Assignment 4
Group 30

John Walker Adam Fiksen Giovanni Charles

December 6, 2012

1 Results

1.1 Part 1: Comparison of values of F1 measure

Emotion	name	Decision Tree	Neural Network	CBR
1	anger	23.5294	34.4186	35.4680
2	disgust	42.8044	63.6086	52.6316
3	fear	46.9799	55.0143	61.9780
4	happiness	75.6381	78.1659	83.3333
5	sadness	39.0533	41.4894	48.9583
6	suprise	74.0360	80.0000	81.9742

1.2 Part 2: T-test of the clean data

1.3 Part 3: T-test of the noisy data

2 Questions

2.1 Which algorithm performed better overall in terms of values of F1 measure (part I)? Which algorithm performed better when comparison was performed using the t-test (part II and part III)? Can we claim that this algorithm is a better learning algorithm than the others in general? Why? Why not?

The CBR performs best overall when dealing with noisy data. We assume it achieves this through our tiered system. Although new cases appear, they are seldom used for classification since they are too far from the typical cases in a cluster. This means our CBR can maintain its generality while still learining from the litte relevant data there is in the noisy data set while the other algorithms are overfitted to the clean data.

2.2 How did you adjust the significance level in order to take into account the fact that you perform a multiple comparison test?

As we are performing multiple statistical tests this increases the chance that a significant result is produced simply by chance for example if you did 100 tests at the $P < 0.1$ significance level then you expect 10 of the tests to come back as significant simply by chance. We therefore had to adjust our signficance level to counter this problem, to do this we used the Bonferroni correction. The Bonferroni correction works by simply dividing your significance (in our case 0.05) by the number of tests (in our case 3) to produce a new significance level for an individual test of 5 / 3;

2.3 Which type of t-test did you use and why?

We used the inbuilt "ttest", paired t-test. We decided to use this since they are not independent between the different algorithms. Since each of our folds are decided deterministically our algorithms use the same training examples to predict corresponding labels so it. Therefore it makes sense to pair these predictions of the same data and compare the difference and check if they are similar.

2.4 Why do you think t-test was performed on the classification error and not the F1 measure? WhatâĂ§s the theoretical justification for this decision?

Each fold will have a different number of positive and negative examples. The F1 will not be identically distributed since it is an average of the recall and precision rates, a precise algorithm will perform better in folds with more positive examples skewing the results.

This means the sample error for that fold is no longer an i.i.d. To calculate the variance of the sample errors the central limit theorm is used under the assumption that the samples are i.i.d, this means that the variance used in the t-test calculation will not be representative.

2.5 What is the trade-off between the number of folds you use and the number of examples per fold? In other words, what is going to happen if you use more folds, so you will have fewer examples per fold, or if you use fewer folds, so you will have more examples per fold?

As the number of folds decreases the size of your training and validation set for each fold increases until you reach one fold which simply splits your data into a 2/3 training and 1/3 validation set, if you then build your system only considering this data you are likely to produce a system which performs very well with your validation data but not very well with unseen data, the system has become overfitted. By performing cross validation you hope to expose your system to a wide variety of data and thus you can see how it performs over the whole dataset.

By increasing the number of folds you therefore seek to avoid overfitting by training the system to predict a wide variety of validation sets, this works until a point but as the number of folds increases the size of the training set decreases, eventually the training set will become so small that the system can't possibly learn enough from it to classify the validation set and the system will begin to underfit the data and appear to be performing worse than it actually does.

As a minor point, many small folds will cause a large overhead in computation and is not suggested for frequent evaluation of the performance.

2.6 Suppose that we want to add some new emotions to the existing dataset. Which of the examined algorithms are more suitable for incorporating the new classes in terms of engineering effort? Which algorithms need to undergo radical changes in order to include new classes?

The eager learning algorithms, decision trees and neural networks, will require the most effort since they will have to be retrained from scratch using a dataset including examples of the new emotion.

Of these neural networks will require the most computational effort due to the large number of parameters. The requirements to correctly predict the new emotion could move any number of parameters for network training meaning the search for an optimal network will have to be redone.

In the lazy learning, case based reasoning, algorithm it is possible for you to create a new cluster and feed it examples through the 'retain' method. However, an active system may have more of an understanding of the typicality of the current cases in the system and therefore the new emotion will be at a slight disadvantage and in the wrong environment it may be isolated and not perform as well. It is still possible to retrain the system from scratch to acheive better classification rates.