Machine Learning Assignment 4: Case Based Reasoning

Group 31
Andrew West

March 9, 2012

1 Implementation

Due to the great freedom of choice available in deciding on a case and case-base structure, it was decided that this problem should be approched in stages. First, a basic Case-Based Reasoning system would be implemented with a simple case retrieval algorithm to act as a benchmark for performance. Then, a more complex case and system structure would be implemented with the aim of improving accuracy of predictions until a maximum could be found. Finally, the CBR would be optimised and re-organised to reduce the time taken to retrieve results, while attempting to maintain the accuracy achieved in the previous step.

Case Structure
Cases originally contained just two values - a vector of active AUs for that case and a label corresponding to the emotion the system believed this case to represent. This was sufficient for the original (simple) system, but it was felt that the performance of later systems could be improved if more information were stored. To this end, an extra field was added to each case - 'typicality', which represented how frequently a given case was observed by the system. This value would later be used by some of the algorithms for accuracy and time optimisation.

Case-Based Reasoning System Structure
The first implementation of the case-base was a simple unsorted list of cases, which could then be iterated over to find similar cases and which could simply have new cases appended to the end. This did allow for accurate results as every existing case was considered with equal weight by the system ensuring nothing would be missed, but this method proved to be extremely slow to produce results. It was found through testing that the relative lengths of active AU vectors of a pair of cases had a high impact on whether or not they were similar, so the case-base was restructured to contain separate 'branches' for cases of different lengths; a branch for cases of length 1, length 2, length 3 etc. All cases with 7 or more active AUs were put on a single branch to prevent the tree from growing too large, as although none of the test examples contained more than 7 active AUs the system was designed to be expandable to more complex cases. Within each branch, cases were sorted in descending order of typicality, so that the most common examples were located at the top of their respective branches and would therefore be checked for similarity first.

Case Retrieval
Multiple similarity measures were tested for case retrieval, and the source code of the function 'retrieve.m' displays the progression of retrieval algorithms with links to the source code for each one - note that most of these were implemented before the case-base was restructured, and that they may no longer work with the current system. Each algorithm was run through the 10-fold cross-validation script 50 times to obtain an average performance value based on the average F_1 value of each run. All of these algorithms were tested on the entire case base - no preselection of cases was attempted in order to maximise accuracy of prediction.
For the most simple CBR system, a random retrieval algorithm was implemented to act as a benchmark for comparison. This simple selects a random case from the entire case base and returns it as a 'similar' case. This performed poorly as anticipated. The next algorithm implemented was the AU vector length comparison test suggested in the spec, where a case which has the same length as the new case (or a similar length if none are exactly the same) is returned. Again, this was expected to perform badly and testing showed only a slight improvement over completely random selection.
The third algorithm checked for the number of matching AUs in the respective active AU vectors of the

two cases being compared. If two cases were tied for similarity with the new case, the system chose the one with the greatest typicality. This produced significantly better results than the previous algorithms, and all subsequent algorithms and their iterations were based on a comparison of the number of matching AUs as this appeared to be the best metric of similarity.
The final algorithm was based upon the previous 'matching' algorithm, with a few modifications. This algorithm was also later altered to use k-Nearest Neighbour when finding similar cases, further improving it's accuracy. As before, the number of matching AUs in the two cases' AU vectors was calculated, and compared with their respective lengths to obtain a 'percentage' similarity measure using the following formula:

$$P = \frac{matches}{length(new\ case\ AU\ vector)}$$

$$Q = \frac{matches}{length(existing\ case\ AU\ vector)}$$

$$Similarity = \frac{P+Q}{2}$$

If $Similarity = 1$, then the cases are exactly the same and this case is immediately returned. Otherwise, this similarity measure is stored in the list of nearest neighbours if it has a higher $Similarity$ than the current lowest ranked nearest nighbour. At first, only the branch of the CBR system containing cases of the same length as the new case is iterated over to produce the list of neighbours, but if no neighbours are found with a similarity above a certain threshold (0.9) then all other branches are also checked for similarities. When a suitable list of k neighbours has been obtained, the 'nearest' label is calculated by comparing the similarities of neighbours in the list modified by their typicality to obtain an 'average' label according to this formula, where N is the array of nearest neighbours:

$$Label_{n,\ n\in(1,6)} = \sum_{i=1}^{k} N_i.similarity^4 \ * \ N_i.typicality,\ where\ N_i.label = n$$

The value of n which has the highest $Label_n$ is then decided to be the most likely and the label corresponding to that value of n (1 for Anger etc.) is returned. If two values of n are tied for nearest value, the first (lowest) value of n is selected. A random choice was originally implemented here, but it was found to make no difference to the results so the extra calculation was removed. The average results for these various algorithms over 50 runs can be seen below:

Algorithm	Average F_1 (50 runs)
Random	0.1479
Length Comparison	0.1522
Matches	0.8766
Improved Matches (single nearest)	0.9493
Improved Matches (k-NN, k = 5)	0.9589

There are several reasons why this retrieval algorithm works best. It is based on the number of matching AUs between two cases, which appears to be the best metric for determining similarity. However, this algorithm also attempts to calculate how significant the matches found are by basing similarity on the proportion of AUs in the vectors that the matches represent. Also, by using k-NN instead of single nearest it is robust to anomalous results due to noisy data and inaccurate previous classifications which have been added to the case-base. Although this should have been of lesser importance to the implemented system as it was generated and tested using 'clean' data, it does appear to have had a small effect of the overall accuracy of the system.

Case Reuse

Case reuse is fairly simple - most of the complexity is instead contained in 'retain.m'. The value of the similar case label is assigned to the new cases' label and the re-labeled new case is returned.

Case Retention

Cases are retained by appending them to the end of the branch of the CBR system which contains cases of their length. Before being added, however, the case is compared to all other cases in the list to check if it is the same as a case which is already present. If it has the exact same active AUs and label as a case which is already in the system, it is instead discarded and the 'typicality' value of the matching case is increased by one. If it has the same AU vector but a different label, it is discarded and the matching case in the system has it's typicality value decreased by one. If this brings it's typicality to zero, it is also discarded from the system. In this way, if the first example of a specific case was incorrectly labeled (due to incorrect training data, etc.) then a correctly labeled example of that case could cancel it out and all further correct examples of that case would be inserted into the system as normal. This also means that if new training data were obtained in which a specific case was now classifed as a different emotion to it's classifiction when the system was originally trained, by continuing to train the system with this new data it could 'learn' that the classification of that case had changed. This feature was tested on the noisy data set as no examples of this could be found in the clean set, and the system maintained a good accuracy rating in excess of 0.9 despite the noisy data.

Additional Issues

Once the system was achieving a consistently high level of accuracy, it was optimised to reduce time taken for retrieval. As mentioned above, this prompted a redesign of the case-base structure into a branched structure and the introduction of the typicality statistic for each case. Immediately after making these changes, the system was tested with new cases only being compared to cases with the same AU vector length, but this significantly worsened results at the expense of making large time savings. To remedy this, it was decided that if no neighbours were found for a new case with a similarity above a certain threshold then all branches would be examined. This threshold was eventually fixed at 0.9 for the data set tested on, as this resulted in some time savings with no noticable degradation of accuracy. Time taken to perform 30 runs of the 10-fold cross-validation script for various iterations of the system can be seen below:

System	Time taken (s)
Flat List (unsorted)	41.158
List (sorted by typicality)	40.318
Length based branches	20.304
Branches (with threshold 0.9)	19.005

2 Results

These are the results of a typical run of the final CBR system:

CBR System Precision/Recall measures per fold

Emotion	TP	TN	FP	FN	Recall	Precision	F_1
Anger	1	9	0	0	1.000	1.000	1.000
Disgust	0	10	0	0	1.000	1.000	1.000
Fear	0	10	0	0	1.000	1.000	1.000
Happyness	4	6	0	0	1.000	1.000	1.000
Sadness	1	9	0	0	1.000	1.000	1.000
Surprise	4	6	0	0	1.000	1.000	1.000

Emotion	TP	TN	FP	FN	Recall	Precision	F_1
Anger	0	10	0	0	1.000	1.000	1.000
Disgust	1	9	0	0	1.000	1.000	1.000
Fear	1	9	0	0	1.000	1.000	1.000
Happyness	4	6	0	0	1.000	1.000	1.000
Sadness	3	7	0	0	1.000	1.000	1.000
Surprise	1	9	0	0	1.000	1.000	1.000

Emotion	TP	TN	FP	FN	Recall	Precision	F_1
Anger	2	8	0	0	1.000	1.000	1.000
Disgust	4	6	0	0	1.000	1.000	1.000
Fear	2	8	0	0	1.000	1.000	1.000
Happyness	0	10	0	0	1.000	1.000	1.000
Sadness	1	9	0	0	1.000	1.000	1.000
Surprise	1	9	0	0	1.000	1.000	1.000

Emotion	TP	TN	FP	FN	Recall	Precision	F_1
Anger	3	6	0	1	1.000	0.750	0.857
Disgust	2	8	0	0	1.000	1.000	1.000
Fear	0	9	1	0	0.000	0.000	0.000
Happyness	1	9	0	0	1.000	1.000	1.000
Sadness	0	10	0	0	1.000	1.000	1.000
Surprise	3	7	0	0	1.000	1.000	1.000

Emotion	TP	TN	FP	FN	Recall	Precision	F_1
Anger	2	8	0	0	1.000	1.000	1.000
Disgust	2	8	0	0	1.000	1.000	1.000
Fear	0	10	0	0	1.000	1.000	1.000
Happyness	3	7	0	0	1.000	1.000	1.000
Sadness	1	9	0	0	1.000	1.000	1.000
Surprise	2	8	0	0	1.000	1.000	1.000

Emotion	TP	TN	FP	FN	Recall	Precision	F_1
Anger	1	9	0	0	1.000	1.000	1.000
Disgust	0	10	0	0	1.000	1.000	1.000
Fear	1	9	0	0	1.000	1.000	1.000
Happyness	3	7	0	0	1.000	1.000	1.000
Sadness	1	9	0	0	1.000	1.000	1.000
Surprise	4	6	0	0	1.000	1.000	1.000

Emotion	TP	TN	FP	FN	Recall	Precision	F_1
Anger	1	9	0	0	1.000	1.000	1.000
Disgust	3	7	0	0	1.000	1.000	1.000
Fear	1	9	0	0	1.000	1.000	1.000
Happyness	2	8	0	0	1.000	1.000	1.000
Sadness	0	10	0	0	1.000	1.000	1.000
Surprise	3	7	0	0	1.000	1.000	1.000

Emotion	TP	TN	FP	FN	Recall	Precision	F_1
Anger	1	9	0	0	1.000	1.000	1.000
Disgust	4	6	0	0	1.000	1.000	1.000
Fear	2	8	0	0	1.000	1.000	1.000
Happyness	1	9	0	0	1.000	1.000	1.000
Sadness	2	8	0	0	1.000	1.000	1.000
Surprise	0	10	0	0	1.000	1.000	1.000

Emotion	TP	TN	FP	FN	Recall	Precision	F_1
Anger	0	10	0	0	1.000	1.000	1.000
Disgust	2	7	0	1	1.000	0.667	0.800
Fear	0	9	1	0	0.000	0.000	0.000
Happyness	4	6	0	0	1.000	1.000	1.000
Sadness	0	10	0	0	1.000	1.000	1.000
Surprise	3	7	0	0	1.000	1.000	1.000

Emotion	TP	TN	FP	FN	Recall	Precision	F_1
Anger	0	10	0	0	1.000	1.000	1.000
Disgust	3	7	0	0	1.000	1.000	1.000
Fear	0	10	0	0	1.000	1.000	1.000
Happyness	2	8	0	0	1.000	1.000	1.000
Sadness	3	7	0	0	1.000	1.000	1.000
Surprise	2	8	0	0	1.000	1.000	1.000

CBR System Average Measures

Precision/Recall Table (all folds)

Emotion	TP	TN	FP	FN	Recall	Precision	F_1
Anger	11	88	0	1	1.000	0.917	0.957
Disgust	21	78	0	1	1.000	0.955	0.977
Fear	7	91	2	0	0.778	1.000	0.875
Happyness	24	76	0	0	1.000	1.000	1.000
Sadness	12	88	0	0	1.000	1.000	1.000
Surprise	23	77	0	0	1.000	1.000	1.000

Average F_1: 0.9680

Confusion Matrix (all folds)

	Anger	Disgust	Fear	Happiness	Sadness	Surprise
Anger	11	0	1	0	0	0
Disgust	0	21	1	0	0	0
Fear	0	0	7	0	0	0
Happiness	0	0	0	24	0	0
Sadness	0	0	0	0	12	0
Surprise	0	0	0	0	0	23