COMP24111 - January 2013 - Answers

Todd Davies

January 9, 2015

Please don't assume these answers are right. This is me attempting a past paper for revision purposes; I could have got it all wrong ;)

Section B

Part 1

a)

We had $165 + 34 + 12 + 96 = 307$ examples.

Since $y = 1$ is the rare class, the value for the sensitivity is:

$$\frac{Correct2}{Correct2 + FalsePositive2} = \frac{96}{96 + 12} = 0.\dot{8}$$

Since $y = 0$ is the less rare class, the value for the specificity is:

$$\frac{Correct1}{Correct1 + FalsePositive1} = \frac{165}{165 + 34} = 0.82915$$

b)

$$w * r * (a - p) * i$$

Where

w	The weight for the input
r	The learning rate (this can be adjusted to change how fast the perceptron learns)
a	The actual label of the data item
p	The predicted label of the data item
i	The value of the data label

c)

- The some ML algorithms will do feature selection automatically. For example, a perceptron adjusts the weights of each feature in the learning process, and therefore less indicative features will get lower weights.
- Charlie should have used cross validation to train his learning algorithm. Using 100 students to train means that he can only test his model on the same data, which will make it 'easy' for the model, and risks overfitting.

- The Perceptron Convergence Theorem isn't

$$\text{classification accuracy} = \text{learning rate} * \text{number of samples}$$

  It is 'If the data is linearly separable, then after a finite number of iterations, then the learning algorithm will converge on a set of weights', besides an accuracy of 90% for this dataset could be achieved by classifying each sample as healthy.

- If he sets $k = 100$ as the parameter for a K-NN classifier, he will just classify as the most common class (healthy), he needs to set a (lower) value of k based on the outcome of a technique such as bootstrapping.
- Charlie should tune the learning rate and the number of training iterations based on the effectiveness of the perceptron. This means he needs to plot a graph, or look at a table of the accuracies of the perceptron for different learning parameters, and pick the best parameters.

Question 2

Part a)

```
id3(inputData) {
  if(inputData.length == 0) { return null; }
  else if(inputData has no unchosen attributes) {
    return Leaf(mostCommon(inputData));'
  }
  if(inputData has uniform labels) {
    return Leaf(uniform class);
  } else {
    Tree branch = new Tree();
    importantFeature = getMostImportantFeature(inputData);
    for(value in inputData[importantFeature]) {
      branch.addBranch(value, id3(data[value]));
    }
    return branch;
  }
}
```

The two base cases are:

- When all the input data has the same label, return a leaf of the class label.
- When we've made a decision on every attribute already at some depth of the tree, return the most common class of the data items left.

This is how to calculate the entropy:

$$H(X) = -\sum_{i=0}^{p} p(x_i) log_2 p(x_i)$$

Where p is the number of dimensions (attributes), and x_i is the ith class.

The formula to determine the most important feature is:

$$Gain = H(X) - H(X|W)$$

Where $H(X)$ is the entropy before the split, and $H(X|W)$ is the weighted average entropy after the split.

The 'most important' attribute is the one that maximises the $Gain$,

part c)

An ensemble learning algorithm aims to use several different models to classify data. A method such as boosting is used to train several different models with different training data, and then they are used as a 'committee' when it's time to classify new data.

For an effective classifier, committee members should be diverse, and should disagree to some extent on the classifications of data.

The difference between boosting and bagging is that boosting trains the models one by one weights bootstrapping for the next model according so that data items incorrectly classified by the previous model are given higher weights.

Bagging is a parallel ensemble learning algorithm (models can be trained at the same time), which could be implemented in a more efficient manner than boosting, which can (due to having to wait for the incorrectly classified items from the previous model) only operate in a serial manner.

Boosting

```
class Ensemble {

  Model[] models;

  Ensemble(trainingData, trainingLabels, numCommittee) {
    Model[] models = new Model[numCommittee];
    boolean[] previousWrong = new boolean[trainingData.length];
    for(int i = 0; i < numCommittee; i++) {
      Bootstrap bootstrap = new Bootstrap(traniningData, previousWrong);
      models[i] = (new Model()).train(bootstrap.trainingData);
      previousWrong = models[i].test(bootstrap.testingData) != trainingLables;
    }
  }

  test(testingData) {
    List<MLClass> results = new List<MLClass>();
    for(Model m : models) {
      results.add(m.classify(testingData));
    }
    return results.average();
  }
}
```

Bagging

```
class Ensemble {

  Model[] models;

  Ensemble(trainingData, numCommittee) {
    Model[] models = new Model[numCommittee];
```

```
    for(int i = 0; i < numCommittee; i++) {
      Bootstrap bootstrap = new Bootstrap(traniningData);
      models[i] = (new Model()).train(bootstrap.trainingData);
    }
  }

  test(testingData) {
    List<MLClass> results = new List<MLClass>();
    for(Model m : models) {
      results.add(m.classify(testingData));
    }
    return results.average();
  }
}
```

Question 4

Part a)

i)

The centroid of D_1 is $(1, 1)$, and the centroid of D_2 is $(2, 3)$.

$$\begin{aligned} J &= \sum_{k=1}^{k} \sum_{X_i} D_k d^2(x, m_k) \\ &= \left(\sqrt{2}^2 + \sqrt{2}^2\right) + \left(\sqrt{13}^2 + \sqrt{13}^2\right) \\ &= 4 + 26 = 30 \end{aligned}$$

ii)

The centroid of D_1 is $(2, 1)$, and the centroid of D_2 is $(1, 3)$.

$$\begin{aligned} J &= \sum_{k=1}^{k} \sum_{X_i} D_k d^2(x, m_k) \\ &= \left(\sqrt{5}^2 + \sqrt{5}^2\right) + \left(\sqrt{10}^2 + \sqrt{10}^2\right) \\ &= 10 + 20 = 30 \end{aligned}$$

iii)

The centroid of D_1 is $(2, \frac{2}{3})$, and the centroid of D_2 is $(0, 6)$.

$$
\begin{aligned}
J_1 &= \sum_{k=1}^{k} \sum_{X_i} D_k d^2(x, m_k) \\
&= \left(\sqrt{\frac{2\sqrt{13}}{3}}^2 \sqrt{\frac{2}{3}}^2 + \sqrt{\frac{2\sqrt{10}}{3}}^2 + \sqrt{5}^2 \right) + 0 \\
&= 5.18
\end{aligned}
$$