CS434 HW2
Due Oct 24 in Class

PART I

In part I, you will use WEKA to analyze the two artificial data sets we generated and one real data set. You will apply the learning algorithms we learned to each data set and compare their performance.

- **Learning Algorithms**. We will compare Perceptron (in this case, *the voted perceptron*), KNN (i.e., *IBk*), decision tree (i.e., *J48*) .You should use the defaults that weka set for these algorithms with the following exceptions:
    1. **trees>J48** Set *unpruned* to True.
    2. **lazy>IBk**. Set *KNN* to 1 (which is the default; we will experiment with other values below).
- **Data Sets**. We will apply these algorithms to the data sets `hw2-1`, `hw2-2`, and `br`. These data sets are available here: http://web.engr.oregonstate.edu/~xfern/classes/cs434/data/data.html. Each data set has one or more training data files and one test data file:

`br data files:`

`br-test.arff`	`br test data file`
`br-train.arff`	`br training data file`

`hw2-1 data files`

`hw2-1-10.arff`	`10 training examples`
`hw2-1-20.arff`	`20 training examples`
`hw2-1-50.arff`	`50 training examples`
`hw2-1-100.arff`	`100 training examples`
`hw2-1-200.arff`	`200 training examples`
`hw2-1-test.arff`	`test data file`

`hw2-2 data files`

`hw2-2-25.arff`	`25 training examples`
`hw2-2-50.arff`	`50 training examples`
`hw2-2-100.arff`	`100 training examples`
`hw2-2-200.arff`	`200 training examples`
`hw2-2-600.arff`	`600 training examples`
`hw2-2-test.arff`	`test data file`

In case you are curious, here is how we generated the two synthetic data sets. The data set `hw2-1` is generated from two Gaussian distributions. One is centered as (1,0) and the other at (0,1). Both have the same co-variance matrix:

$$\begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$$

`hw2-2` is generated as follows. The x coordinate is generated from an exponential distribution with parameter 1.0. The y coordinate is generated from a uniform random distribution in the interval [0,1]. The class is assigned as follows. If $(x > 0.5)$, the example belongs to the positive class, otherwise to the negative class. However, the class label is flipped with probability 0.1 (so-called "10% label noise").

br is a hand written letter data set that contains letter ***b*** and ***r***. Each example is described by 16 attributes corresponding to 16 pixels of a 4 by 4 image.

You will run the learning algorithms on each training data file and evaluate the results on the corresponding test data files.

- **Results**. You should turn the following. Please provide print out of the results.

1. A table in the following format:

N	Method1	Method2	Method3
hw2-1:			
10	xxx	yyy	zzz
20	xxx	yyy	zzz
50	xxx	yyy	zzz
100	xxx	yyy	zzz
200	xxx	yyy	zzz
hw2-2:			
25	xxx	yyy	zzz
50	xxx	yyy	zzz
100	xxx	yyy	zzz
200	xxx	yyy	zzz
600	xxx	yyy	zzz
br:			
614	xxx	yyy	zzz

Where `xxx, yyy, zzz` give the ***error rates*** of each method on the test data. (Use “Supplied test set” for “Test Option” in the classify tab)

2. Graphs of the results for `hw2-1` and `hw2-2` plotting the performance of each algorithm as a function of the size of the training data set (known as a "learning curve"). I recommend using Matlab, Gnuplot or Excel for constructing the graphs. WEKA does not provide an easy way to do this.

3. Plot of the data points for `hw2-1-200` and `hw2-2-200` with lines showing the decision boundaries learned by Decision tree (J48). This will require that you read the decision tree and understand the decision boundary. J48 displayes the tree in the following format:

```
x1 <= 1.0: positive (75.0/17.0)
x1 > 1.0
|   x2 <= 5.0: negative (42.0/12.0)
|   x2 > 5.0: positive (33.0/10.0)
```

The first line indicates a split on feature x1 with threshold 1.0. The first branch leads to a leaf labeled "positive". The numbers in parentheses indicate that this leaf contains 75 data points of which 17 were misclassified. Indentation indicates child nodes. The vertical bars are intended to make it easier to see the indentations.

Note: You should only plot line segments that separate the two classes (not all separating lines chosen by J48).

4. The results of additional experiments with IBk. Specifically, use the data set hw2-1-50 and repeat IBk training with k set to 3, 5, 7 and 9. Report the error rate on the test set from each choice of k. Note that we use odd numbers for k to avoid ties.

Using Weka

These instructions will describe how to apply the learning algorithms to the BR data set. The others can be processed in exactly the same way, of course. When you start up Weka, you will first see the WEKA GUI Chooser, which has a picture of a bird (a weka) and four buttons. You should click on the Explorer button. This opens a large panel with several tabs, and the Preprocess tab will already be selected.

Click on "Open file...", find and select the "br-train.arff" file. The "Current relation" window should now show "Relation" as BR with 614 instances and 17 attributes. The table and bar plot on the right-hand side of the window will show 316 examples in class 0 and 298 in class 1.

Now click on the "Classify" tab of the Explorer window and examine the "Test options" panel. First we will load in the test data. Click on the radio button "Supplied test set". Then click on the "Set..." button. A small "Test Instances" pop-up window should appear. Click on "Open file...", navigate to the "data" folder, and select "br-test.arff". The Test Instances window should now show the relation "BR" with 613 instances and 17 attributes. You may close this window at this point.

Now we will tell Weka which of the 17 attributes is the class variable. Below the Test options panel, there is a drop down menu with the entry "(Num) x16" selected. Click on this and choose "(Nom) class" instead. [Num means numeric; Nom means nominal, i.e., discrete]

Now we need to select the learning algorithm to apply. Go to the "Classifier" panel (near the top) which shows a button: "Choose". Click on "Choose", and you will see a hierarchical display whose top level is "weka", whose second level is "classifiers", and whose third level contains seven general kinds of classifiers: "bayes", "functions", "lazy", "meta", "trees", "rules". Perceptron is under the "functions" category, Decision tree is under the “tree” category and KNN is under the “lazy” category.

Once we have chosen an algorithm, it will be listed next to the “choose” button with its default parameter choices. To change these choices, click on it, you will be given an interface to modify parameters. Click the “More” button to get more information about the parameters. After setting parameters, click ok. Now we are ready to run the algorithm. Make sure you have the right test option and then click on the "Start" button, and the Classifier Output window will show the output from the classifier. This output consists of several sections:

- Run Information: Details of the data set
- Classifier model: The learned model. This part will be different for different algorithms. For example for Decision tree, it will display the learned decision tree.
- Evaluation on test set: This gives various statistics. The key item is the second one: Incorrectly Classified Instances will be expressed as a count and a percentage. You should report the percentages in your answer. One other item of interest comes at the very end: The Confusion Matrix. This shows how many false positive and false negative errors were made.

PART II

Probability

1. (10pts) We have two identical bags. Bag A contains 4 red marbles and 6 black marbles and bag B contains 5 red marbles and 5 black marbles. Now we random chose a bag and drew a marble from the chosen bag and it turns out to be black. What is the probability that the chosen bag is bag A?

2. (6pts) Suppose we have class variable Y and three attributes X_1, X_2, X_3 and we wish to calculate $P(Y|X_1;X_2;X_3)$, and we have no conditional independence information.
   (a) Which of the following sets of probabilities are sufficient for calculation?
      i. $P(Y); P(X_1|Y); P(X_2|Y); P(X_3|Y)$
      ii. $P(X_1;X_2;X_3); P(Y); P(X_1;X_2;X_3|Y)$
      iii. $P(X_1;X_2;X_3); P(Y|X_1); P(Y|X_2); P(Y|X_3)$
   (b) Now suppose we know that the variables X_1, X_2, X_3 are conditionally independent given the class variable Y. Which of the above 3 sets are sufficient now?

Decision tree

3. (20 pts) Given the following data set:

V	W	X	Y
0	0	0	0
0	1	0	1
1	0	0	1
1	1	0	0
1	1	1	0

The task is to build a decision tree for classifying Y.

(a) Compute the information gain of attributes X, V and W respectively.

(b) Use information gain for selecting test and produce the full decision tree generated by the top-down greedy algorithm described in class. (Stopping criterion: stop if all the instances belong to the same class.)

(c) Considering the following two strategies for avoid over-fitting.

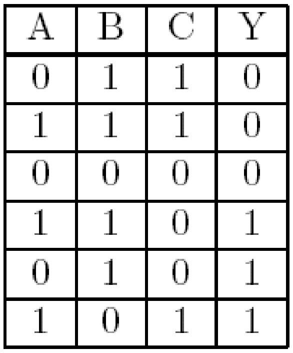
  i. The first strategy stops growing the tree when the information gain of the best test is less than a given threshold ε.

  ii. The second strategy grows the full tree first and then prunes the tree bottom-up: start from the lowest level of the tree and prune a sub-tree if the information gain of the test is less than a given threshold ε. (Note that you should stop checking level t if none of sub-trees at level t+1 satisfies the pruning criterion.

  Let ε be 0.001 for both cases, write down the resulting tree for each strategy and compare their training errors.

(d) Discuss the advantages and disadvantages of each of these two strategies.

4. **Naïve Bayes Classifier. (15pts)** We will use the following training set to build a Naïve Bayes classifier. A, B, and C are three binary attributes and Y is the target class label.



A	B	C	Y
0	1	1	0
1	1	1	0
0	0	0	0
1	1	0	1
0	1	0	1
1	0	1	1

a. Based on the training data, compute the prior distribution for Y.

b. Based on the training data, compute the distributions P(A|Y), P(B|Y) and P(C|Y).

c. What prediction will the naïve Bayes classifier make for a new example (A=1, B=0, C=0)?