Medoid Classification Algorithm (MCA)

Brian T. Luke (lukeb@ncifcrf.gov)

MCA is the author's approximation to the classification procedure used in several studies by the laboratories of Petricoin and Liotta [Bro-05, Con-04, Orn-04, Pet-05, Sri-06, Sto-05]. While their algorithm used a genetic algorithm driver to search for an optimum set of features, allowing for different putative solutions to use different numbers of features (5-20 features), our algorithm uses a mEP feature selection algorithm and all putative solutions have the same number of features *N*. For a given value of *N*, *N* features were selected and the intensities of these features were rescaled for each individual using the following formula [Bro-05, Con-04, Orn-04, Pet-05, Sri-06, Sto-05]:

$$I' = (I - Imin) / (Imax - Imin)$$

In this equation, I is a feature's original intensity, I' is it's scaled intensity, and Imin and Imax are the minimum and maximum intensities found for the individual among the *N* selected features, respectively. If Imin and Imax were from the same features in all samples, a baseline intensity would be subtracted and the remaining values scaled so that the largest intensity was 1.0. Each individual would then be represented as a point in an (*n*-2)-dimensional unit cube. As designed, and as found in practice, Imin and Imax do not represent the same features from one individual to the next, so this interpretation does not hold. Therefore, each individual represents a point in an *N*-dimensional unit cube.

The first training sample becomes the medoid of the first cell, with this cell being classified as the category of this sample. Each cell has a constant trust radius *r*, which is set to 0.1 $(N)^{1/2}$, or ten percent of the maximum theoretical separation in this unit hypercube. If the second sample is within *r* of the first, it is placed in the first cell; otherwise it becomes the medoid of the second cell and that cell is characterized by the second sample's category. This iteration continues until all training samples are processed. Each cell is then examined and the categories of all samples in the cell are compared to the cell's classification. This calculation allows a sensitivity and specificity to be determined for the training data, and their sum represents the quality score for this set of *N* features.

The mEP algorithm initially selects *Npop* parent sets of *N* randomly selected features. The only caveat is that each set of *N* features must be different from all previously selected sets. The medoid classification algorithm then determines the score for each set of features. Each parent set of features generates an offspring set of features by randomly removing one or two of the features and replacing them with randomly selected features, requiring that this set be different from all feature sets in the parent population and in all offspring generated so far. The score of this feature set is determined and the score and feature set is stored in the offspring population. After all *Npop* offspring have been generated the parent and offspring populations are combined. The *Npop* feature sets with the best score are retained and become the parents for the next generation, which is known as an *elitist strategy*.

It should be noted that for a set of *N* features, the number of unique cells that can be generated is on the order of 10^N. Since no training set is ever this large (*N* is 5 or more), only a small fraction of the possible cells will be populated and classified. As was previously shown [Luk07], this limitation causes a significant number of the testing samples to be placed in an unclassified cell, though none of the publications that used this method [Bro-05, Con-04, Orn-04, Pet-05, Sri-06, Sto-05] reported an undetermined classification for any of the testing samples. Instead of searching through a large number of solutions that classified the training samples to a significant extent and find those that minimized the number of unclassified testing samples, all samples are used and a limit is placed on the number of cells. If there are *Ncase* Case samples and *Ncontrol* Control samples, the maximum number of allowed Case and Control cells is 2(*Ncase*)/3 and 2(*Ncontrol*)/3, respectively. If any set of *N* features produced more than this number of Case or Control cells, this set of features is assigned a quality of zero. This allows one-third of the Case and Control samples to effectively represent the testing set without reducing the required coverage of the fingerprint. This restriction also allowed for a test of the dependence of the result on the order of the samples. In general, the MCA procedure is run more than once for each value of *N* and the seed to the random number generator and order of the samples is different for each run.

While this algorithm has been described as being quite similar to a Self-Organizing Map (SOM), the algorithm employed by the groups of Petricoin and Liotta [Bro-05, Con-04, Orn-04, Pet-05, Sri-06, Sto-05], has virtually nothing in common with a SOM. In a SOM [Koh-88], the layout of the cells is determined *a priori*, as are the number of features, *N*, used in the separation. In general, the cells are placed in a rectangular or hexagonal pattern, with a maximum of four or six adjacent cells, respectively. The cells are seeded with random centroids that represent the *N*-dimensional coordinates of each cell. The first training sample is assigned to the cell with the closest centroid and the centroids of this and all the other cells are significantly shifted towards this sample. This procedure is repeated for all samples. Once all samples have been processed, the algorithm repetitively cycles through the list of training samples. In each subsequent cycle, after a sample is assigned to a cell the extent to which this cell's centroid is shifted towards that sample decreases, as does the extent to which the other centroids are affected. This shift becomes significantly smaller for cells that are further from the selected cell, as defined by the initial mapping. When finished, all samples are assigned to cells and each centroid represents an approximate average of the *N* features for all samples in that cell, and the distance between centroids increases as the cells become further apart in the pre-defined map.

Therefore, a SOM has a fixed number of cells, each cell is described by a centroid, and the algorithm cycles through the training data many times to adjust the centroid's coordinates. The algorithm used by the groups of Petricoin and Liotta [Bro-05, Con-04, Orn-04, Pet-05, Sri-06, Sto-05] has an undefined number of cells, each described by a single sample (i.e. a medoid instead of a centroid), and the training data is only processed once.