Voxel selection algorithms for fMRI

Henryk Blasinski

December 14, 2012

1 Introduction

Functional Magnetic Resonance Imaging (fMRI) is a technique to measure and image the Blood-Oxygen Level Dependent (BOLD) signal in the human brain. The BOLD signal is strongly correlated with the brain activity. Consequently fMRI makes it possible to image activity patterns in a brain of a living organism and thus observe and record its responses to a variety of stimuli. For this reason fMRI techniques have become very popular in psychology and cognitive sciences.

A typical fMRI experiment consists in stimulating subjects in some way (vision, hearing, touch etc.) and recording the corresponding activity pattern for a large number of small brain volumes called voxels. Activities within voxels are used as features for machine learning algorithms to relate brain areas with certain types of stimuli. Unfortunately the fMRI data has many pitfalls. In most cases it is characterized with a very large number of features formed by brain voxel activations (up to several thousands) and a low number of training examples, typically of the order of hundreds. Data pre-processing is of particular interest for two reasons. First, selecting a smaller set of features can improve stimulus prediction accuracy and reduce computational complexity. Second, given the location of voxels most relevant to the prediction task, conclusions about brain circuits can be made.

Current research employs a number of voxel selection techniques. They can be as simple as manual segmentation of regions of interest from the entire brain [1] or more elaborate involving statistical significance tests. Recently more advanced algorithms such as ridge regression, lasso, sparse logistic regression or graph Laplacian based methods have started to be more frequently used [2]. Many of these methods promote sparse solutions, i.e. such where many of the voxels are assigned a zero score and therefore considered uninformative. This notion agrees with neurological understanding of brain organization into different regions responsible for different tasks. For example speech recognition will cause increased activity of certain areas of the brain, and will have little influence on other.

So far none of the above methods explicitly encodes correlations between features using a covariance matrix. One of the problems with using empirical covariance is the fact that just a few data points are available making this matrix non-invertible. Therefore it cannot be directly used in optimization algorithms. This project investigates an approximate inverse covariance matrix estimation technique and its applicability to imposing a Gaussian prior distribution on feature scoring weights.

2 Algorithms

2.1 Univariate tests

Univariate tests evaluate the predictive power of each voxel independently of others. The most popular algorithm used is the t-test, which measures the probability p of a population being drawn from a distribution with some mean μ. For each voxel q, this mean is assumed to be the average activity during resting state when no stimulus is shown. It is then compared to the activity distribution of each of the P stimuli and the corresponding $p_{q,l}$ is computed. For an individual stimulus, the voxel score is then determined as $1 - p_{i,k}$ and therefore the entire score for P stimuli becomes

$$s_{q,\mathrm{t-test}} = \sum_{l=1}^{K} (1 - p_{q,l}) \,.$$

Another univariate test consists in evaluating individual voxels based on their performance as single features used for classification. The output label is predicted using just one voxel at a time, and the prediction accuracy denotes the feature score.

The two remaining scoring methods consist in computing the mutual information (MI) or covariance (Cov) between the voxel q time course $x_q^{(:)}$ and the class label indicator variable $\mathbf{1}\{y^{(:)} = l\}$. Since the mutual information definition for continuous variables is not convenient to use, the voxel time course can be quantized to discrete values.

$$s_{q,\mathrm{MI}} = \sum_{l=1}^{P} \mathrm{MI}(\mathbf{1}\{y^{(:)} = l\}, x_q^{(:)})$$

$$s_{q,\mathrm{Cov}} = \sum_{l=1}^{P} \left| \mathrm{Cov}(\mathbf{1}\{y^{(:)} = l\}, x_q^{(:)}) \right|$$

2.2 Multivariate tests

As opposed to univariate tests, where each feature is considered independently, multivariate tests aim at using multiple features at a time. Theoretically such an approach allows to discover feature correlations. Consequently a set of features, each of which performs poorly on its own, may result in a substantially improved classification accuracy or better representation of a particular stimulus. Since in most fMRI experiments stimulus classification is performed, it is desired to find a set of weights θ such that $\hat{y}^{(i)} = f(\theta^T x^{(i)})$ provides the best estimate of the true stimulus label $y^{(i)}$. Very high or low values of θ typically indicate that a particular feature has a significant contribution to the classification decision. In general, consider a P class classification problem with N different features, a feature q has a score

$$s_q = \sum_{l=1}^{P} \left| \hat{\theta}_q^{(l)} \right| \qquad q = 1, \ldots, N.$$

In the most general case, weight parameters are obtained by maximizing their likelihood function together with some penalty $\mathcal{P}(\theta^{(l)})$. In this approach the task is to maximize the likelihood function of the stimulus class label y, given the activity pattern time courses x.

$$\hat{\theta}^{(l)} = \arg\max_{\theta^{(l)}} \left\{ \mathcal{L}(\theta^{(l)}) - \lambda \mathcal{P}(\theta^{(l)}) \right\} = \arg\max_{\theta^{(l)}} \left\{ \sum_{i=1}^{m} \log p(y^{(i)} = l | x^{(i)}; \theta^{(l)}) - \lambda \mathcal{P}(\theta^{(l)}) \right\} \quad (1)$$

where λ is some tradeoff parameter. With $\mathcal{P}(\theta^{(l)}) = 0$, the expression simplifies to a maximum likelihood estimation. However, this method does not naturally promote sparsity in the solution and due to few training examples it overfits the data and generalizes quite poorly. The penalty term $\mathcal{P}$ alleviates these issues. Popular penalty functions involve the l_1 and l_2 norms (lasso and ridge regression) Seen from the probabilistic MAP perspective adding a penalty is equivalent to imposing a constraint on the parameter distribution. For example in ridge regression the prior is a Gaussian distribution $p(\theta^{(l)}) \sim \mathcal{N}(0, I)$.

In a recent paper Kamitani et al. [3] investigated the sparse logistic regression model for feature selection. Their method imposes constraints on hypothesis weight distribution. It is assumed that these weights are Gaussian with zero mean and some variance α^{-1}, $p(\theta_q^{(l)}|\alpha_q^{(l)}) \sim \mathcal{N}(0, \alpha_q^{-1})$. The individual weight variance parameter is not deterministic either, but rather distributed according to a gamma distribution, $p(\alpha_q) \sim \alpha_q^{-1}$. If during the likelihood maximization process the α_i parameter becomes very large, the corresponding voxel is deemed irrelevant and therefore pruned from the set. While this method is quite efficient at selecting a sparse voxel set, the solution relying on the Newton method is computationally complex. It requires multiple inversions of a $N \times P$ square matrix, where N is the number of features and P the number of stimuli. A more computationally efficient, albeit approximate, solution has been proposed by [4].

2.3 Proposed algorithms

2.3.1 Correlated logistic regression

Other methods typically assume independence between weights $\theta_q^{(l)}$. In correlated logistic regression a Gaussian prior distribution on weights $\theta^{(l)}$ is assumed, i.e. $p(\theta^{(l)}|l) \sim \mathcal{N}(0, \hat{\Sigma}_l)$, it however has a non-diagnonal covariance matrix. The penalty function is therefore given by

$$\mathcal{P}(\theta^{(l)}) = \theta^{(l)T} \hat{\Sigma}_l^{-1} \theta^{(l)}.$$

Since the number of data points is smaller than the number of features, the empirical covariance matrix Σ_e is not full rank, and therefore it cannot be inverted. In a recent paper Friedman et al. [5] proposed a method for a sparse, inverse covariance matrix estimation. The matrix $\hat{\Sigma}^{-1}$ is a solution to the maximization problem

$$\hat{\Sigma}^{-1} = \arg\max_{\Sigma^{-1}} \left\{ \log\det \Sigma^{-1} - \operatorname{tr}\left(\Sigma_e \Sigma^{-1}\right) - \rho ||\Sigma_l^{-1}||_1 \right\}$$

where Σ_e is the empirical covariance matrix and ρ is a sparsity promoting hyperparameter. With the inverse covariance matrix estimate in place, the penalty function $\mathcal{P}$ can be easily incorporated into the likelihood function (1) and the maximizing argument can be found using for example gradient descent.

2.3.2 Factor analysis scoring

The final scoring method is based on the factor analysis. By performing factor analysis with k dimensions it is assumed that the N dimensional data can be approximated by using k dimensions only.

$$x^{(i)} = \Lambda z^{(i)} + \Phi$$

where $\Phi \sim \mathcal{N}(0, D)$ and D is a diagonal covariance matrix. The lower the variance of Φ associated with a particular feature, the closer this feature is to being a member of the k dimensional subspace. Assuming that a given stimulus is represented by points on a k dimensional hyperplane, then the

$D_{q,q}$ is inversely proportional to the average distance of the feature q from this hyperplane. The feature score can be therefore given by

$$s_q = \sum_{l=1}^{P} \left| \frac{1}{D_{q,q}} \right|.$$

3 Results

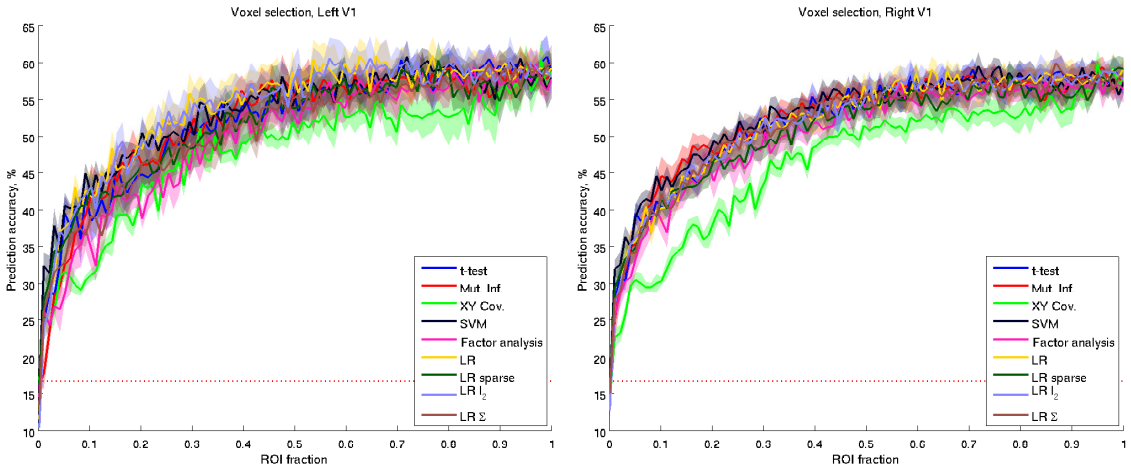
The experiments were conducted on fMRI data, where subjects were shown words at six various eccentricities. Three stimuli were displayed to the left and three to the right of the fixation point. For each subject three experimental runs were performed, each run consisted of 150 time points. In order to limit the computational complexity, the data for two distinct regions of interest (ROI) of only one subject were evaluated. The regions of interest were the early visual areas: left and right V1, each having 444 and 836 features respectively. For each run the time course data for each voxel were normalized to $\mathcal{N}(0, 1)$.

Feature selection algorithms were evaluated using 3-fold cross-validation. One third of the data was used for feature ranking the remaining two-thirds were used as training and test sets in a 5-fold cross-validation. Given a particular voxel ranking, the training and test data sets consisted of n best scoring features only. Due to differences in ROI sizes, n is in fact a fraction of the total number of voxels constituting an ROI. The classification was performed with an SVM classifier with linear kernel [6], and the c parameter selected in a grid search. The remaining hyperparameters ρ and λ were also found using extensive search, however the experimental results did not vary significantly with their choice.

Experimental results are presented in figure 1 where stimulus prediction accuracies for 100 different subsets of best scoring ROIs feature sets of the left and right V1 are given. In each plot a single solid line corresponds to one feature selection method. This line is drawn on top of a colored region, whose boundaries are error bars. The random guess performance of 16.5% is represented by the red dotted line. In all cases feature selection methods allow to achieve an above chance performance with a very small subset of about $\sim 5\%$ of features. For both ROIs, all eight methods exhibit very similar performance, only the feature-output label covariance based scoring is slightly inferior. All curves also demonstrate the law of diminishing returns; increasing the feature count from 5 to 10% has a much bigger impact on prediction than a similar change from 40 to 45%. Selecting more than half of features has virtually no effect on the prediction performance. By comparison using a greedy forward filtering approach, resulted in selecting 0.4 of the left V1 voxels and yielded a 53% prediction accuracy (0.07 and 42% respectively for the right V1). These results are comparable with the ones obtained via different feature scoring methods.

4 Conclusions

This project proposed two new fMRI data voxel selection methods: covariance logistic regression, and factor analysis. In the covariance logistic regression a Gaussian prior on the feature weight distribution is imposed. Since the empirical covariance matrix is not invertible, its approximate sparse inverse is used. The second method, based on the factor analysis, used the random noise variance from this model as indicative of feature relevance. The two methods achieved comparable performance to other commonly used selection mechanisms.



(a) Left V1 (b) Right V1

Figure 1: fMRI data feature selection algorithm evaluation.

References

[1] S. Song, Z. Zhan, Z. Long, J. Zhang, and L. Yao, "Comparative study of SVM methods combined with voxel selection for object category classification on fMRI data." *PloS one*, vol. 6, no. 2, p. e17191, Jan. 2011.

[2] L. Grosenick, B. Klingenberg, B. Knutson, and J. E. Taylor, "A family of interpretable multivariate models for regression and classification of whole-brain fMRI data," *Most*, vol. 94305, no. 650, pp. 1–30, 2011. [Online]. Available: http://arxiv.org/abs/1110.4139

[3] O. Yamashita, M.-a. Sato, T. Yoshioka, F. Tong, and Y. Kamitani, "Sparse estimation automatically selects voxels relevant for the decoding of fMRI activity patterns." *NeuroImage*, vol. 42, no. 4, pp. 1414–29, Oct. 2008.

[4] B. Krishnapuram, L. Carin, M. a. T. Figueiredo, and A. J. Hartemink, "Sparse multinomial logistic regression: fast algorithms and generalization bounds." *IEEE transactions on pattern analysis and machine intelligence*, vol. 27, no. 6, pp. 957–68, Jun. 2005.

[5] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso." *Biostatistics (Oxford, England)*, vol. 9, no. 3, pp. 432–41, Jul. 2008.

[6] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.