Lecture 12, 13 & 14 Overview

• Finishing off reinforcement learning
• Bayesian probability
• Reasoning under uncertainty (Chapter 14)
  – Belief networks (reasoning only) (this lecture)
  – Exact inference
  – Next Lecture
    • Exact inference is NP-hard
    • Approximate inference using Gibbs sampler
  – Term project next Monday
• Later lectures - reasoning in the presence of no uncertainty (propositional logic)
Learning $\hat{Q}$ - Deterministic Worlds

For each $s, a$ initialize table entry $Q(s, a) \leftarrow 0$

Observe current state $s$

Do forever:

- Select an action $a$ and execute it
- Receive immediate reward $r$
- Observe the new state $s'$
- Update the table entry for $\hat{Q}(s, a)$ as follows:
  $$\hat{Q}(s, a) \leftarrow r + \gamma \max_{a'} \hat{Q}(s', a')$$
- $s \leftarrow s'$
Exploitation versus Exploration

• An example (tic-tac)
• The need for exploration
• Factoring in exploration
• Proof of convergence
  – “Problems” with Q-learning
    • Need to stumble across a reward (somehow)
    • Proof of convergence is in the limit
    • Limit is over unusual requirement ...
Convergence: \( \lim_{n \to \infty} \hat{Q} = Q \)

**Proof:** Define a full interval to be an interval during which each \((s, a)\) is visited. During each full interval the largest error in \(\hat{Q}\) table is reduced by factor of \(\gamma\).

Let \(\hat{Q}_n\) be table after \(n\) updates, and \(\Delta_n\) be the maximum error in \(\hat{Q}_n\); that is

\[
\Delta_n = \max_{s, a} |\hat{Q}_n(s, a) - Q(s, a)|
\]

For any table entry \(\hat{Q}_n(s, a)\) updated on iteration \(n + 1\), the error in the revised estimate \(\hat{Q}_{n+1}(s, a)\) is

\[
|\hat{Q}_{n+1}(s, a) - Q(s, a)| = |(r + \gamma \max_{a'} \hat{Q}_n(s', a'))
- (r + \gamma \max_{a'} Q(s', a'))| \\
= \gamma |\max_{a'} \hat{Q}_n(s', a') - \max_{a'} Q(s', a')| \\
\leq \gamma \max_{a'} |\hat{Q}_n(s', a') - Q(s', a')| \\
\leq \gamma \max_{s'', a'} |\hat{Q}_n(s'', a') - Q(s'', a')| \\
|\hat{Q}_{n+1}(s, a) - Q(s, a)| \leq \gamma \Delta_n
\]
Non-Deterministic Worlds

What if reward and next state are non-deterministic?

We redefine $V, Q$ by taking expected values

$$V^\pi(s) \equiv E[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \ldots]$$
$$\equiv E[\sum_{i=0}^{\infty} \gamma^i r_{t+i}]$$

$$Q(s, a) \equiv E[r(s, a) + \gamma V^*(\delta(s, a))]$$
Learning $\hat{Q}$ – Non-Deterministic Worlds

$Q$ learning generalizes to nondeterministic worlds

Alter training rule to

$$\hat{Q}_n(s, a) \leftarrow (1-\alpha_n)\hat{Q}_{n-1}(s, a) + \alpha_n[r + \max_{a'}\hat{Q}_{n-1}(s', a')]$$

where

$$\alpha_n = \frac{1}{1 + visits_n(s, a)}$$

Can still prove convergence of $\hat{Q}$ to $Q$ [Watkins and Dayan, 1992]
Absorbing State

$$\gamma = 0.9$$

Grid World

$$r(s,a)$$ (immediate reward) values

$$Q(s,a)$$ values

$$V^*(s)$$ values

One optimal policy
Temporal Difference Learning

Learn $Q$ Quickly

$Q$ learning: reduce discrepancy between successive $Q$ estimates

One step time difference:

$$Q^{(1)}(s_t, a_t) \equiv r_t + \gamma \max_a \hat{Q}(s_{t+1}, a)$$

Why not two steps?

$$Q^{(2)}(s_t, a_t) \equiv r_t + \gamma r_{t+1} + \gamma^2 \max_a \hat{Q}(s_{t+2}, a)$$

Or $n$?

$$Q^{(n)}(s_t, a_t) \equiv r_t + \gamma r_{t+1} + \cdots + \gamma^{(n-1)} r_{t+n-1} + \gamma^n \max_a \hat{Q}(s_{t+n}, a)$$

Blend all of these:

$$Q^\lambda(s_t, a_t) \equiv (1-\lambda) \left[ Q^{(1)}(s_t, a_t) + \lambda Q^{(2)}(s_t, a_t) + \lambda^2 Q^{(3)}(s_t, a_t) \right].$$
Temporal Difference Learning

\[ Q^\lambda(s_t, a_t) \equiv (1 - \lambda) \left[ Q^{(1)}(s_t, a_t) + \lambda Q^{(2)}(s_t, a_t) + \lambda^2 Q^{(3)}(s_t, a_t) \right] \]

Equivalent expression:

\[ Q^\lambda(s_t, a_t) = r_t + \gamma \left[ (1 - \lambda) \max_a \hat{Q}(s_t, a_t) \right. \]
\[ + \lambda \left. Q^\lambda(s_{t+1}, a_{t+1}) \right] \]

TD(\lambda) algorithm uses above training rule

- Sometimes converges faster than Q learning
- Converges for learning V* for any 0 \leq \lambda \leq 1 (Dayan, 1992)
- Tesauro’s TD-Gammon uses this algorithm
Primer on Probability

\[ P(\text{Event}) = q \]

Various Interpretations of \( q \)

- Frequentist
- Degree of belief
Probability - 1

- Distributions
- Random variables
  - Discrete
    - Sum rule
  - Continuous
- Background state of information
Probability - II

- Discrete Random Variables
- Continuous Random Variables

Probability distribution (density function) over continuous values

\[ X \in [0,10] \quad P(x) \geq 0 \]

\[
\int_{0}^{10} P(x) \, dx = 1
\]

\[ P(5 \leq x \leq 7) = \int_{5}^{7} P(x) \, dx \]
Probability - III

• Conditional Probabilities

• Joint Probabilities

• Product Rule

• Marginalization
Bayes Theorem

\[ P(h, D) = P(D|h) \cdot P(h) = P(D|h) \cdot P(h) \]

\[ P(h|D) = \frac{P(D|h) \cdot P(h)}{P(D)} \]

- \( P(h) \) = prior probability of hypothesis \( h \)
- \( P(D) \) = prior probability of training data \( D \)
- \( P(h|D) \) = probability of \( h \) given \( D \)
- \( P(D|h) \) = probability of \( D \) given \( h \)
About the Hypothesis Space $P(h)$

- Priors
  - Each $h_i$ should be *Mutually exclusive*
  - Together the hypotheses must be *Totally exhaustive*
  - $\Sigma P(h_i)=1$
  - Priors encode knowledge before we see the data
About the Data $P(D)$ and $P(D|H)$

- **Data, $P(D)$**
  - Data is considered to be a sample of all available data.
  - $P(D)$, probability the data will be observed given no knowledge of the hypothesis.
  - Constant for fixed data and if comparing hypotheses, can be ignored

- **Likelihood, $P(D|h)$**
  - Probability a hypothesis generated the observed data or probability of observing data given the hypothesis is true.
  - If the $n$ instances are independent then
    - $P(D|h) = P(D_1|h) \cdot P(D_2|h) \cdots P(D_n|h)$
  - Often use the Loglikelihood ($P(D|h)$).
Bayesian Posterior

- $P(h|D)$ is the posterior probability of the hypothesis (given the data).
- Usual aim of Bayesian learning is to find the MAP estimate
  - Most probable model in the model space
  - May be many highly probable models
A Simple Example

Does patient have cancer or not?

A patient takes a lab test and the result comes back positive. The test returns a correct positive result in only 98% of the cases in which the disease is actually present, and a correct negative result in only 97% of the cases in which the disease is not present. Furthermore, .008 of the entire population have this cancer.

\[
\begin{align*}
P(\text{cancer}) &=  \\
P(+|\text{cancer}) &=  \\
P(+|\neg\text{cancer}) &= \\

P(\neg\text{cancer}) &=  \\
P(-|\text{cancer}) &=  \\
P(-|\neg\text{cancer}) &= 
\end{align*}
\]
Basic Rules of Probability

- **Product Rule**: probability $P(A \land B)$ of a conjunction of two events A and B:
  \[
  P(A \land B) = P(A|B)P(B) = P(B|A)P(A)
  \]

- **Sum Rule**: probability of a disjunction of two events A and B:
  \[
  P(A \lor B) = P(A) + P(B) - P(A \land B)
  \]

- **Theorem of total probability**: if events $A_1, \ldots, A_n$ are mutually exclusive with $\sum_{i=1}^{n} P(A_i) = 1$, then
  \[
  P(B) = \sum_{i=1}^{n} P(B|A_i)P(A_i)
  \]
Bayesian Belief Networks

• Combination of probabilistic modeling and DAGs
• Nodes on graph are propositional variables.
• Links represent apriori known causal dependencies.
• Reasoning by merging semantic models and evidence.
• Efficient representation of joint distribution
Direct World Representations

- Can compute any subset of propositions given another subset.
- Perform different types of reasoning
  - Prediction
  - Abduction
  - Explaining away
- Global semantics
- Local semantics exploit conditional independence
Reasoning with a Bayesian Net

• Reasoning without evidence
• Reasoning with evidence
• Bayesian network reasoning NP-Hard
  – Instance of propositional logic satisfiability problem
• Use Monte Carlo techniques to simulate draws from the joint distribution
Causation and Cognition

• Causal networks
• Causal discovery
• Models of Cognition
  – Propositional models of reasoning with uncertainty
  – Local representations, partial information, distributed parallel processing, inference and reasoning, prediction, abduction, reasoning away
  – Network structure exists in our brain?
  – Human reasoning must be more than propositional reasoning!
  – Dynamic modification of networks