

Pittsburgh Brain Activity Interpretation Competition 2006

Methods Description

Fast Prediction Using Large Numbers of Voxels with Kernel Based Non-linear Partial Least Squares Regression

Project Abstract

Nonlinear partial least squares (PLS) regression was used to predict feature ratings, using from 1000 to about 67000 voxels from lagged datasets. The algorithm uses the Gaussian kernel trick to efficiently process the covariance matrix of fMRI data. Essentially, given X a data matrix, where each row is a volume, and Y a target time course, the algorithm finds an estimate, B , for the equation

$$X * u * B = Y,$$

where u is the principle component of the covariance of X and Y . The kernel trick uses the fact that u can be represented in the space spanned by X' , so that

$$X * X' * b * B = Y,$$

where b are the coefficients of u . As long as only the kernel matrix, $K = X * X'$, is used there is no need to compute the transformation of X into a new feature space.

An important advantage of this method that makes it particularly attractive for fMRI, in comparison say to a neural network, is that an analysis using the full data matrix of all voxels can be processed in a few minutes on a typical workstation. An important disadvantage of this method, in comparison say to support vector machines, is that little is known about effective means of penalization to ensure sparse solutions and avoid overfitting. Thus, the main contribution of this research is to evaluate this method and the possibility of its use. In general, I have found this method is effective when a widely distributed network of voxels, possibly non-significant individually, is useful for prediction.

Introduction

1. What was the approach you used?

I used a nonlinear Partial Least Squares (PLS) Regression method. This method was developed as an outgrowth of the support vector machine learning research, and is described in various papers related to kernel based principle component analysis. My implementation is based on the treatment presented in Shawe-Taylor & Christianini (2004). Figure 1 in the appendix gives a visual demonstration on a toy problem.

2. What was the goal (e.g., general best prediction, particular features, assessment of a particular method, interpretation of a given brain region, etc.).

The goal was two fold. First, I wanted to evaluate the relevance of using a nonlinear method to analyze fMRI data. The nonlinear PLS consistently outperformed the linear PLS in prediction,

even though both are able to solve the regression problem. Secondly, I wanted to explore if it is better or possible to use all the information in the complete fMRI data set. As stated above, some feature ratings, such as ('Other settings') seemed to benefit from a large number of voxels, which is clearly much more information than available in ROI analysis, and it can execute these in 2-3 minutes on a contemporary workstation.

3. What is the previous research utilizing these methods?

Several papers in the fMRI literature have used kernel based approaches for pattern prediction or exploratory multivariate analysis of fMRI data, including kernel based PCA and support vector machine for classification. However, to my knowledge kernel based PLS has not been used, although a linear PLS method for statistical inferencing has been described in detail and used by McIntosh and colleagues (e.g. McIntosh et al, 1996).

4. What did you hope to achieve?

I originally hoped to find a method that could be useful for widely distributed networks of low activation that would not be necessarily deemed active with mass univariate statistics, and a method to handle nonlinear patterns of activity. Basically, a nonlinear multivariate approach. In comparison to a linear PLS, the nonlinear method usually achieved better prediction, although the linear method can fit the data just as well with about the same number of factors. More work will have to be done to identify the necessity and nature of nonlinear fMRI patterns.

5. Why might this be a valuable approach?

It has been pointed out that fMRI analysis is a kind of phrenology. Pattern prediction in fMRI has demonstrated the potential for distributed representations in a wide network of regions (Haxby, et al. 2001). This method is potentially of great importance because it can efficiently search many combinations of regions or numbers of voxels.

Method

The goal of the PLS algorithm is to find a linear regression estimate, B , for the equation

$$X^*u^*B=Y,$$

where u is one principle components of X^*Y . Thus X^*u is the score of each image onto the eigenimage u . Thus u can be represented in the space spanned by X^* , so that the above equation can be written as:

$$X^*X'^*b^*B=Y,$$

where b are the coefficients of u , ie $X^*b=u$. Now, as long as only $K=X^*X'$, the kernel, is used there is no need to compute X , and as long as X^*X' is positive semi-definite, it will have the reproducing kernel Hilbert space property that ensures some feature space does exist.

The algorithm ensures that only K is used by the following. Because u can be found by a singular value decomposition of $X^*Y^*Y^*X$, (which is sometimes also called the kernel trick), then letting $X^*b=u$, we have that $XX^*YY^*XX^*b=XX^*b$ (see appendix for pseudo code).

Results and Discussion

Pre-processing and Training procedure.

I started with the raw fMRI data in analyze format and performed realignment using SPM2 (Wellcome Center for Neuroimaging, UC London). I then performed high pass filtering in SPM2 (512 second period) as part of the functions for extracting the raw data. After this step I had a data matrix for each subject and each movie (9 data sets) that was $T \times N$, where N is approximately 35,000 voxels, and T is the number of volumes in the scanning run.

For each subject and each feature rating, I trained on data from season 1 with targets that were feature ratings for season 1, and used that to predict ratings for season 2 data, and vice versa. I searched through parameter space of lags, variance, correlation thresholds, and either using all volumes or just those in the movie (about 5000 tests). Then I merely selected the best parameters and used that to build predictions for season 3.

Searching Parameter Space.

Executing the algorithm required searching through parameter space to choose reasonable value. The kernel variance was chosen to be the mean of $|X(i,:) - X(:,j)|$ multiplied by 1.5 to 3.5. The number of components in regression were chosen by merely using season 1 (or 2) data to predict season 2 (or 1) data, and recording what the correlation was in the training set that led to the best prediction in the test set. In general, I found this to work better than information criterion, F-tests of error, or looking for 'knees' plots of the percentage of variance accounted for by components. Anecdotally, it seems that the most predictable features (ie Faces and Language), achieved higher correlation to the training set before overfitting.

Several other parameter were chosen that are relevant to the problem of prediction with fMRI data. I built the data matrix with various lags, ie pasting X with $X(t-1)$, $X(t-2)$, etc.. In general I found lag 2 or lag 3 was optimal. In a sense this accounts for variations in hemodynamics response across subjects, regions, and features.

For voxel selection, I varied the number of voxels in X by choosing voxels above some correlation threshold (e.g. 0.05 to .35). In general the most predictable features used less voxels (i.e higher threshold), and the range varied from 1000 to about a maximum of about 67000 voxels entering into the PLS analysis (recall a lag 3 data matrix would be about 4×35000 total possible voxels).

Post-processing.

All predictions were smoothed by a moving average, preliminary tests with synthetic data suggested that kernel PLS will account for spikes and noise in its factors, thus, a slight smoothing helped ensure that a prediction for a new session, which likely does not have noise at the exact same time or space, does not have predictions related to noise fitting. Also, most features prediction benefited from averaging across subjects, and in some cases further

smoothing. This final smoothing amount helped match the smoothness and covariance in the feature ratings.

Methods tried but not used.

I tried avoiding overfitting by using information criterion (AIC or BIC) on the regression part of the method, or by using F-tests that would indicate which additional components were most needed in the training data, or I tried looking for a 'knee' in the amount of variance accounted for in the data. No method seemed close to identifying the number of factors that produce the best prediction, thus I used the brute force search to find the best number. Clearly, more work needs to be done here.

I also found that using smoothed, normalized data achieved lower maximal predictions, and without that, it seems it might hard to use cross-validation methods hard to use across both seasons 1 and 2 data sets (but this is clearly work for the future).

References

Haxby, J.V., Gobbini, M.I., Furey, M.L., Ishai, A. Schouten, J.L, Peitriani, P. (2001) Distributed and Overlapping representations of faces and objects in ventral temporal cortex. *Science* 293, 2425-2430.

McIntosh, A. R., Bookstein, F.L., Haxby, J.L., Grady, C.L. (1996) *Spatial Pattern Analysis Using Partial Least Squares.*

Shawe-Taylor, J., Cristianini, N. (2004) *Kernel Methods for Pattern Analysis.* Cambridge University Press.

Appendix Materials

Pseudo-code

A pseudo-code version of the PLS procedure is the following:

Let X be a matrix of size $T \times N$, where T is the number of fMRI volumes, and N is the number of voxels. Let Y be a vector of size $T \times 1$ that represents an individual feature rating (both are mean centered).

Let $K = G(X * X')$ be a matrix of size $T \times T$, where G represents the Gaussian kernel in which the elements of K , ie k_{ij} , is given by a Gaussian function, $G = \exp[-(X(i,:) - X(:,j))/v]$, where $(i,:)$ is the i -th row, $(:,j)$ is the j -th column, and v is a variance parameter that controls the spread of the kernel. (Note that if G was replaced with a standard dot product, then the algorithm would be a linear partial least squares.)

Find principle component of $KYY' = t$, and solving the linear regression problem for B where $Kt * B = Y$

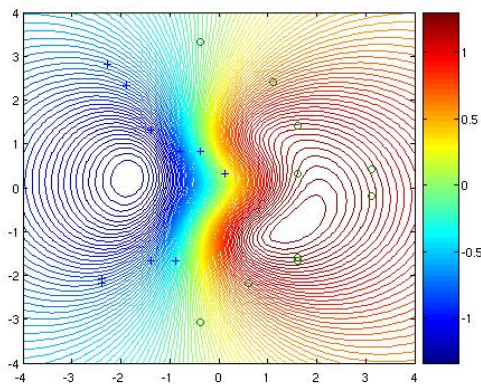
Finally, for the prediction problem one merely replaces $K=XX'$, with $K=ZX'$ where Z is a $S \times N$ data matrix for the unseen data, and S is the number of volumes therein.

Importantly, the PLS algorithm can deflate X (or K for the nonlinear PLS) to extract as many components are in rank of K , which is more components than available from a PCA analysis of the covariance of $X' * Y$. Then the deflated data matrix enters back into the PLS procedure. Each iteration leads to one more component added to the regression equation. Thus, one has to decide how many factors to choose for making predictions.

Toy Problem Demonstration.

A dataset with 2 voxels and 20 observations and was created and assigned to two classes, either '+' (1) or 'o' in Figures 1a and 1b. The +/o classes were given values of -/+ 1 in a target vector. The nonlinear PLS algorithm was applied and tested with a grid of all possible pairs of voxel values in the range +/-4, which produces a contour plot for the two possible components that were extracted. The value of the color in the color bar is essentially a 'factor score' onto a nonlinear component in input space. Notice the dividing line between the classes (for the color bar value of 0) bends and twists to nearly separate the classes. The 2nd component seems to be picking off particular observations that are problematic. This also demonstrates the potential for overfitting.

A.



B.

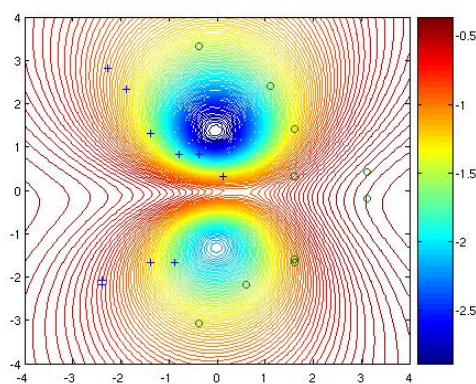


Figure 1. A. is the 1st nonlinear PLS component, B. is the 2nd nonlinear PLS component.

Optional Comment on Competition (not part of review)

What suggestions might you have for next year's competition?

A next step in the competition is to predict outcomes (ratings or brain activity or whatever) for a new subject that was watching a different video, but within the same category of training subjects data.