On-line fMRI Data Classification Using Linear and Ensemble Classifiers

Catrin O. Plumpton, Ludmilla I. Kuncheva,
School of Computer Science, Bangor University, LL57 1UT, United Kingdom
{*c.o.plumpton,l.i.kuncheva*}*@bangor.ac.uk*

David E. J. Linden and Stephen J. Johnston
School of Psychology, Bangor University, LL57 2AS, United Kingdom
{*d.linden,s.johnston@bangor.ac.uk*}*@bangor.ac.uk*

Abstract

The advent of real-time fMRI pattern classification opens many avenues for interactive self-regulation where the brain's response is better modelled by multivariate, rather than univariate techniques. Here we test three on-line linear classifiers, applied to a real fMRI dataset, collected as part of an experiment on the cortical response to emotional stimuli. We propose a random subspace ensemble as a fast and more accurate alternative to component classifiers. The on-line linear discriminant classifier (O-LDC) was found to be a better base classifier than the on-line versions of the perceptron and the balanced winnow.

1. Introduction

Classification of functional magnetic resonance imaging (fMRI) data has allowed the neuroimaging community to uncover discriminating patterns of neural activity that define independent 'thought processes'. This approach has led to an increase in the understanding of the organisation of the functional architecture of the human brain. However, due to the large number of voxels in a typical fMRI scan, the classifier is presented with a massive feature set. Coupled with a relatively small training sample, fMRI classification has a challenging feature-to-instance ratio in the order of 5000:1.

Various classifier models have been used for fMRI classification. Linear classifiers are popular due to their speed and accuracy, including the Support Vector Machine (SVM) classifier with linear kernel [11]. Classifier ensembles are deemed to be more accurate than individual classifiers [5]. The Random Subspace ensemble (RS) is a popular classifier ensemble whereby ensemble members are trained on feature subsets rather than the entire feature set [2]. The ensemble decision is based on majority voting. RS ensembles are particularly suitable for datasets with a large feature-to-instance ratio as they reduce the dimensionality of the feature set and create diversity while retaining the number of instances for training. It has been shown that RS ensembles work well for fMRI data [8, 9].

While off-line fMRI analysis provides valuable insights into the neural activity associated with discrete tasks, many approaches would benefit from fast and accurate on-line classification. Real-time feedback allows for intervention and correction in cases of technical failure, poor task performance or excessive motion during the experiment [14]. Real time feedback also creates opportunities for experiments involving self-regulation and neurofeedback via a brain-computer interface [3, 14].

Real time classification of fMRI data poses new challenges. The number of training instances is further reduced and the classifier must be capable of working within a tight time constraint. The SVM remains a popular choice of classifier for real-time classification of fMRI data [10]. Closely related to the SVM is the Relevance Vector Machine (RVM), which is also popular for real time fMRI classification [3, 4]. For real-time classification, the classifier is required to work on-line, processing data points sequentially, analysing one fMRI brain scan at a time.

On-line classifiers are required to exhibit any time learning and not demand further memory as time progresses. Alongside SVM and RVM, simpler linear classifiers are capable of providing fast and accurate results.

We consider three on-line linear classifier models: the on-line linear discriminant classifier (O-LDC) [6], Rosenblatt's perceptron and the balanced winnow [12]. We propose to use RS ensembles for the three classifiers and compare the performance of individual and the ensemble classifiers on a real fMRI data set.

2. On-line linear classifiers

In on-line classification every data point is classified as it becomes available, and its true class label is recovered immediately after that. The classifier is updated by adding this point to the training set, and recalculating the parameters. Here we use error-driven versions of the three on-line linear classifiers: perceptron, balanced winnow and O-LDC. This means that the coefficients of the linear functions are only updated if the incoming data point is misclassified by the current classifier.

Perceptron. The perceptron first initialises coefficients $\mathbf{w} = [w_0, \ldots, w_n]^T$ as small random numbers. A learning parameter η is also defined. The learning parameter corresponds to the 'readiness to learn' of the algorithm, and defines the weighting of new data points compared to past data. Assuming N data points have been presented to the classifier, denote the next data point as $\mathbf{x}_{N+1} \in \Re^n$ with true label y_{N+1}, unavailable at the time of classification. The data point is augmented, $\mathbf{z} = [1 \quad \mathbf{x}_{N+1}^T]^T$, where the first element, 1, multiplies the bias coefficient w_0. The data point is then classified by the 'current' classifier. The predicted label ($+1$ or -1) for $\mathbf{x}_{N+1}$ is calculated by $y_{\text{predicted}} = \text{sign}\left(\mathbf{z}^{\text{T}}\mathbf{w}\right)$, where $\text{sign}(a) = 1$ if $a \geq 0$ and $\text{sign}(a) = -1$ if $a < 0$. If the data point is misclassified, that is, $y_{\text{predicted}} \neq y_{N+1}$, then the weight vector is updated as $\mathbf{w} \leftarrow \mathbf{w} - \eta\ \mathbf{z}\ y_{\text{predicted}}$.

Balanced Winnow. The balanced winnow is similar in design to the perceptron, however it has two sets of weights, a positive set $\mathbf{w}^+$ and a negative set $\mathbf{w}^-$. Both sets of weights are initialised as positive random numbers, and a learning rate β is chosen. The predicted label for $\mathbf{x}_{N+1}$ is $y_{\text{predicted}} = \text{sign}\left(\mathbf{z}^{\text{T}}\ (\mathbf{w}^+ - \mathbf{w}^-)\right)$. Following a misclassification, the $n+1$ weights of the balanced winnow are updated by $w_i^+ \leftarrow \beta^{-(y_{N+1})\times z_i} w_i^+$ and $w_i^- \leftarrow \beta^{(y_{N+1})\times z_i} w_i^-$, $i = 0, 1, \ldots, n$.

On-line linear discriminant classifier. The on-line linear discriminant classifier (O-LDC) is an adaptation of the linear discriminant classifier. Denote by $P^{(i)}$ the prior probabilities for class i, $\mu(i)$ and Σ are the mean and covariance matrix. The discriminant functions $g_i(\mathbf{x})$ are calculated as

$$g_i(\mathbf{x}) = \ln P^{(i)} - \frac{1}{2}\mu(i)^T\Sigma^{-1}\mu(i) + \mu(i)^T\Sigma^{-1}\mathbf{x},$$

and $\mathbf{x}$ is assigned the label corresponding to the largest $g_i(\mathbf{x})$. For the on-line version of this classifier, the means and *inverse* covariance matrix require updating after each data point.

Let c be the total number of classes and $\mathbf{m}_{N_i}^{(i)}$ be the estimate of the mean for class i, where N_i is the number of points from class i thus far. The update for the mean of class k is calculated as

$$\mathbf{m}_{N_k+1}^{(k)} = \frac{1}{N_k+1}\left(N_k\mathbf{m}_{N_k}^{(k)} + \mathbf{x}_{N+1}\right).$$

The inverse covariance matrix for class k is updated as

$$S_{N+1}^{-1} = \frac{N+1}{N}\left(S_N^{-1} - \frac{S_N^{-1}\ \mathbf{z}\ \mathbf{z}^T\ S_N^{-1}}{\frac{N(N_k+1)}{N_k} + \mathbf{z}^T\ S_N^{-1}\ \mathbf{z}}\right),$$

where $\mathbf{z} = \mathbf{x} - \mathbf{m}_{N_k+1}^{(i)}$. The prior probabilities estimated as $P_N^{(i)} = N_i/N$ are also updated [1, 6].

Random subspace ensembles. In general, when performing classification, the more features that are available, the better the resulting classifier. It is however possible to 'over fit' the classifier on the training set. Features may also be irrelevant or redundant, offering very little to the classification. The high feature-to-instance ratio of fMRI data emphasises this problem.

A good ensemble should be made up of *diverse* classifiers. The Random Subspace method generates diverse classifiers by training each ensemble member on a different feature subset. Define $\mathbf{X} = [x_1, \ldots, x_n]^T$ to be the set of n features (voxels). To create an RS ensemble, we randomly select L feature subsets of size M by drawing without replacement from a uniform distribution over $\mathbf{X}$. These subsets make up the feature sets for the L classifiers. Each of the L classifiers are trained on the respective M features and a final ensemble decision is made by majority vote.

There are many benefits to RS ensembles for fMRI data. Reducing the number of features per classifier reduces the likelihood of over fitting. Also, the algorithm is computationally inexpensive due to the reduced number of features per ensemble member. RS ensembles have been shown to perform well for off-line fMRI data [8, 9], however they have not yet been applied to streaming fMRI data.

3. Material and methods

We use a dataset collected at the School of Psychology, Bangor University, which forms part of a study on perception of emotional pictures. The subject was presented with a series of 'emotionally charged' images in a block type design. Each block consisted of pictures corresponding to a single emotional valence type, either positive, negative or neutral. The resulting data set consists of 204 volume images taken at time steps (TRs) of 1.5s. To reduce the number of irrelevant features we apply a grey matter voxel mask to each functional image, reducing the number of voxels (features) from 83072 to 33274. As an additional preprocessing step advocated in fMRI literature, we apply a t-test and select the 2000 'most relevant' voxels according to the p-values. The data is then split into two sets. The 'training' data T consists of those TRs taken from the first positive and negative blocks of images. TRs corresponding to neutral images or the fixation period are not included in T. The streaming data used for on-line training comprises of all TRs from the end of T onwards.

For each base classifier (perceptron, balanced winnow and O-LDC) we train an off-line (batch) version on T, using class labels 1 and 2 to correspond to negative and positive stimuli respectively. This may correspond with an initialisation period in an fMRI experiment. After this initial training period, we present the online data points one at a time. The 'current' classifier is tested on the data point; if the data point is misclassified then the classifier is updated.The experiment is repeated using classifier ensembles rather than individual classifiers. For the classifier ensembles we choose $L = 11$ and $M = 1000$ (based on the $M = n/2$ recommendation [7]).

Kappa-error diagrams are now an accepted tool for comparing classifier ensembles [13]. The x-axis of the diagram is the diversity of the ensemble, κ. Lower values of κ indicate more diverse ensembles. The y-axis shows the individual error rates of the classifiers. Each pair of classifiers in the ensemble generates one point on the diagram. Ensembles whose 'clouds' of points are situated closer to the bottom left corner of the diagram are usually more accurate.

4. Results

We calculate the cumulative error at each TR. These error progressions for the individual classifiers and RS ensembles are illustrated in Figure 1 (a) and (b) respectively.

The O-LDC outperforms the other classifiers both individually and as an ensemble . The perceptron comes second whilst the balanced winnow yields the worst results. The error progression for the perceptron appears to start at zero as the perceptron correctly classifies the intial TRs, but accuracy drops over time. The final cumulative errors are summarised in Table 1. The RS ensembles all perform better than their individual counterparts, with the O-LDC performing better than either the perceptron or balanced winnow, both individually, and as an ensemble.

Table 1. Final errors for individual classifiers and classifier ensembles.

	LDC	Perceptron	Winnow
Individual classifier	0.0929	0.1214	0.1786
Classifier ensemble	0.0571	0.0857	0.1714

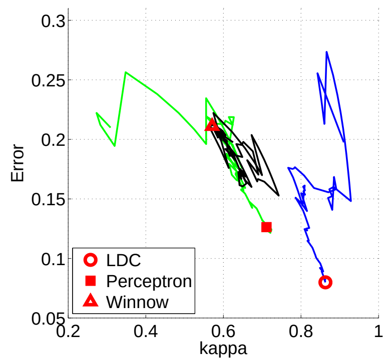


Figure 2. Trajectory of means of kappa error diagrams.

For each TR we calculate the kappa error diagrams for the three ensembles. To demonstrate how the kappa-error diagrams progress with time, we calculate the mean of the three clouds at each TR. Figure 2 plots the trajectories of these means. The endpoint is indicated with a marker. The initial high diversity of the perceptron may explain its early accuracy. This diversity decreases over time. The trajectory of the O-LDC ensemble shows improving diversity and accuracy over time. For the individual O-LDC and O-LDC ensemble we were interested to find *when* the errors occurred. Figure 3 shows the 'design matrix' which is the valence of the stimuli; peaks correspond to positive emotion and

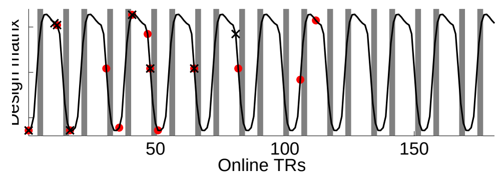
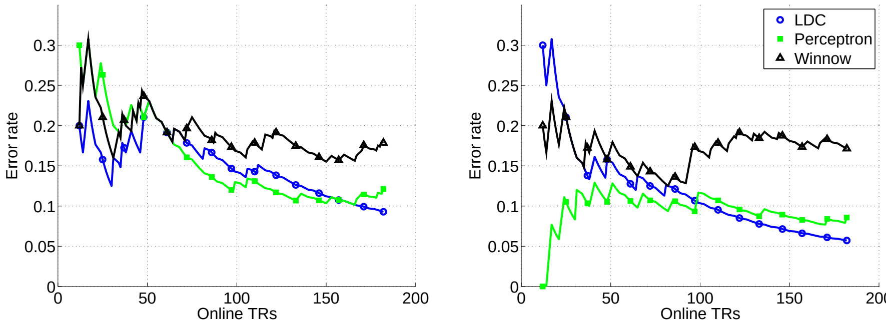


Figure 3. Design matrix highlighting occurrence of individual and ensemble errors.



(a) individual classifiers (b) ensemble classifiers

Figure 1. Cumulative error progression.

valleys to negative emotion. Neutral and fixation TRs are marked by grey vertical stripes. The class labels for the TRs in these stripes are undetermined (neither positive nor negative but a transition between the two states). Errors by the individual classifier are marked by red dots whilst black x's mark errors made by the ensemble. For both classifiers errors are predominantly made in the first half of the experiment. This shows how the classifiers improve over time. Fewer errors occur at maximum valence (peaks and valleys); those that do occur are early in training. It can be seen that ensemble errors occur less frequently and stop earlier than errors arising from the use of the individual O-LDC.

5. Conclusion

Our experiments show that the random subspace ensemble performs better than the individual online linear classifiers. Across both the individual and ensemble experiments, the on-line linear discriminant classifier (O-LDC) outperformed both the perceptron and balanced winnow. As a linear classifier, the O-LDC is fast to train and has demonstrated accurate results making it the best choice, of the methods tested here, for use in real-time pattern classification studies of the human brain. Future work includes comparing the results of the O-LDC ensemble with other classifier ensembles for fMRI data.

References

[1] R. O. Duda, P. E. Hart, D. G. Stork. *Pattern Classification*. Wiley-Interscience, 2000.

[2] T. K. Ho. The random space method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20 (8):832 – 844, 1998.

[3] M. Hollmann, T. Mönch, S. Mulla-Osman, C. Tempelmann, J. Stadler, and J. Bernarding. A new concept of a unified parameter management, experiment control, and data analysis in fMRI: Application to real-time fMRI at 3t and 7t. *Journal of Neuroscience Methods*, 175:154–162, 2008.

[4] M. Hollmann, T. Mönch, C. Muller, and J. Bernarding. Predicting human decisions in socioeconomic interaction using real-time functional magnetic resonance imaging (rtfMRI). In *SPIE-Medical Imaging*, 2009.

[5] L. I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience, 2004.

[6] L. I. Kuncheva and C. O. Plumpton. Adaptive learning rate for online linear discriminant classifiers. In *Proc. S+SSPR*, pages 510–519, Orlando, Florida, USA, 2008.

[7] L. I. Kuncheva and C. O. Plumpton. Choosing parameters for random subspace ensembles for fMRI classification. In *Proc. Multiple Classifier Systems*, 2010.

[8] L. I. Kuncheva and J. J. Rodríguez. Classifier ensembles for fMRI data analysis: An experiment. Magnetic Resonance Imaging (to appear).

[9] L. I. Kuncheva, J. J. Rodríguez, C. O. Plumpton, D. E. J. Linden, and S. J. Johnston. Random subspace ensembles for fMRI classification. *IEEE Transaction on Medical Imaging*. (in press).

[10] S. M. LaConte, S. J. Peltier, and X. P. Hu. Real-time fMRI using brain-state classification. *Human Brain Mapping*, 28:1033–1044, 2007.

[11] S. M. LaConte, S. Strother, V. Cherkassky, J. Anderson, and X. Hu. Support vector machines for temporal classification of block design fMRI data. *NeuroImage*, 26(2):317 – 329, 2005.

[12] N. Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2(4):285–318, April 1988.

[13] D. Margineantu, and T. G. Dietterich. Pruning adaptive boosting. In *Proc. of the Fourteenth International Conference on Machine Learning*, San Francisco, 1997.

[14] N. Weiskopf, R. Sitaram, O. Josephs, R. Veit, F. Scharnowski, R. Goebel, N. Birbaumer, R. Deichmann, and K. Mathiak. Real-time functional magnetic resonance imaging: methods and applications. *Magnetic Resonance Imaging*, 25:989–1003, 2007.