Clickstream Mining with Decision Trees

The project is based on a task posed in KDD Cup 2000. It involves mining click-stream data collected from Gazelle.com, which sells legware products. Your task is to determine: Given a set of page views, will the visitor view another page on the site or will he leave?

The data set given to you is in a different form than the original. In particular it has discretized numerical values obtained by partitioning them into 5 intervals of equal frequency. This way, we get a data set where for each feature, we have a finite set of values. These values are mapped to integers, so that the data is easier for you to handle.

Datasets

You have 5 files in .csv format:

1. trainfeat.csv: Contains 40000 examples, each with 274 features in the form of a 40000 × 274 matrix.
2. trainlabs.csv: Contains the labels (class) for each training example (did the visitor view another page?)
3. testfeat.csv: Contains 25000 examples, each with 274 features in the form of a 25000 × 274 matrix.
4. testlabs.csv: Contains the labels (class) for each testing example.
5. featnames.csv: Contains the "names" of the features. These might useful if you need to know what the features represent.

The format of the files is not complicated, just rows of integers separated by empty spaces.

Stopping Criterion: Chi-squared criterion

What about split stopping? Suppose that the attribute that we want to split on is irrelevant. Then we would expect the positive and negative examples (labels 1 and 0) to be distributed according to a specific distribution. Suppose that splitting on attribute T, will produce sets $\{T_i\}_{i=1}^{m}$

Let p, n denote the number of positive and negative examples that we have in our dataset (not the whole set, the remaining one that we work on at this node). Let (N is the total number of examples in the current dataset):

$$p_i^{'} = p\frac{|T_i|}{N}$$

$$n_i^{'} = n\frac{|T_i|}{N}$$

be the expected number of positives and negatives in each partition, if the attribute is irrelevant to the class. Then the statistic of interest is:

$$S = \sum_{i=1}^{m}\left(\frac{(p_i^{'} - p_i)^2}{p_i^{'}} + \frac{(n_i^{'} - n_i)^2}{n_i^{'}}\right)$$

where p_i, n_i are the positives and negatives in partition $T_{i.}$ The main idea is that S is distributed (if the class is irrelevant) according to a chi-squared distribution with m−1 degrees of freedom.

Now we must compute the p-value. This is the probability of observing a value X at least as extreme as S coming from the distribution. To find that, we compute P (X ≥ S). The test is passed if the p-value is smaller than some threshold. Remember, the smallest that probability is, the more unlikely it is that S comes from a chi-squared distribution and consequently, that T is irrelevant to the class.

Your Task:

Implement the ID3 decision tree learner on Python or R. Your program should use the chi-squared split stopping criterion with the p-value threshold given as a parameter. Use your implementation with the threshold for the criterion set to 0.05, 0.01 and 1. Remember, 1 corresponds to growing the full tree. Answer the following questions:

1. For each value of threshold, what is your tree’s accuracy and size (size equals number of internal nodes and leaves)? What do you observe? If all your accuracies are low, tell us what you have tried to improve the accuracies and what you suspect is failing.

2. Explain which options work well and why.