

Optimisation of Support Vector Machine Classifier Algorithms for use as a brain computer interface in real-time fMRI applications

G. B. Williams¹

¹Wolfson Brain Imaging Centre, University of Cambridge, Cambridge, United Kingdom

Introduction

In recent years support vector machines (SVMs) have been used for analysis of functional magnetic resonance images (fMRI). They are particularly well suited to categorisation and classification of brain states and they can subsequently be used to infer spatial activation[1] as an alternative to the more conventional general linear model (GLM) approaches. At the same time, real-time fMRI (rt-fMRI) technology has been developed to allow the rapid export and analysis of data from the scanner environment. Many rt-fMRI experiments rely on general linear model approaches, which allow good sensitivity for spatial activation. However, they rely on an accumulation of data to provide valid statistics and are less useful for the classification of a single time point. For some contexts, in particular the development of a brain computer interface (BCI), temporal classification of scans into “active” or “baseline” (or “yes” and “no”) are precisely the information that we require, and we are less interested in the spatial pattern of activation. LaConte[2] has previously successfully presented results on the application of SVM classifiers to real-time fMRI datasets with a variety of paradigms, but many questions remain over the optimal choice of experimental design. In particular there is “drift” in classification status that appears to favour active classifications more than baseline classifications which is hypothesised to be due to subject motion[2]. In this work, we explore how different acquisition and processing strategies can increase the reliability of the classifier, while constrained by the fact that results must be reported rapidly and in time-order.

Methods

Acquisition

Eleven healthy volunteers were scanned on a Siemens 3T Tim Trio using a paradigm that has previously been successfully used to identify the supplementary motor area in healthy volunteers and some patients with impaired consciousness [3]. The image matrix was 64x64, with an in-plane resolution of 3x3mm. 32 slices were acquired with a thickness of 3mm (interslice gap 0.75mm). TR was 2 seconds. Subjects were initially asked to imagine playing tennis for 20 seconds, followed by a period of rest for 40 seconds, and “tennis” for a further 20 seconds. This sequence was repeated four times. After four such blocks, the paradigm was reversed (so that it started with 20 seconds of “tennis”), and repeated a further three times. This resulted in a total of 320 time points (with equal numbers of “rest” and “tennis” time points) and a scan time of 10 minutes and 40 seconds. This relatively unusual block design was chosen so that the first moment of the task label was zero. The initial phase shift was inserted in order to test whether the classifier tended to favour “tennis” states since on average they came later in the acquisition, and thus later scans were more similar to them. The reason for the reversal of the block design half way through was to avoid artefacts resulting from any periodic temporal variation in the classifier. We also deliberately use a single acquisition for training and classification in order to eliminate any effects on the classifier resulting from differing calibrations at the start of each acquisition.

Support Vector Machine implementation

SVM light software[4] was used to implement the classifiers. For a binary classification such as this we can define a “decision boundary” in a space with the same dimension as the number of independent measurements in each time point (i.e. the number of voxels in the analysis). One side of the decision boundary we classify a point as being “tennis”, and the other “rest”. In brief, the SVM algorithm determines (from a training data set where the classifications are known) the time points that are most difficult to classify (they are closest to the decision boundary).

A decision function is formed from a linear combination of these points, and this is used to classify subsequent unknown time points. For more details see [1],[2] or [4]. For each analysis, 160 time points were used for the training and 160 were treated as unknown time points, allowing an estimate of the success rate of the classifier for each analysis method. All time points were aligned to correct for subject motion by built-in Siemens image reconstruction software.

Masking procedures: In order to restrict the analysis to voxels within the brain, a mask was created from the first image of the time series. Images were thresholded at 80% of the mean voxel value, which was an ad-hoc threshold that broadly identified brain tissue and excluded voxels with very little signal.

Data preprocessing: Two different preprocessing methods were attempted: with and without temporal detrending. In the latter case a linear intensity drift (calculated from all data acquired so far) was assumed and subtracted from the image on a voxel-by-voxel basis. Additionally, the experiments were repeated after smoothing with a 8mm Gaussian kernel.

Block Ordering: It is interesting to determine whether it is optimal to acquire all the training data prior to classification, or whether the scanner stability during the acquisition is such that it is more appropriate to dynamically update the classifier during the sequence. In order to test this, we have relabelled the data so that the 160 training blocks are distributed through the whole acquisition, and assessed whether this affects the error rate. Each learning/classifier period was constrained to be 40 time points long. For example, we might treat the first 40 scans as “learning” scans where the class is unknown, the next 40 time points as “unknown” states, and repeat this sequence four times. With these constraints there are 20 different arrangements of learning/classifier periods.

Results & Discussion

A typical result (corrected for classifier drift) is shown in Fig. 1 and the results for all subjects and settings summarised in Table 1. They showed that temporal detrending did not in general reduce the classifier drift and in all cases the correction employed by LaConte et al was still able to improve the success rate. The redesigned order of the blocks did not appear to significantly change the corrected success rate from that found in a previous study (88%)[2]. Smoothing the input data also did not significantly change the results. Changing the task order and including learning and classification periods within the same acquisition does not appear to compensate completely for classifier drift despite the presence of motion correction. It was found that it was better to have all learning scans acquired near the beginning of the acquisition, although the exact scheme varied between individuals (see Table 2). Further investigation of this is needed for longer acquisitions.

References

[1] LaConte S. et al. NeuroImage 26:317-329 (2005). [2] LaConte S. et al. HBM 28:1033-1044 (2007). [3] Owen A. et al. Science 313 p1402 (2006). [4] Joachims T. In: Scholkopf B. et al (Eds), Advances in Kernel Methods – Support Vector Learning, MIT Press (1999).

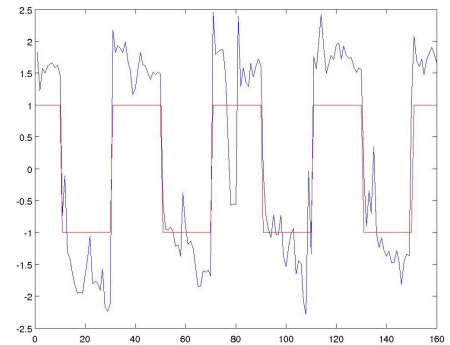


Fig. 1: SVM classification results on a subject “playing tennis”[3]. The red line represents the block design. The correlation between task (red line) and classification result (blue line) is very strong.

	Success Rate (uncorrected)	Success rate (dynamically corrected)
No input detrending	78%	87%
Detrending	73%	80%

Table 1: The error rates for classified data with/without data detrending and with/without correction for classifier drift

	Best scheme (uncorrected)	Best scheme (dynamically corrected)
No input detrending	LLCLCCLC (av. 13%)	LLLLCCCC (av. 3%)
Detrending	LLLCCLCC (av. 7%)	LLLCCLCC (av. 6%)

Table 2: The best overall schemes (‘L’ represents a learning period and ‘C’ as classification period) and average decrease in success across individuals if this scheme is employed rather than their individual optimum.