Homework 2

Christina Pavlopoulou
cpavl001@ucr.edu

Niloufar Hosseini Pour
nhoss003@ucr.edu

Sanjana Sandeep
ssand024@ucr.edu

Andres Calderon
acald013@ucr.edu

January 25, 2016

1 The Data

In order to accomplish the requirements of the homework, we have selected two dataset from UC Irvine Machine Learning Repository[1]. At the following sections, we briefly describe each one of them.

1.1 Zoo[1]

This dataset has 101 objects and 17 attributes. It records characteristics of 101 animals and groups the animals into one of 7 possible categories (mammal, bird, reptile, fish, amphibian, insect, and invertebrate). These 7 categories are represented by integers from 1 to 7 accordingly. Most of the attributes are boolean and it has no missing values.

1.2 Seeds[2]

This dataset contains 7 attributes and one class attribute referring to different characteristics of wheat's seeds. The attribute values are real numbers that is why we had to normalize them, using the given command. The instances are categorized into 3 classes which define the three different varieties of wheat of kernels (Canadian, Rosa, Kama). For each class there are 70 instances, so the total number of records in this dataset is 210. There is not missing values in this data set.

2 The Code

To apply the kNN classifier, the data sets must be divided into two sections: test data set and train data set. We divided the datasets into two halves, so for Zoo we used the first 50 instances as test set and the instances from 51 to 101 as train set. Similarly, for Seeds we used the first 105 instances as test set and the remaining as train set.

Figure 1 shows the code used to run the kNN classifier. We follow to explain in detail the code using the Zoo data set. For Seeds the code is the same, we just change the input file and the number of attributes and instances accordingly.

Line 1 uses the command `xlsread` to import our data (which we copy into the `.xlsx` file). In order to generate random orderings (permutations), we shuffle the rows in our data using the `randperm` command (line 2). Line 3 omits the first column from our attributes because it contains the animals' names and it is unique for each animal. The last column is also omitted because it is the class type. So out test and train sets columns are from 2 to 16 as show lines 3 and 4. In line 5, the group set includes only the last column (since classification is based on this class type). We compute the class type of our test set by running line 6. To calculate the accuracy, we have to compare the class labels we obtained with the original class label. Lines 7 to 9 use the command `classperf` to compute and print the accuracy.

[1] `https://archive.ics.uci.edu/ml/datasets/Zoo`
[2] `https://archive.ics.uci.edu/ml/datasets/seeds`

```
zoo = xlsread ('zoo.xlsx');
zoo_RandomRows = zoo (randperm (101),:);
test = zoo_RandomRows (1:50,2:16);
train = zoo_RandomRows (51:end, 2:16);
group = zoo_RandomRows (51:end,17);
class = knnclassify(test, train, group);
true_label= zoo_RandomRows(1:50,17); //last column is class type
cp=classperf(true_label,class);
cp.CorrectRate
```

Figure 1: Code to run kNN classifier

3 The Answers

3.1 Zoo

1. Work out the default rate accuracy for the dataset, are you beating it?
   The default rate accuracy is 41/101 = 0.40 (the largest category has 41 animals). The original accuracy is 0.9800, so we are beating the default rate accuracy.

2. Make the training data smaller (maybe half its size) and measure the accuracy again. Did it change? Why?
   After making the training set half its size the accuracy became 0.8800. So, the bigger the training data set is, the more accurate results are, because our model is more trained.

3. Try removing some features, one by one. Can you find one that when removed makes no difference (or makes things better)? Can you find one that when removed makes the accuracy much worst?
   Table 1 summarizes the results. So, there are some features in this data set that do not change the accuracy, and such features are irrelevant to the data set. While, the feature 2,5,11 and 14 are relevant to the data set. There is no feature that improves the accuracy if it is removed. There are some features that when removed made the accuracy worse but not much worse.

Feature removed	**Accuracy**
1	0.9800
2	0.9600
3	0.9800
4	0.9800
5	0.9600
6	0.9800
7	0.9800
8	0.9800
9	0.9800
10	0.9800
11	0.8800
12	0.9800
13	0.9800
14	0.9600
15	0.9800

Table 1: Accuracy in Zoo after removing a particular feature.

4. Add one feature column that contains just random numbers (use randn), what happened the accuracy? Add two feature columns that contain just random numbers, then Add four feature columns that contains just random numbers etc. How many did you have add before the accuracy decreased down to the default rate?
   Table 2 shows the accuracy after adding 1, 2 and 4 random features. Just after adding 140 random features, the accuracy decreased down to the default rate.

Random features	Accuracy
1	0.9400
2	0.9800
4	0.9400

Table 2: Accuracy in Zoo after adding random features.

3.2 Seeds

1. Work out the default rate accuracy for the dataset, are you beating it?
   As we described in section 1.2, the instances of this dataset are categorized into 3 classes and each class has the same number of elements. As a result, the default accuracy rate is 0.3333. Using the KNN classifier and dividing the whole dataset into half (training and test set), we beat the default accuracy as we achieve 0.9429.

2. Make the training data smaller (maybe half its size) and measure the accuracy again. Did it change? Why?
   We, also, tried to decrease the training set, to see what happens with the accuracy. Firstly, we decreased it by half and the accuracy dropped to 0.9048. We, even, decreased the training set to its quarter and we got 0.8571 accuracy. The results show, that by decreasing the training set, we get lower accuracy. This happens because our model needs more training to be more accurate. However, we observed that the performance of the model is very good even with less training data.

3. Try removing some features, one by one. Can you find one that when removed makes no difference (or makes things better)?
   Table 3 summarizes the results. Attributes 1,2,3 and 5 are irrelevant during classification, leave them out do not affect the accuracy. Removing attributes 6 or 7 decreases the accuracy, so we can conclude they are relevant to the process. On the other hand, removing attribute 4 will increase the accuracy of the classifier.

Feature removed	Accuracy
1	0.9429
2	0.9429
3	0.9429
4	0.9524
5	0.9429
6	0.9143
7	0.9048

Table 3: Accuracy in Seeds after removing a particular feature.

4. Add one feature column that contains just random numbers (use randn), what happened the accuracy?
   Table 4 shows the accuracy after adding 1, 2 and 4 random features. Although more than 800 random features were added, the accuracy did not decrease down to the default rate. The lowest value for the accuracy was 0.5700.

Random features	Accuracy
1	0.7524
2	0.8190
4	0.8000

Table 4: Accuracy in Seeds after adding random features.

References

[1] Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.