Bayesian Learning

This document will go over one of the most useful forms of statistical inference known as Baye's Rule and several of the concepts that extend from it. Named after Thomas Bayes this rule has far uses that easily extend into machine learning.

Bayes' Rule:

Charles introduces us to the principal concepts governing Bayesian learning in this section. Bayes' rule is an important concept in probability theory because it allows us to make decisions when facing uncertainty. In essence, Bayes' rule allows us to integrate prior information with our data to come up with new information that we can use to confirm our possible suspicions. Ultimately, Bayes' rule is defined with the following formula:

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

where h represents a hypothesis and D represents our data. In this formula we call $P(h|D)$ our posterior probability. The posterior probability can be defined as the conditional probability that is assigned to a hypothesis after relevant evidence (our data) is taken into account. We are able to compute this posterior probability by using the prior probability multiplied by the likelihood function. The prior probability is our prior belief on the hypothesis, or $P(h)$. The likelihood function is the probability that the labels we assign our data are generated from a given hypothesis, or $P(D|h)$. The product is

normalized by the probability of the data $P(D)$ which sums up the likelihood of the data under all hypotheses - that way, our posterior probabilties over all hypotheses sum up to 1. In general it is very easy to compute the likelihood probabilities.

Let's take a look at an example:
"Suppose we are fishing in a river where 60% of fish are catfish and 40% of fish are stripers. At this location we are only allowed to keep fish that are above 5 lbs. 50% of Striper are over 5lbs and all catfish are above 5lbs. If we catch a fish and we get to keep it what is the probability the fish is a striper?"

First we should define our probabilities:
$P(S)$, the probability that a fish is a striper regardless of any other factors. This is the percentage of fish that are stripers i.e. 40%.

$P(C)$, the probability that a fish is a catfish (or in this case not a striper) is 60%.

$P(K|S)$, the probability you get to keep the fish given that it is a striper. This is given in our problem statement as 50%.

$P(K|C)$, the probability you get to keep the fish given it is a catfish. This is also known to be 100%

$P(K)$, The probability we keep the fish. This can be computed using the law of total probability as:
$P(K) = P(K|S)P(S) + P(K|C)P(C) = .5 * .4 + .6 * 1 = 0.8$ or 80%

These five probabilities are our priors (i.e. our prior beliefs of the distribution of fish in the lake)

Now we are ready to use Bayes' theorem to tackle this problem. We can now find $P(S|K)$

$$P(S|K) = \frac{P(K|S)P(S)}{P(K)} = \frac{0.5*0.4}{0.8} = 0.25$$

So there is a 25% chance that when we keep a fish that it is a striper.

Using Bayes' rule for Bayesian Learning:

We can now use Bayes' rule to make a decision on which hypothesis is optimal in relation to our training data. We can do this by finding the maximum-probability hypothesis given the data across all hypothesis. In math notation this looks like:

$$h_{MAP} = argmax_h P(h|D) \; \forall h \in H$$

The MAP subscript on h stands for Maximum a Posteriori, which is the max posterior given all of our priors.

Since we are computing the argmax our prior on the data isn't exactly relevant. That is, we don't care about the $P(D)$ term in the denominator as it affects all computations equally. This works out nicely too since often, finding out $P(D)$ can be quite difficult. In some instances if our assumption that all $P(h)'s$ are equivalent we instead can compute $P(h|D)$ using the maximum likelihood. The maximum likelihood commonly referred to as *ML* is computed as.

$$h_{ML} = argmax_h P(D|h) \; \forall h \in H$$

Choosing the Best Hypothesis:

Our ultimate goal is to choose the best hypothesis for predicting our data. However, sometimes there may be ties or the differences in prediction may be small (e.g. Suppose $h_1 = 0.7$ and $h_2 = 0.71$, should we choose h_1 or h_2). Occam's Razor states that among competing hypotheses, the one with the fewest assumptions should be selected. How do we go about finding the shortest hypothesis? This is where we need to find the minimal description length.

$$h_{MDL} = \; argmin_{h \in H} \, L_{C_1}(h) \; + \; L_{C_2}(D|h)$$

Here $L_C(x)$ is the description length x under the encoding C

Example:
$H = Decision \; Trees$ and $D = training \; data \; labels$

$L_{C_1}(h)$ is the number of bits to describe tree h

$L_{C_2}(D|h)$ is the number of bits to describe D given h

From this we can see that h_{MDL} trades off tree size for training error.

Bayesian Classification:

So far we've found the most probable hypothesis h_{MAP} given a dataset D. The question now is given a new x what is the most probable classification? Unfortunately $h_{MAP}(x)$ isn't always the most probable classification.

We need to come up with the Bayes optimal classifier which can be defined as:

$$argmax_{v_j \in V} \sum_{h_j \in H} P(v_j|h_i)P(h_i|D)$$

Where the $v_j's$ come from the set of possible classifications assigned by H.

Example:

Suppose that we are given

$P(h_1|D) = 0.4 \qquad P(h_2|D) = 0.3 \qquad P(h_3|D) = 0.3$

and

$h_1(x) = + \qquad h_2(x) = - \qquad h_3(x) = -$

Find the most probable classification of x. Using the information given to us we know $V = \{-,+\}$ and

$P(-|h_1) = 0,\ P(+|h_1) = 1$

$P(-|h_2) = 1,\ P(+|h_2) = 0$

$P(-|h_3) = 1,\ P(+|h_3) = 0$

Therefore

$$\sum_{h_i \in H} P(+|h_i)P(h_i|D) = 0.4 \qquad and \qquad \sum_{h_i \in H} P(-|h_i)P(h_i|D) = 0.6$$

Thus we can conclude that the optimal classifier is -.

In the next lesson you will find ways to improve this classification process!