Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Naive random subspace ensemble with linear classifiers for real-time classification of fMRI data

Catrin O. Plumpton [a,*], Ludmila I. Kuncheva [a], Nikolaas N. Oosterhof [b], Stephen J. Johnston [c]

[a] School of Computer Science, Bangor University, Bangor, Gwynedd LL57 1UT, United Kingdom
[b] School of Psychology, Bangor University, Bangor, Gwynedd LL57 2AS, United Kingdom
[c] Psychology Department, Brunel University, Uxbridge, Middlesex UB8 3PH, United Kingdom

ARTICLE INFO

Available online 10 May 2011

Keywords:
Functional magnetic resonance imaging (fMRI)
Online classification
Naive labelling
Classifier ensembles

ABSTRACT

Functional magnetic resonance imaging (fMRI) provides a spatially accurate measure of brain activity. Real-time classification allows the use of fMRI in neurofeedback experiments. With limited labelled data available, a fixed pre-trained classifier may be inaccurate. We propose that streaming fMRI data may be classified using a classifier ensemble which is updated through *naive labelling*. Naive labelling is a protocol where in the absence of ground truth, updates are carried out using the label assigned by the classifier. We perform experiments on three fMRI datasets to demonstrate that naive labelling is able to improve upon a pre-trained initial classifier.

1. Introduction

In recent times, data acquired by functional magnetic resonance imaging (fMRI) have allowed for valuable insights into the human mind, and into those processes which control and reflect human behaviour.

While most fMRI approaches analyse the data off-line (after scanning has been completed), more recently there has been interest in the development and application of real-time fMRI (rtfMRI) [1].

One application is neurofeedback, where a participant performs a certain task while brain activity is measured, and feedback based on this activity is given in real time. In this way, the participant may learn to exercise self-control of specific brain regions, for example those involved in pain perception [5]. This is typically achieved via a closed loop of brain computer interface (BCI) [9,34,4].

As multivariate approaches have been shown to be more sensitive than univariate approaches for off-line fMRI analyses [30,13], it seems sensible to use the former for rtfMRI as well. The efficiency and precision of rtfMRI for brain control has been demonstrated by participants carrying out tasks such as navigating through computer-generated mazes [15], balancing a virtual inverted pendulum [7], predicting decisions in an economic game [16], and moving an arrow towards a target [25]. fMRI classification can be a fast and accurate component of the BCI loop for the purposes of neurofeedback. The neurofeedback loop is sketched in Fig. 1. The participant receives initial instructions and possibly some stimuli. Next, the participant's state of mind is measured and classified. Based on the measured brain state, feedback is given to the participant who may then attempt to adjust the brain state to improve task performance.

To provide an accurate classification of the brain state, the classifier should be properly trained. Given the limited amount of time to collect individual fMRI data for the subject partaking in the experiment, and the large ratio of features to instances, the initially trained classifier may be of insufficient accuracy. It is desirable that the classifier improves with time. Depending on the information available for the updating of the classifier, the online training may be done on labelled or unlabelled data. Previous works assumed that the classifier is not updated with time [15,7], or updated with labelled data [32]. In this study we are interested in the possibility of using unlabelled data to improve on the classification accuracy of the initially trained classifier, termed *semi-supervised learning* [29,33,35]. This is the most natural and hence useful scenario, because in certain experiments, there may be no way to verify the true state of the brain. We propose to use *naive labelling*, where the classifier is updated by adding the new data point to the training set and taking the *predicted* label as the true label. This approach should be taken with caution, guarding against the possibility of a runaway classifier that progressively learns 'the wrong thing' [3]. Our previous study suggests that simple classifiers may benefit from semi-supervised learning [20].

[*] Corresponding author.
E-mail addresses: c.o.plumpton@bangor.ac.uk, cat.plumpton@googlemail.com (C.O. Plumpton), l.i.kuncheva@bangor.ac.uk (L.I. Kuncheva), n.oosterhof@bangor.ac.uk (N.N. Oosterhof), stephen.johnston@brunel.ac.uk (S.J. Johnston).

0031-3203/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2011.04.023

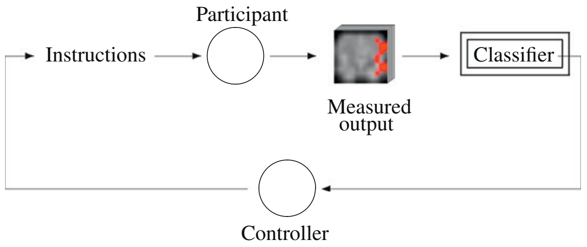

Fig. 1. The neurofeedback loop. Based on the classification output, the participant is instructed to perform a mental exercise that will drive the brain pattern closer to one corresponding to the desirable behaviour.

The rest of the paper is organised as follows. Section 2 discusses classifiers for real-time fMRI, including the random subspace (RS) ensemble, and introduces naive labelling. The dataset and methods are discussed in Section 3 with the results being presented in Section 4. Section 5 concludes the paper.

2. Real-time classification of fMRI data

2.1. Classifier models for fMRI

Various classifier models have been used for fMRI classification. Linear classifiers, including support vector machines (SVM) with linear kernel [26] and linear discriminant analysis (LDA) [18], are popular due to their speed and accuracy. Classifier ensembles are deemed to be more accurate than individual classifiers [19]. The random subspace ensemble (RS) is a classifier ensemble method whereby ensemble members are trained on feature subsets rather than on the entire feature set [14]. 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 [23,24].

In online 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. In the chosen scenario, the true class label is assumed to be the one predicted by the classifier. Applying real-time classification to fMRI neurofeedback presents 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 [25] and relevance vector machine (RVM) [16,15] are popular for real-time classification of fMRI data. Alongside, simpler linear classifiers are capable of providing fast and accurate results. In our previous work, [32], we compared three online linear classifier models for RS ensembles. The online linear discriminant classifier (O-LDC) [21] was shown to outperform Rosenblatt's perceptron and the balanced winnow [28], both as individual classifiers and in an ensemble. The question that we seek to answer in this study is whether O-LDC benefits from semi-supervised learning where the unlabelled data come as a sequence of brain volume images. We will also explore RS ensembles of O-LDC classifiers for the same problem.

2.2. Online linear discriminant classifier

The online linear discriminant classifier (O-LDC) is an adaptation of the linear discriminant classifier. It is chosen in this study because, in agreement with common wisdom [11], we found it to be robust and accurate compared to other linear classifiers for online fMRI (supervised) classification [32].

Let c be the total number of classes, and let $P^{(i)}$ be the prior probability for class i, $i=1,\dots,c$. Assuming that the data for each class come from a multivariate normal distribution with a class-specific mean, $\mu^{(i)}$, and a common covariance matrix, Σ, the optimal discriminant functions $g_i(\mathbf{x})$, $i=1,\dots,c$, are calculated as

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

The object $\mathbf{x}$ is assigned the label corresponding to the largest $g_i(\mathbf{x})$. In the online adaptation, the means and *inverse* covariance matrix are updated after each data point. Let $\mathbf{m}_{N_i}^{(i)}$ be the estimate of the mean for class i, where N_i is the number of points from class i thus far. The total number of points in the series is $N = N_1 + N_2 + \cdots + N_c$. Let S_N be the estimate of the common covariance matrix calculated from the N observations. Suppose that, after classification, the 'true' label of $\mathbf{x}_{N+1}$ is recovered as class k.[1] The recursive update for the mean of class k is calculated as

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

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. This adaptation is a version of the recursive least squares (RLS) method minimising the negative log-likelihood of the data using the class label as a latent variable. Like RLS, the O-LDC update is lossless. This means that the recursively calculated estimates of $\mathbf{m}_{N_k}^{(k)}$ and S_N coincide with these using all N data points received hitherto.

2.3. Random subspace ensembles

Classifier ensembles are less sensitive to noise and redundant features than single classifiers. The problems associated with overfitting are therefore less prevalent in classifier ensembles. This suggests that classifier ensembles are a good approach for datasets with a large feature-to-instance ratio, such as that typically found in fMRI data.

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 $\mathbf{X} = [x_1,\dots,x_n]^T$ to be the set of n features (voxels). To create a 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 overfitting. 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 [23,24], and in [32] we showed the feasibility of the RS ensemble as a technique for online classification of fMRI data. Here we take the challenge a step further. While previously it was assumed that the *true* labels became available straight after the classifier (or the ensemble) labelled the data point, here we

[1] Since we are interested in semi-supervised classification, we will take the label suggested by the classifier as the 'true' label.

assume that the true labels of the brain states are not available beyond the short training session prior to the main experiment.

2.4. Naive labelling

Without being able to pass the true class labels to the classifier, we have the choice of updating the classifier using the predicted *naive* labels, or using the fixed, pre-trained classifier throughout. This scenario is particularly relevant for neurofeedback experiments. A classifier which has been trained off-line on a small dataset will be likely to show a high error rate. Training the classifier using the naive labels does not come without risk. The classifier may be lead astray should updates occur using incorrect class labels. This may lead to 'runaway' behaviour where the classifier becomes less accurate the more data it sees [3].

The chances of avoiding runaway classifiers are related to the amount of off-line training data and on how well the underlying data distribution model is guessed when designing the classifier [20]. It is expected that the lower the amount of training data, the higher the chances of a runaway classifier appearing in the ensemble. Linear discriminant classifiers rely on a simple assumption of Gaussianity of the data, which is often not met in practice. However, even with assumptions not holding, linear classifiers have been found to be surprisingly accurate [11]. As noted in [18], LDA and linear SVM actually do better than non-linear classifiers, possibly because the latter are more prone to overfitting. While a runaway classifier is a realistic possibility if naive labelling is used, our hypothesis is that a sufficient number of classifiers *within the ensemble* will be improved beyond their off-line accuracy, and thus the ensemble will counteract any adverse effects on an individual ensemble member.

3. Material and methods

3.1. Datasets

We use three emotion based datasets which are described below. A summary of the datasets is given in Table 1.

Emotion_Negative (*EN*1 *and EN*2) *Data*: EN1 and EN2 are two runs with different participants in the same experiment. Participants were instructed to up-regulate their target region activity for periods of 20 s ('up'; 10 TR) using negative emotional imagery, alternating with baseline periods of 14 s ('rest'; 7 TR). There were 12 blocks of up-regulation and rest. Classification task is to distinguish between periods of emotion and periods of rest. This simple task is seen as a step towards classification of emotions, which is achieved with the third dataset, EB.

Emotion_Both (EB) *Data*: A sequence of fMRI brain scans was obtained from a single run. The participant viewed 12 blocks of images of positive valence type, 12 blocks of neutral valence and 12 blocks of negative valence. Each block of images lasted for a period of 6 s (four pictures presented for 1.5 s) followed by a period of fixation (12 s duration).[2] Fixation TRs are removed from the dataset. Classification task is to distinguish between positive, negative and neutral emotion.

Data for all three datasets were collected on a 3 Tesla Philips Achieva MR scanner (TR=2 s, TE=30 ms, 30 slices, in-plane resolution $2 \times 2\ \text{mm}^2$, 3 mm slice thickness). Slices were positioned such that the bottom slice was 30 mm ventral to the anterior commissure and angled to encompass all of the ventral prefrontal cortex.

Table 1
Summary of the three fMRI datasets.

Name	Volume size	# Voxels	Classes	# Instances
EN1	$60 \times 31 \times 44$	28426	2	203
EN2	$59 \times 32 \times 44$	28662	2	203
EB	$60 \times 62 \times 45$	29865	3	108

Preprocessing of the data was performed using Brainvoyager QX (Braininnovation, Maastricht, The Netherlands). The data were corrected for intra-subject angular and translational motion and filtered to remove long-term drift. It is noted that data were preprocessed off-line. While it has been shown that head movement correction can be performed in real-time [2] and the other steps could quite likely also be performed in real time, we stress that the results presented here do not consider the effects of preprocessing on classification results, which presents another challenge for applying this approach to real-time fMRI.

Class labels are calculated by using a box-car model offset from the stimuli by 1 TR. By doing this we maintain a simple model whilst accounting for the haemodynamic delay in the BOLD signal. In doing this, we lose one instance from each dataset, hence the figures in the final column of Table 1 not matching the number of TRs. For each data a voxel mask is derived. The resulting number of voxels is indicated in column 3 of Table 1.

3.2. Experimental protocol

Each dataset was then split into two: T, a dataset used for off-line (batch) training, and S, a dataset which will be prepared and presented to the classifier as online (streaming) data. We decided to shuffle and sample from the data. We note that this will break the autocorrelation of the fMRI signal; however, in order to explore semi-supervised learning for streaming fMRI data, the method first needs to be shown to work for stationary, independent and identically distributed (i.i.d.) data. This issue is discussed later in relation to presenting the online data stream.

Having selected T, we oversample the remaining data to construct S with 500 objects. This is the closest approach to construct i.i.d. sets.

Following the recommended procedure by De Martino et al. [6], we pre-selected a fixed amount (K) of voxels. This is achieved by taking the K voxels with maximum activation, based on T. Both training and testing data are normalised, using the mean and standard deviations calculated for T.

We consider three individual classifier and ensemble scenarios:

Scenario A: *no updates* (*fixed*): We train a 'fixed' off-line random subspace ensemble on T alone. The online data points from S are then presented one at a time. The classifiers are *not updated* during the online phase. We measure the running classification accuracy, which at time t is the ratio of correctly labelled data points out of the t seen data points. This allows us to compare whether using naive labelling is better than the no-action scenario.

Scenario B: *supervised updates* (*supervised*): We train the batch version of the classifier on T. The true class labels are assumed to be immediately available after classification is made in the online training. As each point from S is presented the current classifier is trained and tested on the new data point.

Scenario C: *unsupervised updates* (*naive*): Again we train the batch version on T. We assume that class labels for S are unavailable. The classifier is re-trained by augmenting the

[2] The images were selected from a benchmark database, International Affective Picture System (IAPS) [27], where each image has been rated on scales from 0 to 10 on two dimensions: arousal (calm to excited) and valence (negative to positive).

training data with the current observation and the label proposed by *the classifier* as the true label.

We conducted experiments with the following parameters values: the number of pre-selected voxels $K=500$, ensemble size $L=[5,9,11]$ and feature set cardinality $M=[20,50,100]$. The cardinality of the training sets was $|T|=[20,40,100]$. Due to the random nature of the feature selection for the RS ensemble, experiments were repeated 50 times, and the results were averaged.

For each scenario, we also considered the error rates of the individual classifiers which make up the ensembles.

Computational costs of the preprocessing and classification are not assessed quantitatively here. We do not expect this to form a major challenge; however, given that our classifiers have relatively low computational costs, and that earlier studies have demonstrated the feasibility of real-time classification [15,7,16,25] while processing speed of computers is increasing over time.

3.3. Kappa-error diagrams

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

Kappa measures the level of agreement between the classifiers while correcting for chance [8]. The pairwise κ is defined as follows:

$$\kappa=\frac{2(N^{11}N^{00}-N^{01}N^{10})}{(N^{11}+N^{10})(N^{10}+N^{00})+(N^{11}+N^{01})(N^{01}+N^{00})}, \qquad (1)$$

where N^{11} is the number of testing examples on which both classifiers are correct, where N^{00} is the number on which both classifiers are wrong, N^{10} is the number on which classifier 1 is correct and classifier 2 is wrong, and N^{01} is the number where classifier is wrong and classifier 2 is correct.

We seek to answer the following questions:

Individual vs ensemble: For classification of unlabelled fMRI data, we ask whether an individual classifier, or classifiers in an ensemble framework yield better results. In line with previous and existing research we expect classifier ensembles to have higher accuracy than an individual classifier. This may not be true if the individual classifiers deteriorate progressively. At some point the ensemble will become worse than the average individual classifier.

Fixed vs untrained updates: For streaming fMRI data, we want to know whether it is advantageous to update the classifier using naive labels or whether this is detrimental to the ensemble and a fixed pre-trained classifier is more accurate.

4. Results

We calculated the cumulative error progression for each time step. For time-step t the cumulative error is $\sum_{j=0}^{t} e(j)/n$, where $e(j)$ is 0, if the classifier/ensemble has labelled the point at time j correctly, and 1, otherwise. The 'final' errors for the three datasets, taken at time $t=500$ are summarised as colour plots in Figs. 2–4. For each combination of M and $|T|$, we generated ensembles of $L=[5,9,11]$ classifiers, giving a total of 25 individual classifiers. The individual error is taken as the mean error of these 25 classifiers at time $t=500$. Each column of the table represents a value of L, with the last column showing the mean error of the individual classifiers, titled 'I'. Each row of the table corresponds to a value of $|T|$. Within each coloured grid, rows correspond to values of M, and columns to the three RS ensemble methods, fixed (F), naive (N) and supervised (S).

As expected, the supervised classifier is superior to the fixed and naive classifiers. We compare the final error scores at $t=500$ for the fixed ensemble and the naive ensemble in order to see which scenario works best for unlabelled data. The results of this comparison are summarised in Table 2. A '$+$' indicates that the naive ensemble performs better than the fixed ensemble. A '$-$' indicates that the naive ensemble performs worse than the fixed ensemble. Significance was calculated using a paired t-test, uncorrected for multiple comparisons. All statistical analyses in this paper were carried out using the Matlab statistics toolbox.[3] Significant results at $\alpha=0.05$ are indicated by $\oplus$ and $\ominus$.

For these parameters the results suggest that the naive ensemble is on a par or better than the fixed ensemble ($21\oplus$, $23+$, $35-$ and $4\ominus$). For $M\geq 50$, the naive ensemble performs much better than the fixed ensemble, ($20\oplus$, $19+$, $13-$ and $2\ominus$). Dataset EB was the most 'difficult' for the classifiers, as this is where different emotions are being recognised. The other challenge with this dataset is the addition of a third class.

In order to understand the mechanism of improvement through naive labelling ensembles, we look at the progression of the error over time and the corresponding time-trajectory on the kappa-error diagram. Fig. 5(a) shows the error plot for EB2 with $L=11$, $M=100$ and $|T|=100$. The plot is taken from $t=25$ onwards, as at low t there are large fluctuations in the cumulative error leading to the plot appearing noisy and unstable. If the plot was to be taken from $t=0$, then the plots for all scenarios would start from one point, as the same off-line classifier is used in each case. The marker and line colour indicate the base classifier, a solid line indicates the classifier ensemble whilst a dashed line indicates the individual classifier.

We expect to see the error rate of the fixed classifier to remain constant over time. The error rate of the supervised classifier will drop as the classifier sees more data. We hope to see the naive classifier follow the same pattern as the supervised classifier, in that the error will drop as t increases, thus showing the naive labelling strategy to be beneficial.

The figure shows the dashed lines, representing the individual classifiers, above the corresponding solid lines. This indicates that the classifier ensembles outperform the individual classifiers. The error of the supervised ensemble is seen to drop over time and is easily the most accurate classifier ensemble, whilst the error of the fixed classifier ensemble remains constant. The naive ensemble is seen to improve over time, with significantly better results than the fixed ensemble.

Each ensemble can be plotted as a cloud of points in the kappa-error diagram. There is one ensemble at every time point t. It is interesting to see how the cloud shape and position changes with time. For example, the cloud for the fixed ensemble is expected to float about the initial point, as the only difference from one time point to the next will be the estimate of kappa and the individual errors. The classifier and the ensemble parameters do not change, hence the movement will be only a small fluctuation. The supervised ensemble, on the other hand is expected to drop down the plot, indicating that the individual accuracies improve with more data being seen. It is curious how the diversity of the ensemble progresses, i.e., whether the cloud will move to the left (larger diversity) or right. Instead of plotting the entire clouds of points, we decided to plot the trajectories of the centres. The endpoint of each trajectory is indicated with a marker. A good classifier ensemble will be both accurate and diverse, and thus appear near the bottom left hand corner of the

[3] http://www.mathworks.co.uk

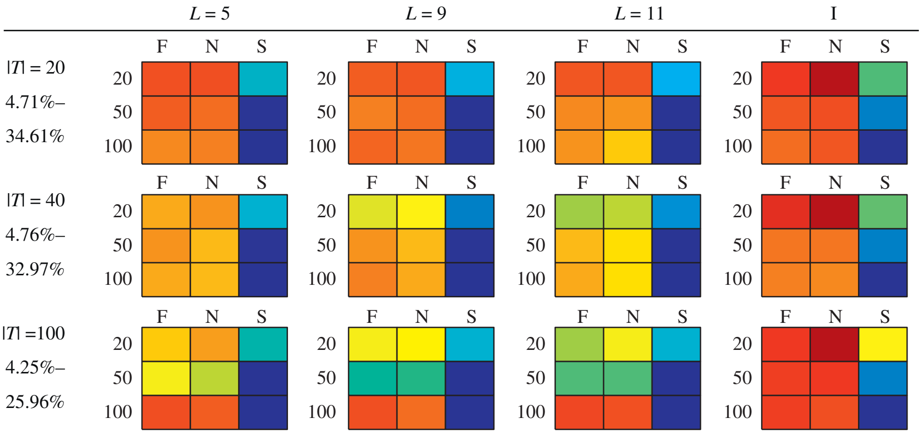

Fig. 2. Final cumulative error scores (%) for the EN1 dataset. The range of final error scores is indicated by %. Error scores are coloured from blue through to red representing low and high errors, respectively. L is the ensemble size. M is the cardinality of feature subsets (values of M are shown as rows of each coloured grid). $|T|$ is the cardinality of the off-line training set. 'F', 'N' and 'S' correspond to fixed, naive and supervised ensembles. 'I' corresponds to mean individual error of classifiers for a given M and $|T|$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

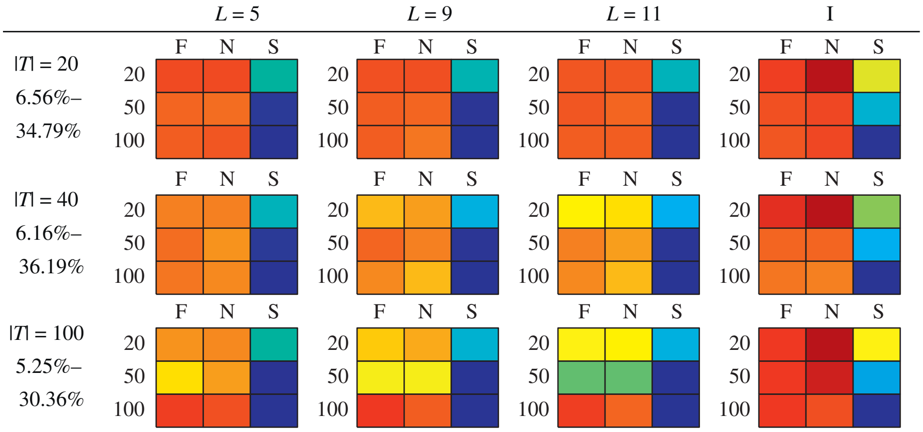

Fig. 3. Final cumulative error scores (%) for the EN2 dataset. The range of final error scores is indicated by %. Error scores are coloured from blue through to red representing low and high errors, respectively. L is the ensemble size. M is the cardinality of feature subsets (values of M are shown as rows of each coloured grid). $|T|$ is the cardinality of the off-line training set. 'F', 'N' and 'S' correspond to fixed, naive and supervised ensembles. 'I' corresponds to mean individual error of classifiers for a given M and $|T|$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

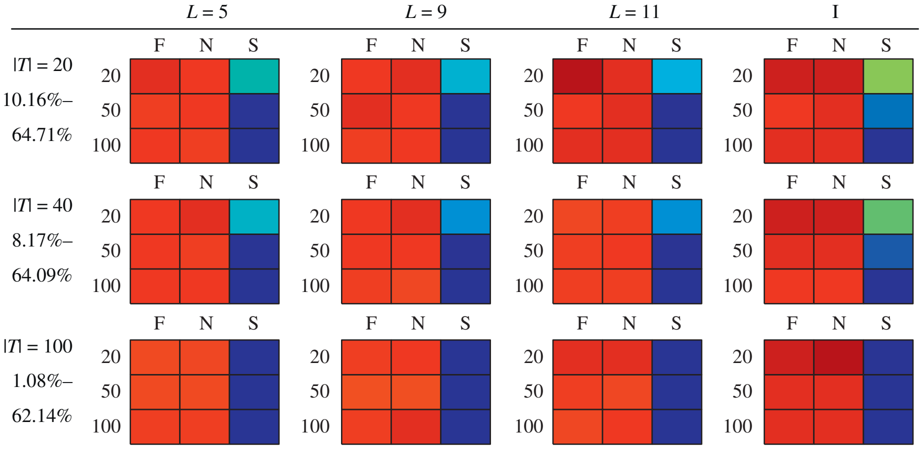

Fig. 4. Final cumulative error scores (%) for the EB dataset. The range of final error scores is indicated by %. Error scores are coloured from blue through to red representing low and high errors, respectively. L is the ensemble size. M is the cardinality of feature subsets (values of M are shown as rows of each coloured grid). $|T|$ is the cardinality of the off-line training set. 'F', 'N' and 'S' correspond to fixed, naive and supervised ensembles. 'I' corresponds to mean individual error of classifiers for a given M and $|T|$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

diagram. As we plot the trajectories of the ensembles over time, if the ensemble improves, we expect to see the trajectory progress towards the bottom left corner.

Fig. 5(b) shows the kappa-error trajectories corresponding to EB2 with $L=11$, $M=100$ and $|T|=100$. The trajectory of the supervised ensemble tracks down as accuracy increases. The diversity decreases slightly, this suggests that all classifiers within the ensemble are being driven towards the optimal classifier, thus are becoming more similar. The fixed ensemble, as expected, shows very little progression. The naive ensemble shows an increase in both accuracy and diversity, both of which are desirable characteristics.

Fig. 6(a) and (b) show the typical patterns of the individual classifiers for the fixed and supervised ensembles, respectively: the error of the fixed classifiers remains constant while the error of the supervised classifiers drops over time.

Table 2
Direct comparison of fixed and naive ensembles. '+' and '−' respectively, represent a 'win' or 'loss' by the naive ensemble. A circle surrounding the + or − indicates that the result is statistically significant at significance level $\alpha = 0.05$. L is ensemble size, M is cardinality of feature subsets, $|T|$ is cardinality of training dataset.

$\lvert T\rvert$ M	EN1			EN2			EB		
	20	40	100	20	40	100	20	40	100
$L=5$									
20	−	−	⊖	−	−	−	+	−	−
50	+	⊕	+	+	⊕	⊖	−	+	+
100	−	+	⊕	−	⊕	⊕	+	+	+
$L=9$									
20	−	−	−	−	−	−	−	−	−
50	−	⊕	⊖	+	⊕	−	+	+	−
100	+	⊕	⊕	⊕	⊕	⊕	−	+	−
$L=11$									
20	+	−	⊖	+	−	−	⊕	−	+
50	+	⊕	−	⊕	⊕	+	−	−	+
100	⊕	⊕	⊕	+	⊕	⊕	−	+	−

For the naive classifiers, we would like to see a similar shape to the supervised classifier. Fig. 7(a) and (b) show two cases of the patterns of the individual naive classifiers. In both cases the naive ensemble performed significantly better than the fixed ensemble. Fig. 7(a) is a case where the naive classifiers show a desirable learning pattern, improving over time. In Fig. 7(b), some classifiers are shown to display runaway behaviours. What is interesting in this case is that the naive ensemble still performs better than the fixed ensemble, indicating that the ensemble environment constrains the runaway behaviour.

4.1. Individual vs ensemble

From Figs. 2–4 we compare the error of the individual classifiers with the error of the ensembles. The error rate for the ensembles can be seen to be lower. This matches the hypothesis that a classifier ensemble is more accurate than an individual classifier. In Fig. 5(a), the error progression of the individual classifiers can be directly compared with the classifier ensembles. The classifier ensembles are seen to be more accurate than their individual counterparts.

4.2. Fixed vs unsupervised updates

From Table 2 we are able to directly compare the results from the fixed classifier with those of the naive ensemble. For the correct parameters the naive ensemble unsupervised update

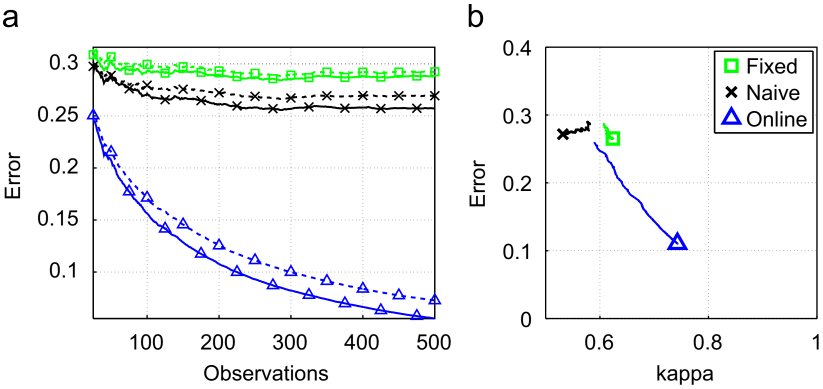

Fig. 5. Figures for EN1 data taken with ensemble size $L=11$, feature set cardinality $M=50$ and training dataset cardinality $|T|=40$. (a) Error progression. Solid line indicates classifier ensembles. Dotted line indicates individual classifiers. Plot illustrates changes in error over time. (b) Kappa-error progression. Kappa-error progression plots the changes in pairwise accuracy and diversity as the classifier ensembles learn over time.

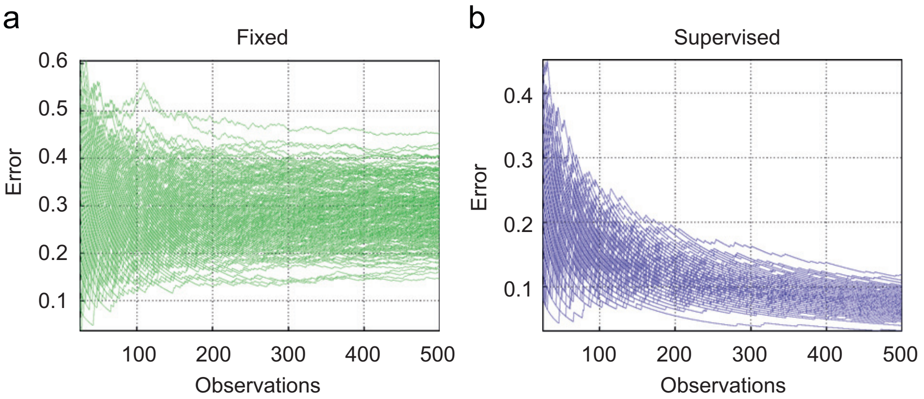

Fig. 6. Typical error progression of individual classifiers (dataset: EN1, size of ensemble: $L=5$, cardinality of feature subsets: $M=100$, cardinality of training dataset: $|T|=20$). (a) Error rate of fixed classifiers remains constant. (b) Error rate of supervised classifiers drop over time.

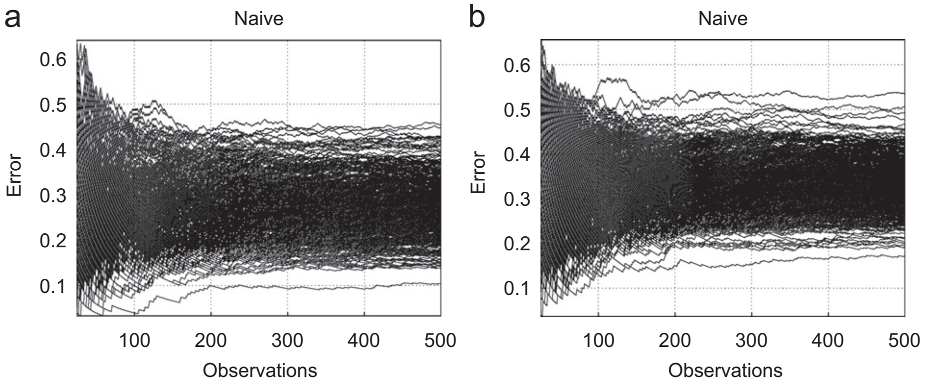

Fig. 7. Comparison of different individual classifier progressions for the naive ensemble. (a) Dataset EN1, size of ensemble $L=9$, cardinality of feature subsets $M=100$ and cardinality of training dataset $|T|=40$. (b) Dataset EN2, size of ensemble $L=11$, cardinality of feature subsets $M=50$ and cardinality of training dataset $|T|=20$.

strategy is beneficial to the ensemble. Specifically, when the cardinality of the feature set is $M \geq 50$ the naive ensemble performs better than the fixed ensemble. The method is tested on two 2-class datasets and one 3-class dataset. The method performs best for the 2-class datasets, when an accuracy of the initial fixed ensemble is higher.

5. Conclusion

We have shown that classifier ensembles are more accurate than individual classifiers. Our experiments also show that given an appropriate choice of parameters, classifiers updating using the naive labelling strategy perform well within an ensemble framework. In particular, on simple 2-class datasets. We have shown that for sufficient training data, a naive classifier ensemble performs significantly better than a fixed, pre-trained classifier ensemble.

During a real-time fMRI experiment, there is the potential for concept drift. An online classifier working in this environment is required to be capable of updating and adapting during the course of the experiment. Naive labelling offers an intuitive solution to this problem.

In our experiments we have treated the data as i.i.d., though this is not strictly the case for fMRI in general. This approach serves as a first step towards semi-supervised learning for streaming fMRI data. The non-i.i.d. case raises new questions. Autocorrelations and the non-stationary nature of streaming fMRI data may weight and ‘pull’ an online classifier in a certain direction, encouraging runaway traits. To counter this we may harness the correlative properties of the fMRI signal within the classifier, and thus use the multiple instances to our benefit. Future work includes using the naive ensemble for streaming fMRI data in order to simulate a real-time scenario. More comparisons with other classifiers such as the SVM, and using different base classifiers with the naive update strategy will provide further insight into the potential applications of the method.

Acknowledgements

We would like to thank Dr. D. Linden of Bangor University and Dr. S. Johnston for supplying us with the datasets.

References

[1] R.W. Cox, A. Jesmanowicz, J.S. Hyde, Real-time functional magnetic resonance imaging, Magnetic Resonance in Medicine 33 (2) (1995) 230–236.

[2] R.W. Cox, A. Jesmanowicz, Real-time 3D image registration for functional MRI, Magnetic Resonance in Medicine 42 (1999) 1014–1018.

[3] F. Cozman, I. Cohen, M. Cirelo, Semi-supervised learning of mixture models, in: Proceedings of the 20th International Conference on Machine Learning 2003, pp. 99–106.

[4] R.C. deCharms, Applications of real-time fMRI, Nature Reviews Neuroscience 9 (9) (2008) 720–729.

[5] R.C. deCharms, F. Maeda, G.H. Glover, D. Ludlow, J.M. Pauly, D. Soneji, J.D.E. Gabrieli, S.C. Mackey, Control over brain activation and pain learned by using real-time functional MRI, Proceedings of the National Academy of Sciences of the United States of America 102 (51) (2005) 18626–18631.

[6] F.D. Martino, G. Valente, N. Staeren, J. Ashburner, R. Goebel, E. Formisano, Combining multivariate voxel selection and support vector machines for mapping and classification of fMRI spatial patterns, NeuroImage 43 (2008) 44–58.

[7] A. Eklund, H. Ohlsson, M. Andersson, J. Rydell, A. Ynnerman, H. Knutsson, Using real-time fMRI to control a dynamical system by brain activity classification, in: Proceedings of MICCAI’09, Springer, London, UK, 2009.

[8] J.L. Fleiss, Statistical Methods for Rates and Proportions, John Wiley & Sons, 1981.

[9] M. van Gerven, J. Farquhar, R. Schaefer, R. Vlek, J. Geuze, A. Nijholt, N. Ramsay, P. Haselager, L. Vuurpijl, S. Gielen, P. Desain, The brain–computer interface cycle, Journal of Neural Engineering 6 (2009).

[11] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning, Springer, 2001.

[13] J.D. Haynes, G. Rees, Decoding mental states from brain activity in humans, Nature Reviews Neuroscience 7 (7) (2006) 523–534.

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

[15] M. Hollmann, T. Mönch, S. Mulla-Osman, C. Tempelmann, J. Stadler, 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 (2008) 154–162.

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

[18] M. Misaki, Y. Kim, P.A. Bandettini, N. Kriegeskorte, Comparison of multivariate classifiers and response normalizations for pattern-information fMRI, NeuroImage 53 (2010) 103–118.

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

[20] L. Kuncheva, C. Whitaker, A. Narasimhamurthy, A case study on naïve labelling for the nearest mean and the linear discriminant classifiers, Pattern Recognition 41 (2008) 3010–3020.

[21] L.I. Kuncheva, C.O. Plumpton, Adaptive learning rate for online linear discriminant classifiers, in: Proceedings of S+SSPR, Orlando, Florida, USA, 2008, pp. 510–519.

[23] L.I. Kuncheva, J.J. Rodríguez, Classifier ensembles for fMRI data analysis: an experiment, Magnetic Resonance Imaging 28 (2010) 583–593.

[24] L.I. Kuncheva, J.J. Rodríguez, C.O. Plumpton, D.E.J. Linden, S.J. Johnston, Random subspace ensembles for fMRI classification, IEEE Transactions on Medical Imaging 29 (2010) 531–542.

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

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

[27] P. Lang, M. Bradley, B. Cuthbert, International affective picture system (IAPS): technical manual and affective ratings.

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

[29] K.P. Nigam, Using unlabeled data to improve text classification, Ph.D. Thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, US, 2001.

[30] K. Norman, S. Polyn, G. Detre, J.V. Haxby, Beyond mind-reading: multi-voxel pattern analysis of fMRI data, Trends in Cognitive Science 10 (9) (2006) 424–430.

[32] C.O. Plumpton, L.I. Kuncheva, D.E.J. Linden, S.J. Johnston, On-line fMRI data classification using linear and ensemble classifiers, in: Proceedings of 20th International Conference on Pattern Recognition2010.

[33] M. Seeger, Learning with labeled and unlabeled data, Technical Report, University of Edinburgh, 2001.

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

[35] H. Zhu, X.L. Tang, Classifier geometrical characteristic comparison and its application in classifier selection, Pattern Recognition Letters 26 (6) (2005) 829–842.

Catrin O. Plumpton studied Mathematics at Bangor University, UK before completing M.Sc. in Computer Systems in 2007. She is now studying Ph.D. at the School of Computer Science, Bangor University. Her main research interests include machine learning, specifically classifier ensembles and fMRI data analysis.

Ludmila I. Kuncheva received her M.Sc. degree from the Technical University of Sofia, Bulgaria, in 1982, and her Ph.D. degree from the Bulgarian Academy of Sciences in 1987. Until 1997 she worked at the Central Laboratory of Biomedical Engineering at the Bulgarian Academy of Sciences. Prof. Kuncheva is currently a Professor at the School of Computer Science, Bangor University, UK. Her interests include pattern recognition and classification, machine learning, classifier combination and fMRI data analysis. She has published two books and above 150 scientific papers.

Nikolaas N. Oosterhof received a double master's degree in Computer Science and in Philosophy of Science, Technology and Society from the University of Twente, the Netherlands, and a third master's degree in Cognitive Science from the University of Amsterdam, the Netherlands. Currently he is a Ph.D. candidate in Cognitive Neuroscience at Bangor University, UK, where he uses pattern classification and multivariate statistics for the analysis of fMRI data.