

Pittsburgh Brain Activity Interpretation Competition 2006

Methods Description

Gaussian process regression and recurrent neural networks for fMRI image classification

Project Abstract

We describe our approach for fMRI image classification using Gaussian process regression (GPR) and recurrent neural networks (RNN). The feature attributes were extracted from the 3D volume images by spatially clustering the voxels with high mutual information with respect to the feature ratings. Each cluster mask corresponds to one feature attribute for a volume image and is calculated as the mean of all the voxels in the cluster mask for the particular image. The prediction for movie 3 for a subject is performed using movies 1 and 2 (blanks removed) from the same subject as training using two classification schemes – GPR and RNN.

Introduction

The motive for our participation in the competition without much prior experience on similar problems was that we felt that the problem was very interesting and well posed (not to mention the prospect of fame and fortune, however improbable). Furthermore we were interested in comparing the performance of our methods that were designed without much *a priori* knowledge about the domain versus methods that explicitly use such knowledge.

Our effort initially centered on the design of classifiers operating on simple image moment features, and the tuning of their parameters to obtain the best accuracy. We however realized that the information in the images is noisy, and therefore spent most of our time subsequently on the design of better features. We use the mean of a set of contiguous voxels in each image as a feature. The method of image feature extraction is described below. We tried standard regression methods such as GPR and RNN with slight modifications to use some prior assumptions (perhaps incorrect) such as the temporal dependence of the response of the brain to visual stimuli. We briefly describe the methods below.

Method

Image Feature Extraction:

Feature attribute set 1: The feature extraction was conducted as follows. For each feature rating we found its mutual information to the value at every voxel in the image, separately for every subject and movie (movies 1 and 2 only, since we only have labels for them). For each subject, movie and feature rating, we found the top 1000 voxels that have the highest mutual information and created a subject-movie-and-feature rating-specific mask of the most ‘informative’ voxels.

We then took the union of the masks for all three subjects, both the training movies and all the feature ratings to obtain a mask of approximately 11000 voxels.

We took the (x,y,z) locations of the 11000 voxels and clustered them using k-means into 100 clusters. Each cluster is therefore a spatially proximal set of ‘informative’ voxels. We then computed the 100 features on each brain image by finding the average intensity of all the voxels in a cluster on that particular image. We constructed the feature attribute datasets for each subject and movie (all three movies) on which our regression algorithms operate.

Feature attribute set 2:

This was computed exactly as feature set 1, except a separate feature dataset is constructed for each feature rating. We chose the top 5000 pixels for each subject, movie and feature yielding a final voxel mask of approximately 11000 voxels. This was done because we felt that in order to reduce noise in the feature values, it is best to predict each feature rating with only the voxels that are related to it. Moreover the feature rating specific mask was clustered into 200 clusters, meaning that each brain image is transformed into 200 feature attributes separately for each feature rating.

Regression:

Gaussian Process Regression: To predict the feature ratings for movie 3 of a subject (test set) we used the data from movies 1 and 2 from the same subject as training. This was done because we observed considerable differences between the behaviors across subjects. We used feature set 2 and predicted each feature rating separately by using the feature attributes computed using the feature rating in question.

Preprocessing: We first performed blank normalization by subtracting from every sample in the training and test set the mean of the first 50 samples (blanks). We then removed all the instances corresponding to blanks from the training set. We then performed principal component analysis (PCA) and rotated the training and test sets onto the PCA axes. The PCA was performed on the covariance matrix computed on the dataset obtained by stacking the training and test sets. The data are then normalized by dividing each feature by its standard deviation. We then choose the top 80 features that have the highest mutual information with respect to the particular feature rating on the training set.

Regression: We used Gaussian process regression [1] using covariance matrix that depends on both the spatial and temporal distance of the points in the 80-dimensional feature space. Assume the vector of feature ratings on the training set is denoted by *Trainlabels*.

We first compute the distance matrices *TestTrain* and *TestTest* which are the squared distances of all the test points to all training points and all test points to all test points respectively. To incorporate temporal dependence of the distances we smooth these distance matrices by convolving the matrices with a Gaussian window of size 5x5 pixels with standard deviation =1.

We use this distance matrix to compute the covariance matrix which is a Gaussian kernel. The parameters for the kernel were set using cross-validation. The covariance matrix and the feature ratings for the training set are then used to predict the feature ratings for the test.

Post processing: Since the three subjects are rating the same movie we averaged the predictions obtained on the three subjects and used this as the prediction for all three.

Recurrent Neural Networks:

To predict ratings for movie 3 of a subject we used the data from movies 1 and 2 from the same subject as training. Each feature was predicted separately. The Recurrent Neural Network model was applied to the data represented using the feature attribute set 1 described above (i.e., 100 features to represent the brain image).

Preprocessing: We performed linear scaling and translation of the feature ratings to the interval [0,1]. Starting from the dataset represented using feature attribute set 1 (described above), we performed normalization forcing mean to be 0 and variance to be 1, i.e., all 100 features were normalized subtracting their mean and dividing by their standard deviation. Then from the training set we removed all the instances corresponding to blanks (label 31).

Regression: We used Recurrent Neural Networks (RNN) [2] with one hidden layer of 4 units. The recurrences were on the hidden layer only. We performed 5-fold Cross Validation, i.e., the training set was divided into 5 consecutive sets, and the training was done on all possible combinations of 4 of these sets, using the remaining set to stop the backprop algorithm to avoid overfitting. We performed 3 training trials for each validation set and we stopped the model when it yielded the highest correlation on the validation set. Among these 15 models we retained only those with positive correlation on the validation sets. The remaining models were then used to compute the outputs on the test set (movie 3), and all outputs were averaged weighted according to their correlations on the validation sets.

Development: The regression algorithms were tested and tuned using movie 1 for all subjects as training and movie 2 as test set and vice versa.

Results and Discussion

There are at least three kinds of context that are present in this problem that ought to be explicitly exploited to arrive at a reasonable accuracy. They are 1) the context imposed on the various feature ratings due to the non-arbitrariness of the visual scene, i.e. there are important relationships between the values of the various feature ratings, 2) the context imposed by the inertia in the scene, i.e., since the subjects are watching a video and not still images there cannot be drastic change in the scene from one frame to the next and 3) context imposed by the inertia in the response of the brain to visual stimuli.

We also believe that the image feature representation is much more critical issue for this problem than the choice of the regression algorithm. Prior knowledge about the domain should be exploited in designing these features.

We have tried several other methods which had various degrees of success. We tried several sets of image features by varying the number of pixels chosen and also using correlation instead of mutual information. We are still experimenting with other feature sets. We also tried other regression schemes like least squares regression, Parzen windows etc. but found that GPR and RNN gave the best results. We also tested standard feedforward neural networks and time-delayed neural networks. They performed slightly worse. We observed that for the "easiest" labels only one hidden unit was sufficient to obtain good results. In the final implementation we decided to use 4 hidden units because we did not know whether the task with two movies the regression problem required a more complex model.

All the programs used here have been developed in python/octave and will be made available upon request.

References

- [1] *Gaussian Processes for Machine Learning*, C. E. Rasmussen and C. K. I. Williams, The MIT Press, Cambridge, MA, 2006.
- [2] *A Field Guide to Dynamical Recurrent Networks*, John F. Kolen and Stefan C. Kremer (Editors), Wiley-IEEE, 2001