CS 4780/5780 Machine Learning Due September 23, 2014

Assignment 2: Decision Trees and Hypothesis Testing

Instructor: Thorsten Joachims *Total Points: 100*

Course Policy: ***Read all the instructions below carefully before you start working on the assignment, and before you make a submission.***

- *Assignments are due at the beginning of class at 2:55pm on the due date in hard copy. Please do not submit printouts of your code unless you are explictly asked to.*
- *Write your NetIDs with submission date and time on the first page of the hard copy. Failing to do so might result in a deduction of points.*
- *Late assignments can be submitted in class or in the handback room (216 Gates Hall). Since it is only open from 12-4pm M-F, weekend submissions (all code and answers to all questions) should be made digitally via CMS, with a hard copy delivered to the handback room as soon as possible afterwards.*
- *No assignment will be accepted after Sunday, September 28, 10am. This is also when the solutions will be released.*
- *The submission time of whatever you submit last (hard copy or CMS) is counted as the official submission time and will determine the late penalty. On days the handback room is closed, CMS is used to determine the submission time.*
- *Upload only the code you wrote to CMS. Do not upload any additional libraries or datasets. Provide a README file with your submission that includes a list of all libraries and instructions needed to run your code. You only need to upload your answers to CMS if you submitted them late on a day the handback room is closed.*
- *All sources of material must be cited. Assignment solutions will be made available along with the graded homework solutions. The University Academic Code of Conduct will be strictly enforced, including running cheating detection software.*
- *You can use common file I/O, plotting and linear algebra libraries for the coding problems. If you are planning on using some other libraries, please ask on Piazza before using them.*
- *Whenever we ask you to compute numbers, always show your work in order to get full credit.*
- *No more than one submission per group.*
- *Please visit the course homepage for details on regrade requests.*

Problem 1: Decision Tree [35 points]

After a hectic first week of going to seven classes, you finally sit down and decide which classes you want to stay in and which classes you want to drop. As you are not confident

about your decisions, you try to collect the opinions of your friends on how they decide whether to drop or stay in a class. Your friends mainly base their decisions on three factors: the difficulty level of homework 1, whether there is a time conflict with another class, and whether the lecture room is too crowded. There are three values for the difficulty level of homework 1: *easy*, *just right* and *difficult*. The other two attributes (conflict and crowded) have only two values (*yes* or *no*). You collected 8 such decisions from your friends and summarized them in the table below.

HW1	Conflict	Crowded	Stay?
Easy	No	Yes	Yes
Just Right	Yes	No	No
Difficult	Yes	Yes	No
Easy	Yes	Yes	Yes
Just Right	No	No	Yes
Difficult	No	Yes	No
Just Right	No	No	Yes
Easy	Yes	No	No

(a) Use the ID3 algorithm and build a decision tree by hand to decide whether a student is going to stay in a class. Use information gain as the splitting criterion. Show all calculations of entropy and information gain. If there is a tie in the leaf node in the number of 'Yes' and 'No' instances, break the tie by deciding 'Yes'.

(b) Because you are suspected to have an allergy, your doctor asks you to keep a log of your eating habits for 30 days (let's call them days 1-30). Every day, you write down the day (as a number), whether you had diary (yes / no) and finally, whether you had any digestion problems (yes / no). Also assume that you don't actually have an allergy, so whether you suffered from any problems each day is randomly yes or no with equal probability. Suppose you give this data the ID3 algorithm to predict whether you will have digestion problems given the day and your consumption of dairy products. Explain in two sentences why ID3 will almost always split on the day attribute first.

(c) Let's suppose you repeat this experiment infinitely many times. After day 30, you just start again numbering your days at 1. More formally, assume that the dairy and day variables are independently and uniformly distributed over their respective values. Will ID3 still suffer from the same problem as in (b)? Give an algebraic expression to justify your answer (*Hint:* Information gain).

(d) Now, let us define the training error of a node as the number of misclassified training examples that fall under that particular node. Assume that there are 2 possible labels for training instances (positive and negative). Assume also that an impure leaf always uses the majority label of its instances to make predictions. Prove that for any training set, the change in training error resulting from a split is ≤ 0 for

any possible split.

(e) Design a dataset with only two attributes to show that ID3 does not always produce an optimal decision tree. An optimal decision tree is defined as the decision tree with the smallest number of decision nodes that is consistent with a training set T. To make things easier, you can assume that you can have multiple copies of the same training example in the training set. Give a table with the dataset and show both trees in a side-by-side comparison.

(f) Explain in at most two sentences why ID3 would never pick the same attribute twice to perform a split assuming there are only categorial attributes.

Problem 2: Decision Tree II [35 points]

In class, you saw an example of the TDIDT algorithm for learning decision trees with categorical attributes. We will now extend the algorithm to work with numerical attributes. The dataset is based on the Wisconsin Breast Cancer Data Set. The decision tree you build will take in a variety of cell attributes and will determine whether it is malignant (M) or benign (B). On CMS you will find the `BCan` dataset: `bcan.train` (the training set), `bcan.validate` (the validation set) and `bcan.test` (the test set). It has been processed to adhere to the SVM-Light format. For further information on this format, see `http://www.cs.cornell.edu/people/tj/svm_light/`.

There are 9 numerical attributes with no missing values. These have been filled in during the pre-processing step for your convenience. Each node in the decision tree shall be a clause of the form $(Attribute_i \leq j)$? where $1 \leq i, j \leq 9$. To achieve this, try all possible values j for every attribute i and pick the most suitable attribute-value combination according to the splitting criterion. If there are multiple attribute-value pairs that are equally good, pick the attribute with the lowest i value and then the value with the smallest j among those combinations. For impure leaf nodes, use the majority class as a node label. If there is a tie, say that the node is malignant (M).

For this assignment, we define the depth of a tree as the number of nodes in the longest path of that tree including the leafs. Hence, a single decision node would have depth 2 (longest path includes the root node and one leaf node).

(a) Implement the ID3 algorithm which is just the TDIDT algorithm with information gain as the splitting criterion. Recall that we defined accuracy as the number of correctly classified examples over the total number of examples. Train until no more progress is possible, i.e., there is no more attribute-value pair that would have an information gain of > 0. Then:

i. Report the training set accuracy.

ii. Report the test set accuracy.

iii. Report the number of total nodes (including the leafs), the number of leaf nodes as well as the depth of the tree.

iv. Print out the attribute-value pairs of the top (root) node and of all its children.

(b) Now restrict the depth d of the decision tree to be in $\{2, \ldots, 20\}$.

i. Plot the test set accuracies against d.

ii. Report the number of total nodes (including the leafs) and the number of leaf nodes for each d in a table.

iii. Discuss the shape of the curve and possible causes in at most three sentences in total.

iv. Report the test set accuracy for the tree with depth d that did best on the validation set. If there is a tie, use the smallest depth.

Problem 3: Hypothesis Testing [30 points]

Now we will evaluate the performance of decision trees against kNN classifiers for a version of a puzzle called Petals and the Rose. There, two dice are rolled and the output is a boolean denoting whether the score was zero or not. More formally, there are two random variables X_1, X_2 distributed uniformely over $\{1, 2, 3, 4, 5, 6\}$. The output variable Y is true if one of the dice shows a 3 or 5. More formally $Y = \bigvee_{i=1}^{2} (X_i == 3) \vee (X_i == 5)$.

(a) On CMS you will find `hyp_test_pred.txt` of predictions for over 400 turns of the game. The first column is whether the score was indeed true or not, the second column lists the predictions of a kNN classifier with $k = 3$, and the third lists the predictions of a decision tree with 20 nodes.

i. Compute the confidence intervals for $z_{95\%}$ for the prediction error for each classifier. You can find the formula for computing these intervals on the lecture slides. You may use the normal approximation if needed.

ii. Use the binomial sign test (i.e., McNemar's test) to decide whether with 95% confidence the error rate of the decision tree is significantly different from the kNN classifier. Show each step of the calculation. Again, using the normal approximation is fine.

(b) Now qualitatively reason why one classifier might work better than the other in this case. Consider the training set shown in Figure 1. The plus symbol "+" denotes examples with label *true* and the circle "$\circ$" denotes examples with a *false* label. Draw the decision boundary of a kNN classifier with $k = 1$. If there are multiple nearest neighbors, assign the label to be positive if at least one is positive.

(c) Consider the decision tree shown in Figure 3. Generate the decision boundary of

this tree for the same training set as in (b). Note that since the decision tree uses discrete values, you can deviate a bit from the true boundary for better legiblity.

(d) The true labels for each point in the two-dice case are shown in Figure 2. Now imagine you modify the game so that you roll 5 dice at the same time. Again, your output Y will be true if any of the dice show a 3 or 5. Based on the decision boundaries generated in parts (b) and (c) and the layout of the true labels, would you in general prefer a kNN or decision tree classifier for this five-dice version of the problem? Explain in at most 2 sentences why.

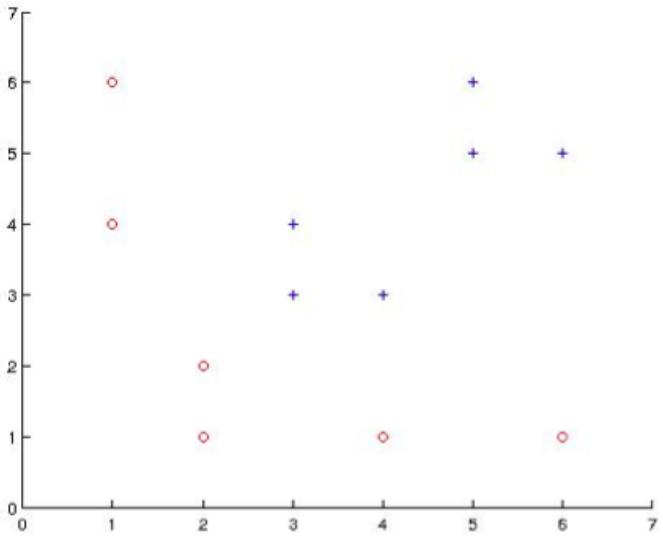

Figure 1: Training Set for two-dice Petals around the Rose. A1 along X-axis, A2 along Y-axis.

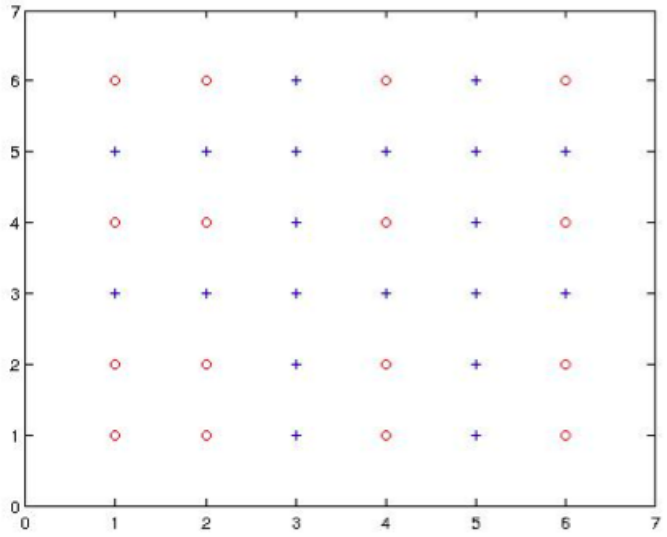

Figure 2: True labels for the two-dice version of Petals around the Rose. A1 along X-axis, A2 along Y-axis.

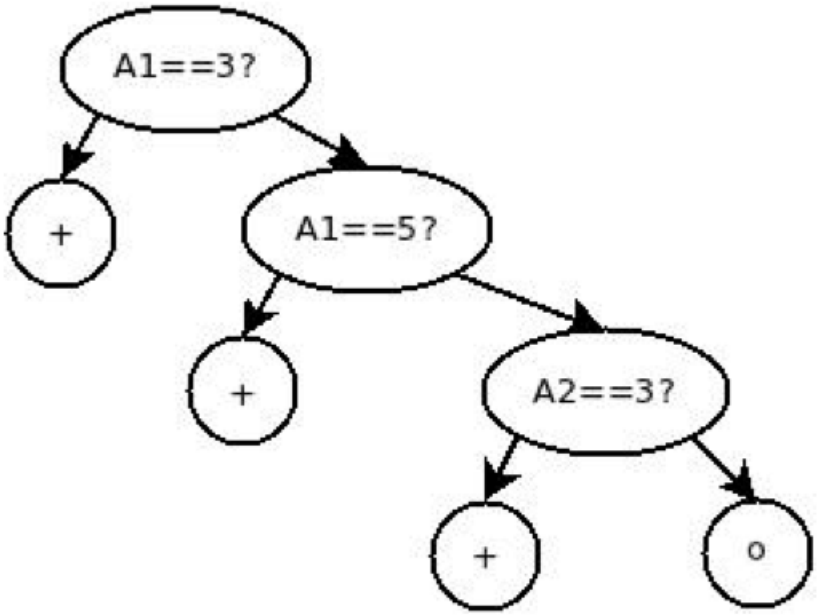

Figure 3: Decision tree for two-dice Petals around the Rose. The left branch represents the predicate evaluating to true.