CS 188: Artificial Intelligence
Spring 2010

Lecture 13: Probability
3/2/2010

Pieter Abbeel – UC Berkeley
Many slides adapted from Dan Klein.

Announcements
- Upcoming
  - “new” Tomorrow/Wednesday: probability review session
  - 7:30-9:30pm in 306 Soda
  - P3 due on Thursday (3/4)
  - W4 going out on Thursday, due next week Thursday (3/11)
  - Midterm in evening of 3/18

Today
- We’re almost done with search and planning!
  - MDP’s: policy search wrap-up
- Next, we’ll start studying how to reason with probabilities
  - Diagnosis
  - Tracking objects
  - Speech recognition
  - Robot mapping
  - … lots more!
- Third part of course: machine learning

MDPs recap
- MDP recap: \((S, A, T, R, s_0, \gamma)\)
  - In small MDPs: can find \(V(s)\) and/or \(Q(s,a)\)
    - Known \(T, R\): value iteration, policy iteration
    - Unknown \(T, R\): Q learning
  - In large MDPs: cannot enumerate all states

Policy Search

Function Approximation
- \(Q(s, a) = w_1 f_1(s, a) + w_2 f_2(s, a) + \ldots + w_n f_n(s, a)\)
- Q-learning with linear q-functions:
  - \(\text{transition} = (s, a, r, s')\)
  - \(\text{difference} = (r + \max_{a'} Q(s', a') - Q(s, a))\)
  - \(Q(s, a) = Q(s, a) + \alpha \text{[difference]}\)
  - \(w_i = w_i + \alpha \text{[difference]} f_i(s, a)\)
- Intuitive interpretation:
  - Adjust weights of active features
    - E.g. if something unexpectedly bad happens, disprefer all states with that state’s features
- Formal justification: online least squares
Policy Search Idea

- Problem: often the feature-based policies that work well aren’t the ones that approximate V / Q best
- Solution: learn the policy that maximizes rewards rather than the value that predicts rewards
- This is the idea behind policy search, such as what controlled the upside-down helicopter

Policy Search

- Simplest policy search: Start with an initial linear value function or Q-function
- Nudge each feature weight up and down and see if your policy is better than before

- Problems:
  - How do we tell the policy got better?
  - Need to run many sample episodes
  - If there are a lot of features, this can be impractical
    - Mostly applicable when prior knowledge allows one to choose a representation with a very small number of free parameters to be learned

Toddler (Tedrake et al.)

Take a Deep Breath…

- We’re done with search and planning!
- Next, we’ll look at how to reason with probabilities
  - Diagnosis
  - Tracking objects
  - Speech recognition
  - Robot mapping
  - … lots more!
- Third part of course: machine learning

Today

- Probability
  - Random Variables
  - Joint and Marginal Distributions
  - Conditional Distribution
  - Product Rule, Chain Rule, Bayes’ Rule
  - Inference
  - Independence
- You’ll need all this stuff A LOT for the next few weeks, so make sure you go over it now!
- Probability review session tomorrow 7:30-9:30pm in 306 Soda --- you will benefit from it for many lectures/assignments/exam questions if any of the material we are about to go over today is not completely trivial!
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 \( P(\text{Color} | \text{Distance}) \)

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

Uncertainty

- General situation:
  - Evidence: Agent knows certain things about the state of the world (e.g., sensor readings or symptoms)
  - Hidden variables: Agent needs to reason about other aspects (e.g., where an object is or what disease is present)
  - Model: Agent knows something about how the known variables relate to the unknown variables
- Probabilistic reasoning gives us a framework for managing our beliefs and knowledge

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
- Like variables in a CSP, random variables have domains
  - \( R \) in \{true, false\} (sometimes write as \( \{+r, -r\} \))
  - \( D \) in \([0, \infty)\)
  - \( L \) in possible locations, maybe \((0,0), (0,1), \ldots\)

Probability Distributions

- Unobserved random variables have distributions
  - A distribution is a TABLE of probabilities of values
  - A probability (lower case value) is a single number
- A joint distribution over a set of random variables:
  - \( P(X_1, X_2, \ldots, X_n) \)
  - Size of distribution if \( n \) variables with domain sizes \( d \)?
  - Must obey:
    \[
    \sum_{(x_1, x_2, \ldots, x_n)} P(x_1, x_2, \ldots, x_n) = 1
    \]
  - For all but the smallest distributions, impractical to write out

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)
  - \( P(X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n) \)
  - Size of distribution if \( n \) variables with domain sizes?
  - Must obey:
    \[
    \sum_{(x_1, x_2, \ldots, x_n)} P(x_1, x_2, \ldots, x_n) = 1
    \]
  - For all but the smallest distributions, impractical to write out

Probabilistic Models

- A probabilistic model is a joint distribution over a set of random variables
- Probabilistic models:
  - (Random) variables with domains
  - Assignments are called outcomes
  - Joint distributions: say whether assignments (outcomes) are likely
  - Normalized: sum to 1.0
  - Ideally: only certain variables directly interact
- Constraint satisfaction problems:
  - Variables with domains
  - Constraints: state whether assignments are possible
  - Ideally: only certain variables directly interact

Distributions over \( T, W \)

<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>0.6</td>
</tr>
<tr>
<td>cold</td>
<td>rain</td>
<td>0.3</td>
</tr>
</tbody>
</table>

Constraints over \( T, W \)

<table>
<thead>
<tr>
<th>( T )</th>
<th>( W )</th>
<th>( P )</th>
</tr>
</thead>
<tbody>
<tr>
<td>fog</td>
<td>cold</td>
<td>F</td>
</tr>
<tr>
<td>hot</td>
<td>rain</td>
<td>F</td>
</tr>
<tr>
<td>cold</td>
<td>sun</td>
<td>F</td>
</tr>
<tr>
<td>cold</td>
<td>fog</td>
<td>F</td>
</tr>
</tbody>
</table>
Events

- 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:
  \[ P(\text{hot and sunny}) = 0.5 \]
  \[ P(\text{hot or sunny}) = 0.7 \]
- Typically, the events we care about are partial assignments, like $P(T=\text{hot})$

Conditional Probabilities

- A simple relation between joint and conditional probabilities
  \[ P(a|b) = \frac{P(a, b)}{P(b)} \]

Conditional Distributions

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

Marginal Distributions

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

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)