CO395 Machine Learning
CBC #4
t-test

Group 1

Yong Wen Chua, ywc110

Thomas Morrison, tm1810

Marcin Baginski, mgb10

Marcin Kadziela, mk4910

Contents

Results of the t-test

Clean dataset

	DT vs. ANN	DT vs. CBR	ANN vs. CBR
Emotion 1	3.0498	8.9429	3.9477
Emotion 2	4.2663	3.9580	1.0657
Emotion 3	1.0668	6.0045	2.2500
Emotion 4	4.6751	6.2453	-1.2377
Emotion 5	3.8522	6.9003	5.5578
Emotion 6	2.9902	3.5254	0.5006

Table 1: t-values for every emotion and algorithm on the *clean* dataset

	DT vs. ANN	DT vs. CBR	ANN vs. CBR
Emotion 1	different	different	different
Emotion 2	different	different	similar
Emotion 3	similar	different	similar
Emotion 4	different	different	similar
Emotion 5	different	different	different
Emotion 6	different	different	similar

Table 2: t-values for every emotion and algorithm on the *clean* dataset

Noisy dataset

	DT vs. ANN	DT vs. CBR	ANN vs. CBR
Emotion 1	7.7947	8.4406	1.0476
Emotion 2	6.2755	7.9588	2.2119
Emotion 3	2.0966	1.9214	0.2191
Emotion 4	2.5572	5.3489	0.5100
Emotion 5	1.0530	1.8700	1.3814
Emotion 6	3.9404	3.0950	-0.7212

Table 3: t-values for every emotion and algorithm on the *noisy* dataset

	DT vs. ANN	DT vs. CBR	ANN vs. CBR
Emotion 1	different	different	similar
Emotion 2	different	different	similar
Emotion 3	similar	similar	similar
Emotion 4	similar	different	similar
Emotion 5	similar	similar	similar
Emotion 6	different	different	similar

Table 4: Interpretation of the t-values for every algorithm for the *noisy* dataset

Questions

Performance of the algorithms

For the *clean* dataset, it looks that the CBR algorithm performed clearly better than the Decision Trees. The error rates were statistically different for each emotion and further, manual examination of them shows, that they were almost always smaller in the CBR. It seems that the Artificial Neural Networks perform statistically better than the Decision Trees for most emotions (except emotion 3). It is difficult to assess the performance of the ANN vs. CBR since the t-test returned mixed results.

In case of the *noisy* dataset, all three algorithms perform similarly and it is difficult to assess whether any of them has performed significantly better. This is particularly the case in ANN vs. CBR since according to the t-test the distributions of errors for both algorithms across all the emotions are statistically similar.

Having said that, it cannot be claimed that any of the algorithms is clearly a better learning technique than the others. The performance of the algorithms is heavily dependent on the data which needs to be classified and the implementation (especially in case of the CBR). For example, let's assume that we have to classify a vector of 5 features, $[a, b, c, d, e]$ to a binary class. Additionally, let's assume that the first feature a is the most decisive one and carries the most information for the classification. The Decision Trees would actually perform very well on this particular problem, since during training they rely on the information gain of each attribute. Especially if we pruned the tree after the first iteration of the learning algorithm, it might achieve superb performance. On the other hand, a simple implementation of the CBR would try to compare the entire feature vector, even though the variables $[b, c, d, e]$ might actually carry no information whatsoever.

Adjustment of the significance level

Since we have 3 algorithms and we want to compare each with every other, we need $\frac{3\times 2}{2} = 3$ multiple comparisons. Our initially chose significance level was $\alpha = 0.05$ which, after applying the Bonferroni correction, is equal to $\alpha = \frac{0.05}{3} \approx 0.02$. The t-value for 9 degrees of freedom and $\alpha = 0.02$, which we used to determine whether the samples are statistically different, is $t = 2.821$.

Type of the t-test

In each fold for each algorithm the test set consisted of the same examples. For this reasion, we used the paired t-test because the values of the error rate which we are comparing are clearly not independent.

Classification error vs. F_1 measure

F1 measure is based only on the true positives, false positives and false negatives, while the error rate also considers the true negatives. For this reason, the error rate is a more comprehensive measure and hence better suited for the t-test.

Trade-off between the number of folds and examples per fold

We have a fixed number of examples (1004 for the *clean* dataset and 1001 for the *noisy* dataset) which we should split into n folds. Increasing the number of examples in each fold also increases the accuracy/resolution of the confusion matrix. In other words, if we assume that the confusion matrix follows some probability distribution (and so does the error rate for this matrix), then increasing the number of examples for each fold is like drawing more samples from this probability distribution. Obviously, if we have more samples from a given probability distribution, then we can compute a better approximation of its probability density function. To sum up, having more test examples per fold allows us to generate a better approximation of the performance of the algorithm for the unseen data.

The disadvantage of increasing number of samples in each fold is that we are decreasing the number of times we can compute the error rate which we use in the t-test. By having the smaller number of samples for the t-test we are essentially decreasing the confidence that it will return meaningful results. For example, if we used the t-test to assess only four samples drawn from Normal distribution with the exact same mean and variance, the t-test might often fail and claim that they come from distributions with two different means. On the other hand, comparing 10000 samples will reduce the probability of such error almost to zero. The same principle can be applied to our case of trying to determine whether the error rates for two different algorithms are statistically different.

Additional emotions

In the answer to this question we are assuming that the new emotions do not change the classifications assigned to the already existing data.

If we wanted to add new emotions to the dataset, the Case Based Reasoning algorithm would require the fewest changes. We would simply add the new examples to the existing Case Base which would be the only required change.

The Decision Trees and Neural Networks however would require a complete re-training. We would need to partition the new set of examples together with the old ones into training and validation set and then run the training algorithm. For the Decision Trees and the Neural Networks, the existing trees and networks would have to be completely discarded and their number would also change (especially if we used n single-output ANNs). Also, for the ANN, we would additionally need to optimise the entire set of parameters, which is the a very computationally expensive task.

Appendix 1 - error rates on the clean dataset

Decision Trees

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10
Emotion 1	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Emotion 2	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Emotion 3	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Emotion 4	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Emotion 5	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Emotion 6	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%

Table 5: Error rates for each fold and each emotion returned by the Decision Trees algorithm on the *clean* dataset

Artificial Neural Networks

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10
Emotion 1	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Emotion 2	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Emotion 3	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Emotion 4	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Emotion 5	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Emotion 6	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%

Table 6: Error rates for each fold and each emotion returned by the ANN algorithm on the *clean* dataset

Case Based Reasoning

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10
Emotion 1	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Emotion 2	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Emotion 3	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Emotion 4	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Emotion 5	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Emotion 6	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%

Table 7: Error rates for each fold and each emotion returned by the CBR algorithm on the *clean* dataset

Appendix 2 - error rates of the noisy dataset

Decision Trees

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10
Emotion 1	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Emotion 2	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Emotion 3	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Emotion 4	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Emotion 5	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Emotion 6	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%

Table 8: Error rates for each fold and each emotion returned by the Decision Trees algorithm on the *noisy* dataset

Artificial Neural Networks

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10
Emotion 1	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Emotion 2	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Emotion 3	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Emotion 4	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Emotion 5	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Emotion 6	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%

Table 9: Error rates for each fold and each emotion returned by the ANN algorithm on the *noisy* dataset

Case Based Reasoning

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10
Emotion 1	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Emotion 2	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Emotion 3	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Emotion 4	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Emotion 5	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Emotion 6	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%

Table 10: Error rates for each fold and each emotion returned by the CBR algorithm on the *noisy* dataset