{t-1}} P(x_{t-1}, e_{1:t-1})P(x_t | x_{t-1})P(e_t | x_t)
\]

\[
= P(e_t | x_t) \sum_{x_{t-1}} P(x_t | x_{t-1})P(x_{t-1}, e_{1:t-1})
\]

- To get \( B_t(X) \) compute each entry and normalize
An HMM is defined by:
- Initial distribution: $P(X_1)$
- Transitions: $P(X_t|X_{t-1})$
- Emissions: $P(E|X)$
Example HMM

True
False
0.500
0.500
0.500
0.182
0.818
0.500
0.883
0.373
0.627

Rain_0 → Rain_1 → Rain_2

Umbrella_1
Umbrella_2
Example Pac-man
Summary: Filtering

- Filtering is the inference process of finding a distribution over $X_T$ given $e_1$ through $e_T$: $P(X_T | e_{1:t})$
- We first compute $P(X_1 | e_1)$: $P(x_1 | e_1) \propto P(x_1) \cdot P(e_1 | x_1)$
- For each $t$ from 2 to $T$, we have $P(X_{t-1} | e_{1:t-1})$
- Elapse time: compute $P(X_t | e_{1:t-1})$

$$P(x_t | e_{1:t-1}) = \sum_{x_{t-1}} P(x_{t-1} | e_{1:t-1}) \cdot P(x_t | x_{t-1})$$

- Observe: compute $P(X_t | e_{1:t-1}, e_t) = P(X_t | e_{1:t})$

$$P(x_t | e_{1:t}) \propto P(x_t | e_{1:t-1}) \cdot P(e_t | x_t)$$
Recap: Reasoning Over Time

- **Stationary Markov models**

  \[ P(X_1) \quad P(X|X_{-1}) \]

- **Hidden Markov models**

  

  \begin{tabular}{ccc}
  X & E & P \\
  rain & umbrella & 0.9 \\
  rain & no umbrella & 0.1 \\
  sun & umbrella & 0.2 \\
  sun & no umbrella & 0.8 \\
  \end{tabular}
Recap: Filtering

Elapse time: compute $P( X_t | e_{1:t-1} )$

$P(x_t|e_{1:t-1}) = \sum_{x_{t-1}} P(x_{t-1}|e_{1:t-1}) \cdot P(x_t|x_{t-1})$

Observe: compute $P( X_t | e_{1:t} )$

$P(x_t|e_{1:t}) \propto P(x_t|e_{1:t-1}) \cdot P(e_t|x_t)$

Belief: $\langle P(\text{rain}), P(\text{sun}) \rangle$

- $P(X_1) <0.5, 0.5>$ Prior on $X_1$
- $P(X_1 | E_1 = \text{umbrella}) <0.82, 0.18>$ Observe
- $P(X_2 | E_1 = \text{umbrella}) <0.63, 0.37>$ Elapse time
- $P(X_2 | E_1 = \text{umb}, E_2 = \text{umb}) <0.88, 0.12>$ Observe
Particle Filtering

- Sometimes $|X|$ is too big to use exact inference
  - $|X|$ may be too big to even store $B(X)$
  - E.g. $X$ is continuous
  - $|X|^2$ may be too big to do updates

- Solution: approximate inference
  - Track samples of $X$, not all values
  - Samples are called particles
  - Time per step is linear in the number of samples
  - But: number needed may be large
  - In memory: list of particles, not states

- This is how robot localization works in practice
Our representation of $P(X)$ is now a list of $N$ particles (samples)

- Generally, $N \ll |X|$
- Storing map from $X$ to counts would defeat the point

$P(x)$ approximated by number of particles with value $x$

- So, many $x$ will have $P(x) = 0!$
- More particles, more accuracy

For now, all particles have a weight of 1

Particles:
- (3,3)
- (2,3)
- (3,3)
- (3,2)
- (3,3)
- (3,2)
- (2,1)
- (3,3)
- (3,3)
- (2,1)
Particle Filtering: Elapse Time

- Each particle is moved by sampling its next position from the transition model
  \[ x' = \text{sample}(P(X'|x)) \]

- This is like prior sampling – samples’ frequencies reflect the transition probs

- Here, most samples move clockwise, but some move in another direction or stay in place

- This captures the passage of time
  - If we have enough samples, close to the exact values before and after (consistent)
Particle Filtering: Observe

- Slightly trickier:
  - Don’t do rejection sampling (why not?)
  - We don’t sample the observation, we fix it
  - This is similar to likelihood weighting, so we downweight our samples based on the evidence

\[ w(x) = P(e|x) \]

\[ B(X) \propto P(e|X)B'(X) \]

- Note that, as before, the probabilities don’t sum to one, since most have been downweighted (in fact they sum to an approximation of \( P(e) \))
Particle Filtering: Resample

- Rather than tracking weighted samples, we resample.

- N times, we choose from our weighted sample distribution (i.e. draw with replacement).

- This is equivalent to renormalizing the distribution.

- Now the update is complete for this time step, continue with the next one.

Old Particles:
- (3,3) w=0.1
- (2,1) w=0.9
- (2,1) w=0.9
- (3,1) w=0.4
- (3,2) w=0.3
- (2,2) w=0.4
- (1,1) w=0.4
- (3,1) w=0.4
- (2,1) w=0.9
- (3,2) w=0.3

New Particles:
- (2,1) w=1
- (2,1) w=1
- (2,1) w=1
- (3,2) w=1
- (2,2) w=1
- (2,1) w=1
- (2,1) w=1
- (1,1) w=1
- (3,1) w=1
- (2,1) w=1
- (1,1) w=1
Recap: Particle Filtering

At each time step $t$, we have a set of $N$ particles / samples

- Initialization: Sample from prior, reweight and resample
- Three step procedure, to move to time $t+1$:
  1. **Sample transitions**: for each particle $x$, sample next state
     \[
     x' = \text{sample}(P(X'|x))
     \]
  2. **Reweight**: for each particle, compute its weight given the actual observation $e$
     \[
     w(x) = P(e|x)
     \]
  - **Resample**: normalize the weights, and sample $N$ new particles from the resulting distribution over states
Particle Filtering Summary

- Represent current belief $P(X \mid \text{evidence to date})$ as set of $n$ samples (actual assignments $X=x$)
- For each new observation $e$:
  1. Sample transition, once for each current particle $x$
     \[ x' = \text{sample}(P(X' \mid x)) \]
  2. For each new sample $x'$, compute importance weights for the new evidence $e$:
     \[ w(x') = P(e \mid x') \]
  3. Finally, normalize the importance weights and resample $N$ new particles
Recap: Particle Filtering

- **Particles**: track samples of states rather than an explicit distribution

![Diagram of Particle Filtering Process]

- Elapse
- Weight
- Resample

<table>
<thead>
<tr>
<th>Particles:</th>
<th>Particles:</th>
<th>Particles:</th>
<th>(New) Particles:</th>
</tr>
</thead>
<tbody>
<tr>
<td>(3,3)</td>
<td>(3,2)</td>
<td>(3,2)</td>
<td>(3,2)</td>
</tr>
<tr>
<td>(2,3)</td>
<td>(2,3)</td>
<td>(2,3)</td>
<td>(2,2)</td>
</tr>
<tr>
<td>(3,3)</td>
<td>(3,2)</td>
<td>(3,1)</td>
<td>(3,2)</td>
</tr>
<tr>
<td>(3,2)</td>
<td>(3,3)</td>
<td>(3,3)</td>
<td>(3,3)</td>
</tr>
<tr>
<td>(3,2)</td>
<td>(3,2)</td>
<td>(3,2)</td>
<td>(3,2)</td>
</tr>
<tr>
<td>(1,2)</td>
<td>(1,3)</td>
<td>(1,3)</td>
<td>(1,3)</td>
</tr>
<tr>
<td>(3,3)</td>
<td>(2,3)</td>
<td>(2,3)</td>
<td>(2,3)</td>
</tr>
<tr>
<td>(3,3)</td>
<td>(3,2)</td>
<td>(3,2)</td>
<td>(3,2)</td>
</tr>
<tr>
<td>(2,3)</td>
<td>(2,2)</td>
<td>(2,2)</td>
<td>(3,2)</td>
</tr>
</tbody>
</table>
Video of Demo – Moderate Number of Particles
Video of Demo – One Particle
Video of Demo – Huge Number of Particles
Robot Localization

- In robot localization:
  - We know the map, but not the robot's position
  - Observations may be vectors of range finder readings
  - State space and readings are typically continuous (works basically like a very fine grid) and so we cannot store $B(X)$
  - Particle filtering is a main technique
Robot Localization

QuickTime™ and a GIF decompressor are needed to see this picture.
Which Algorithm?

Exact filter, uniform initial beliefs
Which Algorithm?

Particle filter, uniform initial beliefs, 300 particles
Which Algorithm?

Particle filter, uniform initial beliefs, 25 particles
### P4: Ghostbusters

- **Plot:** Pacman's grandfather, Grandpac, learned to hunt ghosts for sport.

- He was blinded by his power, but could hear the ghosts’ banging and clanging.

- **Transition Model:** All ghosts move randomly, but are sometimes biased.

- **Emission Model:** Pacman knows a “noisy” distance to each ghost.
Dynamic Bayes Nets (DBNs)

- We want to track multiple variables over time, using multiple sources of evidence.
- Idea: Repeat a fixed Bayes net structure at each time.
- Variables from time $t$ can condition on those from $t-1$.

Discrete valued dynamic Bayes nets are also HMMs.
Exact Inference in DBNs

- Variable elimination applies to dynamic Bayes nets
- Procedure: “unroll” the network for T time steps, then eliminate variables until $P(X_T|e_{1:T})$ is computed

Online belief updates: Eliminate all variables from the previous time step; store factors for current time only.
DBN Particle Filters

- A particle is a complete sample for a time step
- **Initialize**: Generate prior samples for the t=1 Bayes net
  - Example particle: $G_1^a = (3,3) \quad G_1^b = (5,3)$

- **Elapse time**: Sample a successor for each particle
  - Example successor: $G_2^a = (2,3) \quad G_2^b = (6,3)$

- **Observe**: Weight each entire sample by the likelihood of the evidence conditioned on the sample
  - Likelihood: $P(E_1^a | G_1^a) \times P(E_1^b | G_1^b)$

- **Resample**: Select prior samples (tuples of values) in proportion to their likelihood
SLAM

- **SLAM** = Simultaneous Localization And Mapping
  - We do not know the map or our location
  - Our belief state is over maps and positions!
  - Main techniques: Kalman filtering (Gaussian HMMs) and particle methods

- [DEMOS]

DP-SLAM, Ron Parr
Best Explanation Queries

- Query: most likely seq:

\[
\arg \max_{x_{1:t}} P(x_{1:t} | e_{1:t})
\]
State Path Trellis

- State trellis: graph of states and transitions over time

- Each arc represents some transition \( x_{t-1} \rightarrow x_t \)

- Each arc has weight \( P(x_t|x_{t-1})P(e_t|x_t) \)

- Each path is a sequence of states

- The product of weights on a path is the seq’s probability

- Can think of the Forward (and now Viterbi) algorithms as computing sums of all paths (best paths) in this graph
Viterbi Algorithm

\[ x_{1:T}^* = \arg \max_{x_{1:T}} P(x_{1:T} | e_{1:T}) = \arg \max_{x_{1:T}} P(x_{1:T}, e_{1:T}) \]

\[ m_t[x_t] = \max_{x_{1:t-1}} P(x_{1:t-1}, x_t, e_{1:t}) \]

\[ = \max_{x_{1:t-1}} P(x_{1:t-1}, e_{1:t-1}) P(x_t | x_{t-1}) P(e_t | x_t) \]

\[ = P(e_t | x_t) \max_{x_{t-1}} P(x_t | x_{t-1}) \max_{x_{1:t-2}} P(x_{1:t-1}, e_{1:t-1}) \]

\[ = P(e_t | x_t) \max_{x_{t-1}} P(x_t | x_{t-1}) m_{t-1}[x_{t-1}] \]
Example