Discussion 2
Probabilistic Machine Learning, Fall 2016

1. Decision Trees

Concepts

(a) Provide an intuitive explanation of why the *information gain criterion* helps us choose a good split in a decision tree.

(b) Is it possible that the depth of a learned decision tree can be larger than the number of training examples used to create the tree? Explain why or why not.

Practice

Consider the following dataset. Our goal is to predict the last column ("Can Play") with the input features. We will be using the **ID3 algorithm** which is essentially **C4.5** for construction of the decision tree.

HW Due?	Temperature	Humidity	Weather	Can Play
Yes	Hot	High	Sunny	No
Yes	Mild	High	Sunny	No
No	Cold	Normal	Rainy	No
No	Cold	High	Rainy	No
Yes	Mild	High	Overcast	No
No	Cold	Normal	Sunny	Yes
No	Hot	Normal	Rainy	Yes
No	Hot	High	Overcast	Yes
No	Mild	Normal	Overcast	Yes
No	Cold	Normal	Rainy	No
Yes	Cold	High	Overcast	No
No	Cold	High	Sunny	Yes

Training

HW Due?	Temperature	Humidity	Weather	Can Play
Yes	Cold	Low	Rainy	Yes
No	Cold	Low	Rainy	No
No	Hot	Low	Sunny	Yes
Yes	Hot	Low	Overcast	No

Testing

Figure 1: Data for Decision Trees.

(a) What is the first feature that you would split on?

(b) What is the number of levels of the decision tree constructed (leaf nodes included)?

(c) What's the training error of this decision tree?

(d) Given the test dataset, how many testing points do we classify incorrectly?

2. Entropy

The entropy H of a discrete random variable X with n possible values is defined by the formula

$$H(X) = -\sum_{i=1}^{n} p(x_i)\ log\ p(x_i).$$

Similarly, the joint entropy of two random variables X with n outcomes and Y with m outcomes is:

$$H(X,Y) = -\sum_{i=1}^{n}\sum_{j=1}^{m} p(x_i, y_j)\ log\ p(x_i, y_j).$$

2.1

As $p(x)$ is the probability of outcome x, $0 \leq p(x) \leq 1$ for any outcome. Use this fact to prove that $H(X) \geq 0$ for any random variable X.

2.2

Consider a discrete random variable Y with a uniform probability distribution over n outcomes, i.e. $p(y) = 1/n$. Is $H(Y)$ bounded as $n \to \infty$?

2.3

The conditional entropy $H(Y|X)$ is defined $H(Y|X) = -\sum_{i=1}^{n}\sum_{j=1}^{m} p(x,y)\ log\ p(y|x)$. Show that $H(X,Y) = H(X) + H(Y|X)$. Here are some hints: $log\ \frac{a}{b} = log\ a - log\ b$; $\quad p(y|x) = \frac{p(x,y)}{p(x)}$; $\quad \sum_{j=1}^{m} p(x_i, y_j) = p(x_i)$

Information Gain

The term "information gain" has been used to describe the extra information we gain about X by including a new variable Y that may lend extra predictive power in a model. Formally, the information gain (also called the mutual information) is defined as:

$$I(X;Y) = \sum_{i=1}^{n}\sum_{j=1}^{m}\ p(x_i, y_j)\ log\ \frac{p(x_i, y_j)}{p(x_i)p(y_j)}$$

2.4

When two random variables X and Y are independent, then their joint distribution $P(X,Y)$ factorizes into $P(X,Y) = P(X)P(Y)$. What is the information gain for two independent random variables? What does this say about the ability of one variable to explain the other?

2.5

Show that $I(X;Y) = H(X) + H(Y) - H(X,Y)$.

3. Algorithm Comparison

We would like you to gain a practical understanding of common ML algorithms for your own projects. Many of these algorithms have been integrated into packages in Matlab/Python/R. For example, Matlab has a ML toolbox that has several widely used algorithms built into it. R has packages that you can load with all the ML algorithms. For this discussion section, we have created a basic skeleton platform for you to use to run standard ML algorithms in. The TAs will demonstrate a short script showing a comparison of several algorithms as applied to modeling credit card transactions.