Machine Learning CBC - Assignment 4: Case Based Reasoning

Chris Carter (cc1808), Mark Law (ml1909), Michael Thorpe (mt2309) and Fraser Waters (fjw08)
(Group **32**)

Implementation Details

The case structure is a list of AU values and the typicality of that set. In our structures, the typicality is defined as the number of times the set of AUs has been classified as each emotion. Initially, our structure was defined as having a single emotion attribute and a corresponding typicality value which represented the number of times those AUs were classified as that emotion, however due to noise we could be presented with two cases with identical AU sets and a different solution. To overcome this, the typicality is instead a vector, showing the count for each emotion.

The CBR dataset is simply an unsorted list of cases. We considered implementing the CBR dataset as a binary tree, however this was unsuitable for two reasons:

- The binary tree would sort the cases that was placed into it, however this sorting would be completely unnecessary.
- When trying to find the best case using the retrieve function, each case in the CBR dataset needs to be inspected. This would mean that the tree would effectively be flattened to a list, and therefore making it less efficient than using a list to begin with.

RETRIEVE, REUSE, and RETAIN

The retrieveCases function first builds a list of cases sorted by the “distance”, using a given metric. This metric compares the AU vector for each of the cases in the system with the new case we at trying to classify. The function then returns a list of the cases with the lowest metric retrieved.

If there is a case which has exactly the same AUs as the search case the algorithm returns this case.
If no such case can be found then the algorithm selects the first k - 1 items in the list. It sums the typicality vectors of all the cases which are the same distance away from the search case and takes the maximal emotion. It then selects the case which has the highest typicality for this emotion and uses this as case k. It returns cases 1 to k.

The retrieve function first calls retrieveCases and calculates the number of cases returned (n). For each of the cases it multiplies the typicality by n - the rank of the case (by distance) and takes the sum. It chooses the emotion which has the highest value in this typicality vector (choosing randomly between equal emotions). It returns a case with this emotion.

The reuse function applies a classification to the new case by giving it the emotion of the case which is most similar to it. This function bases its emotion choice on the output from the retrieve function.

The retain function adds the solved case to the CBR. This method is also used when creating the CBR with the initial training data. If I case already exists in the CBR system i.e they have the same set of AUs, then the typicalities are added together. This ensures that there is only one entry in the CBR system with a set of AU values.

Similarity Measure

We have experimented with three different similarity measures, which is used when comparing two cases. This gives an indication of the number of changes that need to be made to the AUs of one case in order to be identical to another. Firstly, each AU list is converted to a 45 bit-array, so that they are all the same length and can be compared with simple arithmetic and binary operators efficiency. The measures we use are explained below:

- Comparison Method 1 - Average number elements in each minus the number of common elements.
- Comparison Method 2 - The Manhattan distance squared minus the number of common elements squared. This method gives a higher weight to the square of the Manhattan distance. The reason for squaring is to make part of this sum more significant if it has a higher value.
- Comparison Method 3 - Absolute difference (Manhattan distance) between the two bit-arrays - This tends to result in many cases with the same measure value.

Shown below are three graphs which represent the value of k against the average f1 measure over the six emotions. This has been done for each of the comparison methods shown above, and is used to find the ideal value for k.

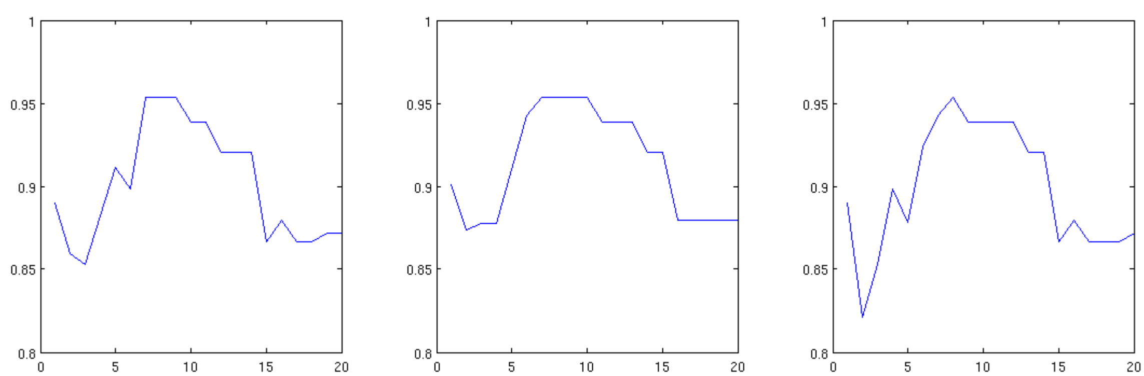

Figure 1: Value of k (x-axes) against average f1 measure (y-axes) for comparison method 1 (left), method 2 (centre) and 3 (right)

As the peaks for each graph are at the same height, we have chosen to use metric 2 because the maximum of the graph covers the longest distance. This means that we can be more certain of the data falling onto the peak if we were to add more data to our dataset. We have chosen k to be 7, as it is in the middle of this peak, meaning that if there is a shift in the data, it can shift either way and we will not lose accuracy.

Results for k = 7, using Comparison Metric 2

	Anger (1)	Disgust (2)	Fear (3)	Happiness (4)	Sadness (5)	Surprise (6)
Anger (1)	10	0	0	0	0	0
Disgust (2)	0	21	0	0	0	0
Fear (3)	1	1	7	0	0	0
Happiness (4)	0	0	0	24	0	0
Sadness (5)	1	0	0	0	12	0
Surprise (6)	0	0	0	0	0	23

	Anger (1)	Disgust (2)	Fear (3)	Happiness (4)	Sadness (5)	Surprise (6)
Recall	0.8333	0.9545	1.0000	1.0000	1.0000	1.0000
Precision	1.0000	1.0000	0.7778	1.0000	0.9231	1.0000
F_1 measure	0.9091	0.9767	0.8750	1.0000	0.9600	1.0000