CS 4780/5780 Machine Learning

Assignment 2: Decision Trees and Hypothesis Testing

Instructor: Thorsten Joachims

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 on the due date in hard copy.

- Write your NetIDs with submission date and time on the first page of the hard copy.

- No assignment will be accepted after the solution is publicized (about 4 days after due)

- The submission time of whatever you submit last (hard copy or CMS) is counted as the official submission timeand will determine the late penalty

- Upload only the code you wrote to CMS. Do not upload any additional libraries. Provide a README file with your submission that includes a list of all libraries and instructions needed to run your code.

- Attach a hard copy of your code to the submission.

- Late assignments can be submitted in class or to Ian Lenz in Upson Hall 5151. Since the fifth floor of Upson is locked on the weekends, weekend submissions (all code and answers to all questions) should be made digitally via CMS, with a hard copy delivered to Ian as soon as possible afterwards.

- 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.

- No more than one submission per group.

Problem 1: Model Selection and Validation [20 points]

(a) Consider the training data S_{train} shown in Figure 1, and the two test sets S_{test1} and S_{test2} shown in Figures 2 and 3. o's indicate positive cases, x's indicate negative.

What are the training and testing accuracies (for both test sets), expressed in terms of the fraction of points classified correctly, for a linear model which classifies a case as positive if $x_2 \geq x_1$?

(b) Give the same for the decision tree shown in Figure 5.

(c) Give the χ^2 statistic for the McNemar test comparing these two models for S_{test1}. With what confidence can we say that one model generalizes better than the other for this validation set?

(d) Repeat part c for S_{test2}.

(e) Did your answers change for the two test sets in parts a and b? What about parts c and d? Why might one stay the same, while the other changes?

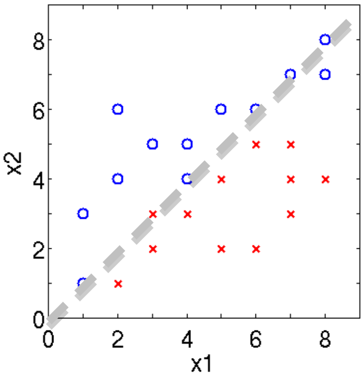

Figure 1: Training data S_{train}

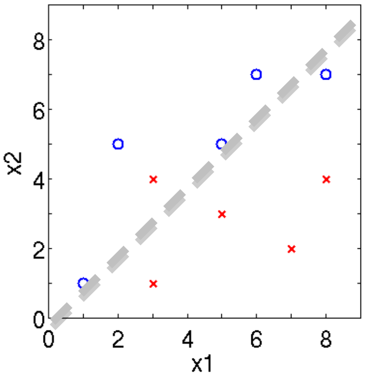

Figure 2: Test data S_{test1}

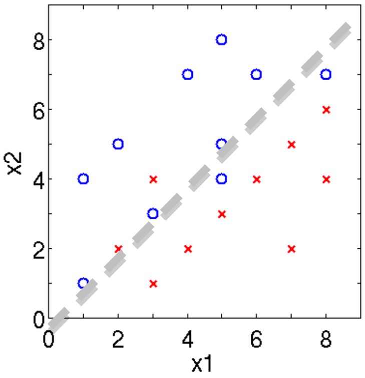

Figure 3: Test data S_{test2}

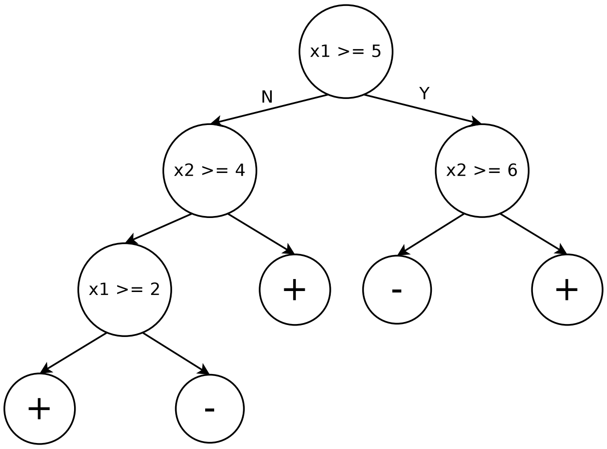

Figure 4: Decision tree for Problem 1

Problem 2: Model Averaging with Decision Trees [40 points]

In this problem we will look at a general machine learning technique of *model averaging*, applied to decision trees. We will compare how a collection of multiple decision trees fares against a single tree, in a simple binary classification task on synthetic data.

In parts *a* and *b* of this problem, you are given synthetic data and you are asked to experimentally investigate the effect of combing multiple decision trees into a single classifier.

In parts *c* and *d* of this problem, you will investigate this idea from a mathematical standpoint, and in part *e*, you will combine your empirical and theoretical results.

For parts *a* and *b* you are given a small toy dataset with 290 training instances and binary labels (*circle.train* – in the usual *SVMLight* format). The instances $\mathbf{x}_{train} \in \mathbb{R}^2$ (in the *SVMLight* file, features 0 and 1 correspond to x and y respectively) were generated by sampling a 10x10 grid, such that:

- Exactly 145 points fall **inside** a circle of radius 3 at the center of the square. These instances belong to the **positive** class.
- Exactly 145 points fall **outside** a circle of radius 3 at the center of the square. These instances belong to the **negative** class.

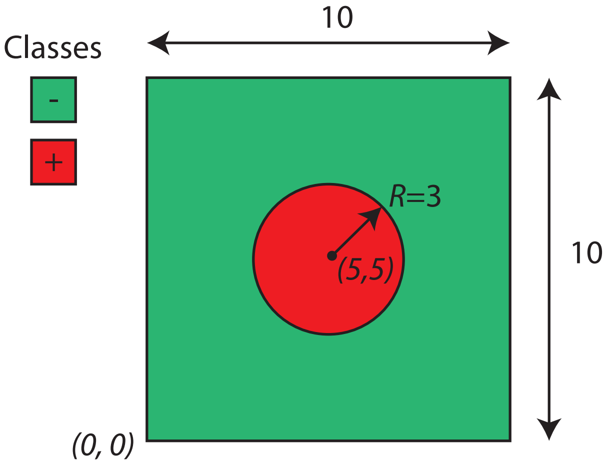

Figure 5: Synthetic dataset for Problem 2

(a) *Individal decision trees* Implement and train a TDIDT decision tree on the provided training set to obtain 0 training error. Use *information gain* as a criterion for splitting, with intermediate nodes of the tree splitting according to the **attribute** $\geq$ **threshold** where **threshold** is an integer (as in HW1).

Generate a plot that shows the decision boundary (region) of this decision tree. The easiest way to do it, is to produce a fine grid of points (test instances) in the 10x10 square, query your decision tree for the label at each point, and then color the respective point with the color of the corresponding label. With a fine enough grid, this will generate regions nicely shaded according to their predicted class.

(b) *Averaging decision trees* We will now explore the idea of combining miltiple decision trees for producing a single classifier. We will be training 101 decision trees as follows:

For $j \in [1, 101]$

- Randomly select 20% of your training data into a smaller training set X_j.
- Train a decision tree T_j to obtain 0 training error on the training set X_j (same splitting criteria as above).

End For

Note that the same training instance ***may*** *appear multiple times in different subsets* X_j

Generate a plot that shows a decision boundary (region) of 2 such decision trees. Select two trees that are particularly illustrative examples of overfitting. Plot their decision boundaries (regions) as you did in part *a*.

Now you will combine all 101 decision trees into a single classifier. For a test instance $\mathbf{x}_{test}$, this classifier will predict +1 if the majority of the 101 decision trees predict +1 on $\mathbf{x}_{test}$, and −1 otherwise.

Generate a plot that shows a decision boundary (region) of this combined classifier.

Comment on how the boundary of the combined classifier compares to the boundaries of the individual classifiers, **and** to the true boundary that was used to generate the data. How does the combined classifier compare to the individual classifiers in terms of overfitting?

(c) Show that a prediction $\hat{y} \in \{-1, 1\}$ of a decision tree T on test instance $\mathbf{x}_{test}$ can be expressed as follows:

$$\hat{y} = \text{sign}\left(\sum_{i}^{n} y_i K(\mathbf{x}_{test}, \mathbf{x}_i)\right)$$

where $\hat{y} = T(\mathbf{x}_{test})$, n is the number of training instances and $K(\mathbf{x}_{test}, \mathbf{x}_i)$ is a similarity measure between a test instance $\mathbf{x}_{test}$ and a training instance $\mathbf{x}_i$, and y_i is the label (binary) of the training instance $\mathbf{x}_i$. The similarity measure $K(\mathbf{x}_{test}, \mathbf{x}_i) = 1/k$ if the test instance $\mathbf{x}_{test}$ is in the same leaf with the training instance $\mathbf{x}_i$ and 0 otherwise, and k is the total number of training instances in that leaf.

(d) Now consider the case of averaging predictions from M different trees:

$$\hat{y} = \text{sign}\left(\frac{1}{M}\sum_{j}^{M}\sum_{i}^{n} y_i K_j(\mathbf{x}_{test}, \mathbf{x}_i)\right)$$

where $K_j(\cdot,\cdot)$ is the similarity measure corresponding to tree T_j. Show that this predicion can also be written in the form:

$$\hat{y} = \text{sign}\left(\sum_{i}^{n} y_i \tilde{K}(\mathbf{x}_{test}, \mathbf{x}_i)\right)$$

where $\tilde{K}$ is a new similarity measure. Give an expression for $\tilde{K}$ in terms of the variables above, **and** an intuitive interpretation of the similarity measure for the decision tree model average. Assume $K(\mathbf{x}_{test}, \mathbf{x}_i) = 0$ if $\mathbf{x}_i \notin X_j$.

Problem 3: Text Categorization with Decision Trees [40 points]

In this problem, we will be performing text categorization with a decision tree. We will be classifying online newsgroup posts into 1 of 4 categories: *AUTO*, *COMPUTERS*, *RELIGION*, *SPORTS*. We will use the same *bag-of-words* feature representation as used in HW1. Three files are supplied for this problem: *groups.train*, *groups.test*, *group.vocab*. *groups.train* and *groups.test* are in the familiar *SVMLight* format and each include a total of 2000 documents (instances), and a vocabulary (feature) size of 2000. Feature-word mappings are provided in the *groups.vocab* file, where the first number on every line of the *groups.vocab* file specifies the index of the feature.

(a) *Decision Tree I* Implement and train a TDIDT decision tree with the *information gain* criterion, with splits of the form **feature** x > 0 (i.e. test if a word corresponding to **feature** x is present in the document) to obtain 0 training error. **Report** test error that this tree obtains on the test set *groups.test*.

(b) *Informative words* List the words (from *groups.vocab*) corresponding to the splits at the **top** 2 levels of the tree. List the words corresponding to the splits at the **bottom** 2 levels of the tree. What difference do you observe between the words at the top and the words at the bottom of the tree. What can you hypothesize about the generalization ability of the complete tree you just trained?

(c) *Early stopping* Modify the decision tree you learned in part *a* by restricting its depth to 10 (i.e. cut off all nodes below the 10^{th} level). **Report** training and test accuracy of this classifier on the test set *group.test*.

(d) *Comparing classifiers* We will now compare the generalization accuracy of the decision tree you trained in part *a* with the early-stopped tree in part *c*. Use the McNemar test and explain whether you can conclude with greater than 95% confidence whether one of the two classifiers has lower generalization accuracy than the other. **Show** all work that gets you to your conclusion.

(e) *Comparing learning algorithms* Suppose that you hypothesize that simply checking for the appearance of the word in the document is not enough. You decide to modify your decision tree learning algorithm to additionally check if the word had also appeared more than once (i.e. your tree can now split according to the rule **feature** x $> \{0, 1\}$. Modify your decision tree to accommodate for this change, and compare the generalization accuracy of the decision trees produced by the training algorithm in part *c* (one threshold, 10 levels) and the modified algorithm that includes the additional splitting threshold (two thresholds, 10 levels). Concatenate training and test sets supplied, and perform 5-fold cross-validation over the combined sets. Use the *paired t-test* to establish whether the two algorithms have different classification accuracies at the 95% confidence level (you can ignore that the different folds are not statistically independent when you apply the test, but always keep in mind that the test is only approximate in this situation). **Show** all work that gets you to your conclusion.

(f) *Model selection* We will now explore the technique of cross-validation for picking an optimal parameter (maximum depth of the decision tree) for our classifier with 1 threshold (**feature** x $>$ 0). Perform 5-fold cross-validation over the combined set from part *e* once for each value of maximum depth in the set $\{2, 3, 5, 10, 50, 80\}$. **Plot** the average accuracy for each run on a linear scale. What tree depth results in the highest average accuracy?