asst 12 might be posted!
Particle Filters
supervised learning: learning a function or a density
unsupervised learning: explaining data
filtering: estimating state, particularly under change
Bayesian Belief Update

type A coins have \( P(\text{heads}) = 0.5 \)
type B coins have \( P(\text{heads}) = 0.6 \)
type C coins have \( P(\text{heads}) = 0.9 \)

A drawer contains two A's and one B and one C. You reach into the drawer and randomly pick a coin. What is the probability that the coin is each type?

You flip the coin and get heads. Now what is the probability that the coin is each type?

You flip the coin again and get heads again. Now what is the probability that the coin is each type?
Updating for Localization

\[ P(s|o) = \frac{P(o|s)P(s)}{P(o)} \]
Updating for Localization

\[ P(s|o) = \frac{P(o|s)P(s)}{P(o)} \]

\[ P(s'|s, u, o') = \frac{P(o'|s, u, s')P(s, u, s')}{P(s, u, o')} \]
Updating for Localization

\[ P(s | o) = \frac{P(o | s)P(s)}{P(o)} \]

\[ P(s' | s, u, o') = \frac{P(o' | s, u, s')P(s, u, s')}{P(s, u, o')} \]

\[ P(s' | s, u, o') = \alpha P(o' | s')P(s' | s, u) \]
Monte Carlo Localization

\[ S \leftarrow \text{samples from prior} \]
\[ w \leftarrow \text{uniform distribution} \]

repeat forever:

for each sample \( s_i \) and weight \( w_i \),

\[ s_i \leftarrow \text{sample from } P(S'_i|s_i, u) \]
\[ w_i \leftarrow P(o|s_i) \]
\[ S \leftarrow \text{sample from } S \text{ with } P(s_i) \propto w_i \]

+: nonparametric, scalable computation and accuracy, simple

-: kidnapping, high D
Break

- asst 11
- asst 12
- Tue May 5 9-noon: project presentations
- Mon May 11 2pm: final papers
Hidden Markov Models
Probabilistic Models

Naive Bayes:

$k$-Means:

Markov chain:

MDPs:

Hidden Markov model:
\[
\begin{align*}
P(x_t = j) &= \sum_i P(x_{t-1} = i) P(x_t = j | x_{t-1} = i) \\
P(o_t = k) &= \sum_i P(x_t = i) P(o = k | x = i)
\end{align*}
\]
The Model

Particle Filters

HMMs

- Models
- The Model

Viterbi Decoding

\[
P(x_t = j) = \sum_{i} P(x_{t-1} = i)P(x_t = j | x_{t-1} = i) \\
P(o_t = k) = \sum_{i} P(x_t = i)P(o = k | x = i)
\]

More concisely:

\[
P(x_t) = \sum_{x_{t-1}} P(x_{t-1})P(x_t | x_{t-1}) \\
P(o_t) = \sum_{x_t} P(x_t)P(o | x)
\]
Viterbi Decoding
probability of a sequence multiplies forward in time
dynamic programming backward through time
The Algorithm

given: transition model $T(s, s')$
sensing model $S(s, o)$
observations $o_1, \ldots, o_T$

find: most probable $s_1, \ldots, s_T$

initialize $S \times T$ matrix $v$ with 0s
$v_{0,0} \leftarrow 1$
for each time $t = 0$ to $T - 1$
  for each state $s$
    for each new state $s'$
      score $\leftarrow v_{s,t} \cdot T(s, s') \cdot S(s', o_t)$
      if score $>$ $v_{s', t+1}$
        $v_{s', t+1} \leftarrow$ score
        best-parent($s'$) $\leftarrow$ $s$
  trace back from $s$ with max $v_{s,T}$
EOLQs

What question didn’t you get to ask today?
What’s still confusing?
What would you like to hear more about?

Please write down your most pressing question about AI and put it in the box on your way out.

Thanks!