1 Markov Decision Processes and Bellman Equations

Review 4x3 Grid World
MDPs consist of

- A set of States $s \in S$
- A set of Actions $a \in A$
- A Transition Function $T(s, a, s')$
- A Reward Function $R(s, a, s')$
- A start state (optional)
- A terminal state (optional)

Markov decision processes attempt to find the optimal policy $\pi * (s)$ which is the best decision at any given state (s) at any particular iteration k.
We can do this by starting with a vector of all states initialized to 0, called V_0. Next utilize V_{k-1} to calculate V_k with the **Bellman Equation**:

$$max \Sigma T(s, a, s')(R(s, a, s') + \gamma V_{k-1}(s'))$$

Where γ is the discount (given). Remember, even though Dr. J says that this exponentially increases as k increases, this increase is built in. Just multiple $V_{k-1}(s')$ by γ

1.1 Policy Evaluation

The MDP where we attempt to create the best action $\pi*$ for any state s

1.2 Policy Extraction

Perform a one step expecti-max to calculate

$$argmax \Sigma T(s, a, s')(R(s, a, s') + \gamma V_{k-1}(s'))$$

2 Intro to Machine Learning

2.1 1R

Attempts to characterize and predict a model based off of a single attribute. 1R is not capable of evaluating numerical data.
Remember the weather.nominal.arff, specifically looking at Outlook. For 14 datasets, we can analyze each class within Outlook: Sunny, Overcast, Rainy.

- Sunny, 2 Yes, 3 no. The majority of this group is no's with an error of 2/5
- Overcast, 4 Yes, 0 no. The entirety of this group is yes with an error of 0/4.
- Rainy, 3 yes, 2 no. The majority of this group is yes with an error of 2/5.

So for Outlook, we have an error of 4/14. We repeat this process for all other attributes within the dataset, and choose the attribute with the smallest error. For arguments sake, we'll assume Outlook is the smallest error. Thusly, we'll choose the prediction of instances based off of their Outlook status. If a instance is Sunny, we will say don't play. If an instance is Overcast, we'll say yes play. If an instance is Rainy, we'll say yes play.

2.2 Naïve Bayes

In contrast to 1R, **Naïve Bayes** takes into consideration all attribute within an instance.

- Probability Theory (Bayes, 1763)
  - Given the "new" instance

Outlook	Temperature	Humidity	Windy	Play
Sunny	cool	high	true	?

  - What is the predicted class?
    - Likelihood of *Yes* = $^2/_9 \times {}^3/_9 \times {}^3/_9 \times {}^3/_9 \times {}^9/_{14} = .0053$
    - Likelihood of *No* = $^3/_5 \times {}^1/_5 \times {}^4/_5 \times {}^3/_5 \times {}^5/_{14} = .0206$
    - So *(converting to probabilities by normalization)*,
      - Probability of *Yes* = $^{.0053}/_{(.0053+.0206)} = 20.5\%$
      - Probability of *No* = $^{.0206}/_{(.0053+.0206)} = 79.5\%$

Outlook			Temperature			Humidity			Windy			Play	
	yes	*no*		*yes*	*no*		*yes*	*no*		*yes*	*no*	*yes*	*no*
sunny	2	3	hot	2	2	high	3	4	false	6	2	9	5
overcast	4	0	mild	4	2	normal	6	1	true	3	3		
rainy	3	2	cool	3	1								
sunny	2/9	3/5	hot	2/9	2/5	high	3/9	4/5	false	6/9	2/5	9/14	5/14
overcast	4/9	0/5	mild	4/9	2/5	normal	6/9	1/5	true	3/9	3/5		
rainy	3/9	2/5	cool	3/9	1/5								

3 Decision Trees

3.1 Iterative Dichotomizer 3 (ID3)

For any attribute within the dataset, calculate the **information gain**. To calculate information gain, we must first calculate the entropy of each class within the attribute, then the entropy of the class as a whole.
This one is incredibly intense to write out. Look at your homework for calculating entropy

3.2 C4.5/J48

4 Rule Generation

4.1 PRISM

5 Clustering

5.1 IREP/RIPPER/PART

5.2 K-Means

5.3 Hierarchical Clustering