Feature Selection for fMRI Classification

Chuang Wu
Program of Computational Biology
Carnegie Mellon University
Pittsburgh, PA 15213
chuangw@andrew.cmu.edu

Abstract

The functional Magnetic Resonance Imaging (fMRI) has provided us with an approach of revealing the activity of brain. Due to the large amount of data in fMRI studies, feature selection techniques are used to select particular features for classifier. In this project, Spectral Clustering is implemented to construct features to achieve best reconstruction of the data and be most efficient for making predictions.

1 Introduction

1.1 Motivation

Over the past decade, a variety of different functional Magnetic Resonance Imaging (fMRI) experiments have been done in order to understand the human brain activity pattern when doing some certain task. By recording the activity pattern of human brain as images of 3D voxel, we are able to visualize the picture of the pattern, find statistical differences in bran activity during different tasks, and a more challenge problem is to train the data with a classifier so as to predict brain activity given any of the pattern, such as whether the human subject is reading a sentence or looking at a picture, or whether the subject is reading an ambiguous or non-ambiguous sentence, etc. Machine Learning is a most powerful approach to train the classifier and then use the classifier to discriminate between different cognitive states.

A typical fMRI experiment produces a three-dimensional image related to the human subject's brain activity every half second. The experiment consists of a set of trials, and the data is partitioned into trails, (reading a sentence, observing a picture, and determining whether the sentence correctly described the picture). There are about 6 human subjects in the data; each of the 40 trials lasts approximately 30 seconds. Only a fraction of the brain of each subject was imaged. The data is marked up with 25-30 anatomically defined regions (called "Regions of Interest"). Each image contains approximately 5,000 voxels (3D pixels), across a portion of the brain.

Learning this series of brain data to an experimental condition label requires many challenges,

one of which is the extremely sparse noisy data with extremely high dimensional features. This would cause the over-fitting problem for the classifier. Hence it is necessary to apply some feature selection method to make learning tractable and prevent over-fitting due to spurious correlations. The objective of feature selection is three-fold: improving the prediction performance of the predictors, providing faster and more cost-effective predictors, and providing a better understanding of the underlying process that generated the data. There are a number of generic feature construction methods, including: clustering; basic linear transforms of the input variables (PCA/SVD, LDA); more sophisticated linear transforms like spectral transforms (Fourier, Hadamard), wavelet transforms or convolutions of kernels; and applying simple functions to subsets of variables, like products to create monomials.

1.2 Goal of this work

The goal of this work is to derive features with Spectral Clustering from the original brain image data as the input for the classifiers to achieve best reconstruction of the data and be most efficient for making predictions. Clustering has long been used for feature construction. The idea is to replace a group of “similar” variables by a cluster centroid, which becomes a new derived feature. The new derived feature then is treated as a representative for the whole cluster as the new input for the classifier. The clustering will greatly reduce the number of features and meanwhile without losing much information. The most popular algorithms include K-means and hierarchical clustering.

The clustering method used in this project is ‘Spectral Clustering’. Spectral methods recently emerge as effective methods for data clustering, image segmentation, Web ranking analysis and dimension reduction. The spectral clustering algorithm is based on the concept of similarity between points instead of distance, as other algorithms do. The implemented algorithm is formulated as graph partition problem where the weight of each edge is the similarity between points that correspond to vertex connected by the edge. The goal of the algorithm is find the minimum weight cuts in the graph, but this problem can be addressed by the means of linear algebra, in particular by the eigenvalue decomposition techniques, from which the term "spectral" derives.

2 Method

2.1 Intuition

Spectral cluster could tell the intrinsic features of the data, revealing the underlying cluster. One of the biggest differences between Spectral clustering and K-mean clustering is that, K-mean requires the initial value of K, i.e. the number of clusters. The initial value of K is kind of arbitrary; in contrast Spectral clustering has a rich structure with interesting properties and deep connections to principal component analysis. Hence the goal of this work is to implement a spectral clustering method to cluster the image data in order to reduce the feature number, construct new features, and improve accuracy of classifier. The code for manipulation and visualization of the fMRI data has been provided, as well as ‘Naïve Bayes Classifier’ and ‘Logistic Regression Classifier’. Hence the work needed to be done in this project is to

implement the Spectral Clustering algorithm, and combine with the existing code to increase the prediction accuracy.

2.2 Description of the algorithms

The algorithm starts with well-motivated objective functions; optimization eventually leads to eigenvectors, with many clear and interesting algebraic properties. At the core of spectral clustering is the Laplacian of the graph adjacency (pairwise similarity) matrix, evolved from spectral graph partitioning.

The detailed steps for the 'Spectral Clustering' in this project are [2]:

1, Constructing a matrix M, in which rows are corresponding to image and the columns are corresponding to voxels.

2, Normalize over the column (if the affinity is defined by Euclidean distance).

3, Construct Affinity Matrix A. The affinity is defined as $A_{ij} = \exp(C(i,j)^2/2\sigma^2)$ if $i \neq j$, and $A_{ii}=0$, where C(i,j) is the correlation between the two vectors of voxels.

4, Define D to be the diagonal matrix whose (i,i)-element is the sum of A's i-th row, and construct the matrix $L = D^{-1/2}AD^{-1/2}$.

5, Find x1,x2,...,xk, the k largest eigenvectors of L (chosen to be orthogonal to each other in the case of repeated eigenvalues), and form the matrix X = |x1,x2,..,xk| by stacking the eigenvectors in columns.

6, Form the matrix Y from X by renormalizing each of X's rows to have unit length (i.e. $Y_{ij} = X_{ij}/(\Sigma_j X_{ij}^2)^{1/2}$).

7, Treating each row of Y as a point in R_k, cluster them into k clusters via K-means or any other algorithm.

8, finally, assign the original point S_i to cluster j if and only if row i of the matrix Y was assigned to cluster j.

Here the scaling parameter σ^2 controls how rapidly the affinity A_{ij} falls off with the correlation between two voxels. Although there is an automatically way for choosing it, I used cross-validation to have a best choice, which will be shown later.

The definition of affinity in the algorithm is critical. The choice of the affinity depends on what property we are looking at the voxels. The affinity is defined as correlation between two voxels in this work. The reason is that, for each voxel, it has a curve of activities over time. Hence, there is correlation between each pair of voxels. If the correlation is high, it means that the two voxels have similar activities throughout different snapshots, which mean the two voxels might have similar functions, or locate in the same Region of Interest, etc. Therefore the two voxels would be grouped into the same cluster. And it is safe to average the voxels within the cluster to construct a new feature.

Given this idea, the matrix M in step (1) is constructed by stacking the data for each voxel throughout the all the images and trials. Hence for a certain subject, the cluster is trying to cluster voxels based on its pattern throughout the whole experiment. As a result, the affinity matrix of voxels is about 5kx5k size, with each element presenting the affinity between the two voxels. Then the spectral cluster will cluster the voxels based on this matrix.

3 Experiment

I am mainly testing the algorithm on subject '05680'. There are 54 trials, 2800 snapshots in the data, and the number of voxels is 5062. Two existing classifiers are provided – 'Logistic Regression Classifier' and 'Naïve Bayes Classifier'. The original accuracy ('original' here means it is got before the feature selection, i.e. there are 5062 features) for the two classifiers are 80% for 'Naïve Bayes Classifier' and 60% for 'Logistic Regression Classifier'.

To combine the cluster results to the classifier. One thing need to do is to figure out the cluster number. Although the Spectral clustering is able to tell the number of clusters by looking into the eigengaps, sometimes the eigengaps are not obvious, so that we need to set the number of clusters manually. A way of choosing cluster number is to use 10-fold cross-validation on a variety of cluster numbers, testing on each epoch and training on the remaining nine. The testing numbers of clusters are chosen as integer times of ROI numbers, i.e. 25, 50, and so on. The result is shown in Figure 1 and Figure 2.

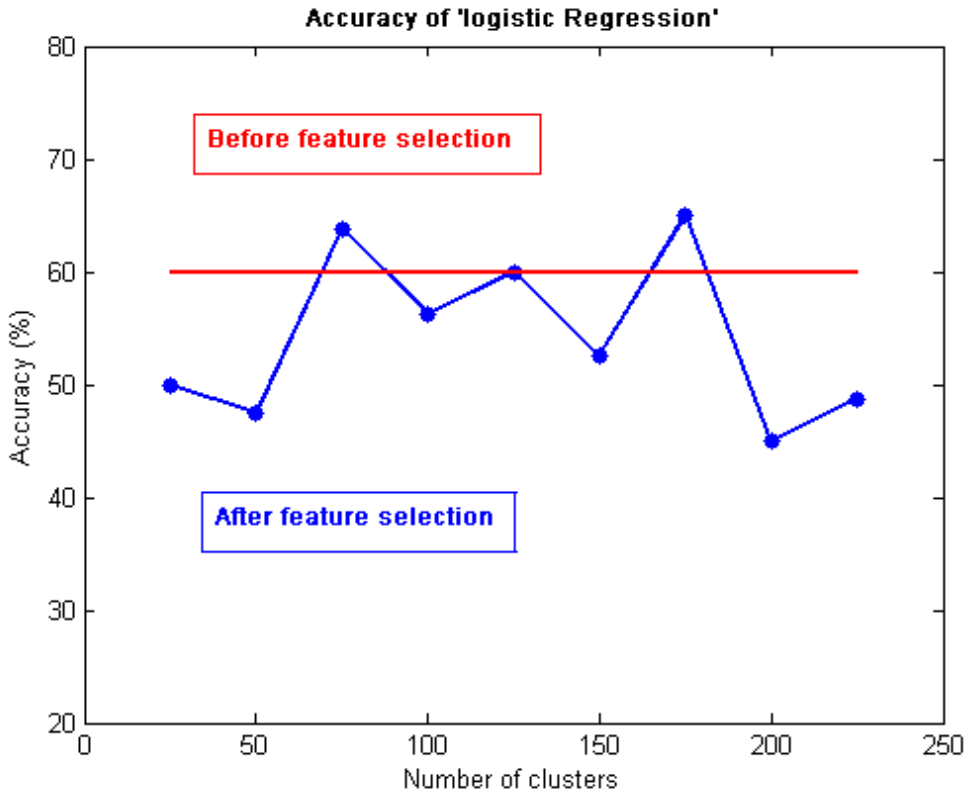

Figure 1. The accuracy of prediction over the number of clusters for 'Logistic Regression Classifier'. The red line is the original accuracy (60%) before the feature selection. The blue dots are the accuracy for the classifier after the Spectral Clustering. The accuracies after feature selection are higher than the original classifier when the cluster numbers are 75 and 125 (63.75% and 65%,

respectively).

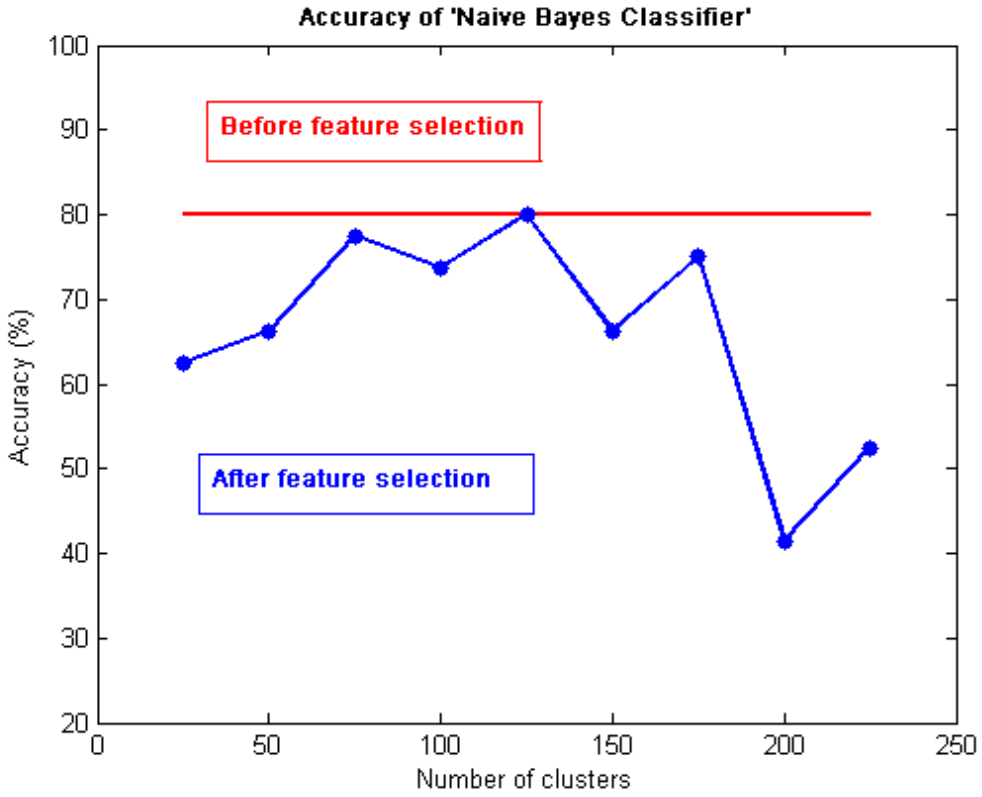

Figure 2. The accuracy of prediction over the number of clusters for ‘Naïve Bayes Classifier’. The red line is the original accuracy (80%) before the feature selection. The blue dots are the accuracy for the classifier after the Spectral Clustering. The accuracies after feature selection are equal to the original classifier when the cluster number is 125.

Figure 1 and 2 show that for ‘Logistic Regression Classifier’ the accuracy is higher or close to the original accuracy when the number of clusters is around 75-175; for ‘Naïve Bayes Classifier’, the accuracy after the feature selection is equal or closer to the original accuracy when the number of clusters is in the range of 75-150, too. The sigma value is chose via cross-reference with the cluster number. Based on the sigma value study (Figure 3 and 4), I chose the 2*Sigma^2 to be 1 to produce high accuracy. And based on the cluster number study here, I chose 125 as the cluster number in the sigma value study below.

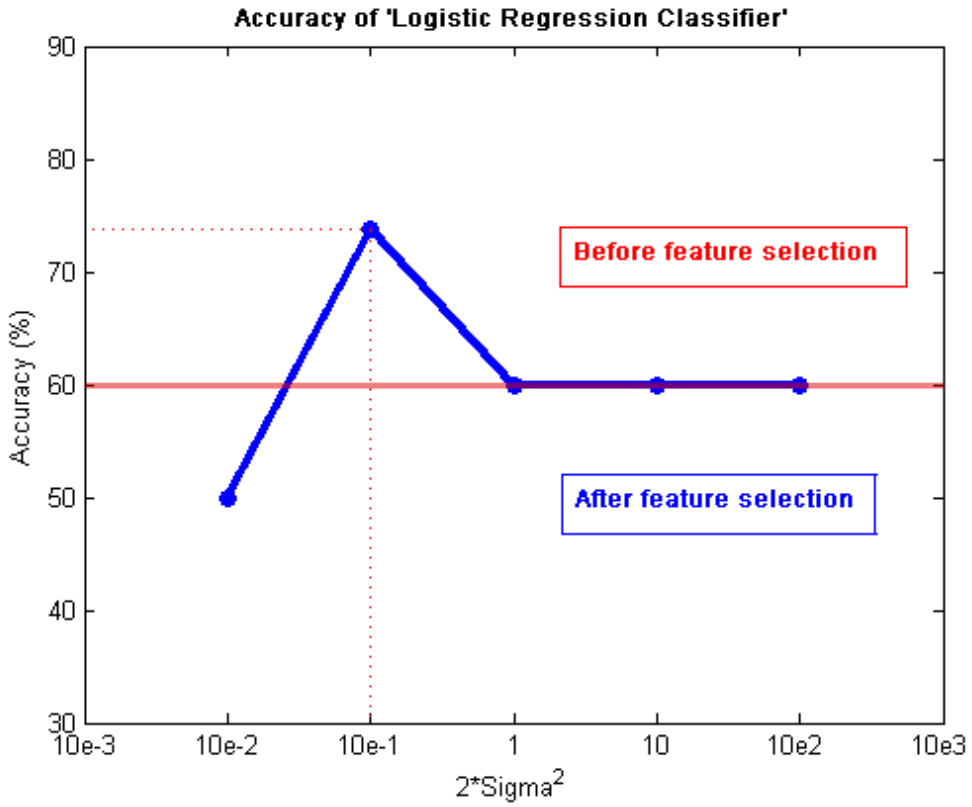

Figure 3. The accuracy of prediction over different sigma value of 'Logistic Regression Classifier'. The blue dots are the accuracy for the classifier after the Spectral Clustering on different sigma values. The accuracy is (73.75%) higher than the original accuracy (60%) when 2*Sigma^2 is 10e-1, and equal when 2*Sigma^2 is 1, 10, and 100. The cluster number is set to 125 given the observation from Figure 1 and 2.

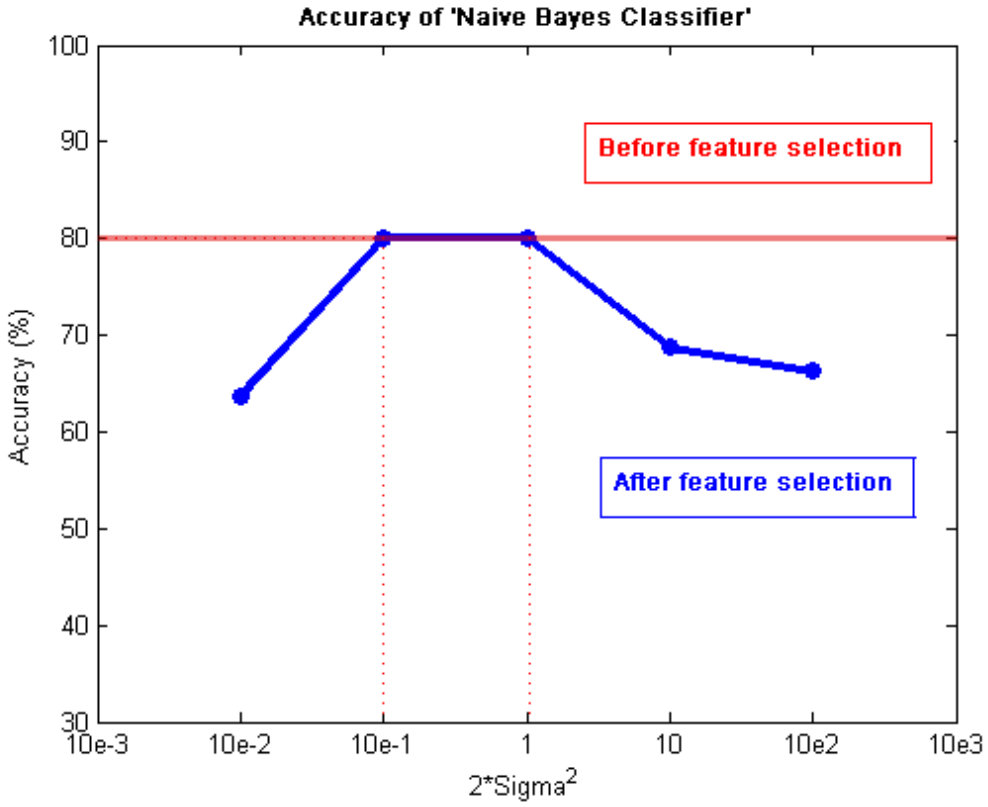

Figure 4. The accuracy of prediction over different sigma value of 'Naïve Bayes Classifier'. The blue dots are the accuracy for the classifier after the Spectral Clustering on different sigma values. The accuracy is (80%) equal to the original accuracy when 2*Sigma^2 is 10e-1 and 1, and lower otherwise. The cluster number is set to 125 given the observation from Figure 1 and 2.

Another interesting question is how to choose the sigma value to define the Affinity. The scaling parameter σ^2 controls how rapidly the affinity A_{ij} falls off with the correlation between two voxels. A higher sigma value will make the affinity value lower, hence the cluster might be not tight enough; whereas a lower sigma value will increase the affinity, and it will make the clusters ambiguity. Although there is an automatically way for choosing it (i.e. pick up the sigma value when the tightest (smallest distortion) clusters are got after clustering Y's rows), I used cross-validation to figure out a best choice (i.e. get highest accuracy).

In Figure 3 and 4, I fix the number of clusters to 125 (based on the results from the 'cluster number' study), and construct the affinity matrix at a variety of sigma values. The results showed that, when 2*sigma^2 is in the range of 10e-1 to 1, the accuracy is higher or equal to the original accuracy.

4 Conclusions

Spectral Clustering is able to do feature selection for fMRI data. It improves the accuracy for 'Logistic Regression Classifier' when certain values of clusters number and sigma are chosen; and it produced the same accuracy for 'Naïve Bayes Classifier' on the original data under the similar conditions. The best results obtained with either of the two classifiers are similar, reflecting that the best results are not generated by chance for just one of the classifiers. The cluster number and sigma values could be chosen by cross-validation. After feature selection, the number of features is set around 125, which is much less than the original number of features 5062. Hence feature selection with Spectral Clustering could greatly reduce the number of features to achieve best efficiency for the prediction. The number of clusters is about 5 times of the number of ROIs, which somewhat reflects that, ROI is not functional coherent. Roughly on average about 1/5~1/3 of the ROIs might have the same/similar function/activity given a certain kind of signal.

References

[1] Wang, X., Mitchell, T. (2002) Detecting cognitive states using machine learning. *Iterim working paper.*

[2] Andrew Y. Ng, Michael J., and Yair W. (2002) On Spectral Clustering: Analysis and an algorithm. *In NIPS 14.*

[3] Jay P. (2005) Understanding Feature Selection in fMRI. *Research Report.*

[4] Pereira F. Rob M. Marcel J. Tom M. Nikolaus K.. (2006) Decoding of semantic category information from single trial fMRI activation in response to word stimuli, using searchlight voxel selection. *Research Report.*

[5] Nikolaus K. Rainer G. and Peter B.. (2006) Information-based functional brain mapping.

PNAS, doi:10.1073/pnas. 0600244103

[6] Isabelle G. Andre E. (2003) An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research 3(2003) 1157-1182.*