NIPS workshop on New Directions on Decoding Mental States from fMRI Data, 12/8/2006

Automated fMRI Feature Abstraction using Neural Network Clustering Techniques

Radu Stefan Niculescu
Siemens Medical Solutions
Computer Aided Diagnosis
Malvern, PA 19355
stefan.niculescu@siemens.com

Tom M. Mitchell
Carnegie Mellon University
Computer Science Department
Pittsburgh, PA 15213
tom.mitchell@cmu.edu

Abstract

In this paper we propose a method to automatically find useful abstractions of the fMRI data using a new neural network clustering technique. The purpose of these data abstractions is to alleviate the computational burden by reducing dimensionality, to minimize the risk of overfitting by reducing the number of free model parameters, and to uncover what relationships among voxels can help explain in what cognitive state a subject is at a given point in time. We show that our method outperforms classical machine learning methods like SVM, GNB and kNN in terms of accuracy.

1 Introduction

In this paper we study the applicability of a new neural network clustering method to the task of automatically finding abstractions of the fMRI data that maximize the accuracy of a given classifier model. The motivation for studying clustering is that activity in the brain usually happens in clusters and rarely only in isolated voxels. Therefore, if we see a set of neighboring voxels that exhibit high activation patterns for one task, we are confident that the activity is relevant while an isolated voxel that is very active may be just the result of noise and it can be discarded. The purpose of abstracting the data is twofold:

1. Reducing the dimensionality of the data. Typically, a snapshot of the brain consists of few thousands voxels and a trial may have tens of snapshots. Therefore, classifiers could perform better in terms of time if the dimension of the feature set is reduced. Additionally, in a typical cognitive task, the number of trials is pretty small and therefore having a model with a lower complexity prevents overfitting.
2. Coming up with a model of what is happening in the brain. For example, hidden layer units in a neural network may tell us that the condition we are looking at depends on the sum of activities of two voxels while each of the two voxels alone may be poor predictors of the condition. More generally, feature abstractions may correspond to what happens in the hidden cognitive states of a specific condition.

Some examples of abstractions include: averaging the voxels in a specific region, compacting a region of the brain to a representation consisting of its few most active voxels, reducing the space granularity of the data using bigger voxels. The model proposed in this paper is essentially different in that it abstracts features based on the hidden nodes of a neural network model. Each of these hidden nodes will summarize the activity in a cluster of neighboring voxels in the brain.

A short description of available clustering techniques is given in Section 2. Section 3 details the model proposed. In Section 4 we introduce our fMRI dataset and we describe the experiments performed. We conclude with a brief summary along with some ideas for future work.

2 Related Work: Clustering Techniques

Partitioning a given set of points into homogeneous groups is one of the most fundamental problems in pattern recognition. It belongs to a class of unsupervised learning problems. By analogy with the information theory concepts, clustering can be viewed as an optimization problem where we look for an optimal representation of a large dataset with a more compact "code" (set of clusters of data points). The functional to be minimized is referred to as distortion, measuring how good our representation is.

There exist two different forms of the problem setup dependent on the data representation. In the first form data points are assumed to have a vector form and clustering means deriving a set of vectors that quantize the data with minimal error. In the second setting, known as pairwise clustering, the data points are characterized by distances between them instead of coordinates, and the dataset structure is hidden in the distance matrix.

Obviously the second setting is more difficult; however in many cases the set of pairwise distances is the only information available. Moreover, sometimes the "distance" function is not a metric (e.g. the triangle inequality does not hold) that makes the task of finding a consistent low-dimensional representation of the data impossible.

Both settings were approached with several different methods and algorithms. The most popular algorithm for the first one is K-means, belonging to the class of EM methods. It is based on the idea of iterative reestimation of optimal "codevectors" (centroids of clusters) interleaved with repartitioning of the dataset based on the new centroid values. The shortcomings of this algorithm are 1) poor scaling, 2) unknown number of clusters (K), 3) convergence to local minima - the iterative process may not find the globally optimal partitioning and 4) vector representation of data is required.

The above methods deal with the vector representation of a dataset. However, sometimes the vector representation of the data is not easy to obtain. Then, the second setting of the clustering problem arises. One of the intuitive approaches to that problem is hierarchical clustering. It is called hierarchical because at each level a single point or group of points is joined with another, in such a way that when combined, the elements remain together throughout the remainder of the procedure. Methods of that type can be agglomerative (clumping) or divisive (splitting). All of them require distance functions for calculating distances between clusters (in order to decide which clusters to merge or to split). Agglomerative methods are usually preferred since they require less computation. Those approaches are described in [1] in detail.

For the readers who want to learn more about clustering, we provide a comprehensive list of references in the end of the paper.

3 Approach

3.1 Goals

The goal of our experiments is to find useful abstractions of the fMRI data based on clusters of voxels. By *useful abstraction* we mean a combination of the voxels that best summarizes the data in a set of voxels in order to maximize the accuracy of a model (a neural network in our case) for discriminating between two cognitve states. Each such subset of voxels will be called a cluster. We restrict a voxel to belong to only one cluster. In our experiments we try to distinguish between the following two cognitive states: "Subject reading a sentence" (class 1) and "Subject looking at a picture" (class 0).

3.2 The Model

Here we propose a new method of feature abstraction that aims at maximizing the accuracy of a neural network. The idea is the following: using a backpropagation style algorithm, we train a neural network where the input vector is the set of all voxels. The neural network will have a number of hidden units equal to the number of clusters we are trying to get. Each cluster summarizes an important feature of the data. Unlike standard backpropagation, at the end of each iteration we will try to enforce the condition that each feature has at most one outgoing edge with a non-zero weight and, based on that weight, we will assign the feature (voxel) to a specific cluster (hidden unit).

Some intuitive conditions that our clustering must satisfy are:

1. Voxels in the same cluster must be close together.
2. If a voxel is assigned to one cluster, then the weight from that voxel to the corresponding cluster (hidden unit) should have big magnitude compared to the weights from that voxel to the other clusters.
3. In order to be important, the clusters should not be too small. In our experiments, we always noticed that the clusters had enough voxels, so we did not experiment with any heuristics for deleting small clusters.

In order to satisfy the above conditions, we decided to assign a feature (voxel) i to the cluster $j = argmax_k \frac{\|w_{ik}\|}{d_{ik}}$ where w_{ik} represents the weight from voxel i to cluster k and d_{ik} is the distance from voxel i to the center of the cluster k. This way we encourage a voxel to be assigned to a closer cluster center and we penalize for small magnitude of the weight (in other words, we prevent a voxel from belonging to a cluster where its weight does not count).

3.3 The Algorithm

1. Initialize the weights with small random numbers. Set learning rate to 0.1.
2. Initialize the centers of the clusters to be all equal to the center of mass of the voxels.
3. Run stochastic backpropagation for each sample in the training set.
4. For each input feature i, find the hidden unit $j = argmax_k \frac{\|w_{ik}\|}{d_{ik}}$.
5. Set $w_{ik} = 0$ for all $k \neq j$, assign voxel i to cluster j and recompute the center of the cluster j.
6. Compute the error on the early stopping set. If it is smaller than before, save the current clustering and weights. If it is larger, check if there was no improvement in

the last fixed number of epochs (1000 by default) and, if so, GO TO 7. Otherwise GO TO 3.

7. Output clustering. Report accuracy on the validation set.

4 Experiments

We ran our experiments on the StarPlus dataset. The StarPlus experiment was designed to engage several different cortical areas, in order to look at their interaction. In this dataset a subject first sees a sentence(semantic stimulus) for 4 seconds, such as "The plus sign is above on the star sign.", then a blank screen for 4 seconds, and finally a picture (symbol stimulus) such as

$$\frac{+}{*}$$

for another 4 seconds, during which the subject must press a button for "yes" or "no", depending on whether the sentence matches the picture seen or not. Snapshots of the brain were taken every half second. The subject is instructed to rehearse the sentence in his/her brain until the picture is presented rather than try to visualize the sentence immediately. The second variant switches the presentations of sentences and pictures, and the instruction is to keep the picture in mind until the presentation of the sentence.

In this dataset there are three main conditions: *fixation* (the subject is looking at a point on the screen), *sentence followed by picture* and *picture followed by sentence*. We have 10 trials in *fixation* and 20 in each of the other conditions. For each trial, we have 32 time snapshots of the brain.

There are two main drawbacks with running our experiments on multiple subjects. First, different subjects have different brain shapes and number of voxels in each ROI. Second, different subjects exhibit different activation patterns for a given task. Therefore we decided to ran our experiments on a per subject basis. Here we report results only for one of the subjects in our study, but same method can be employed for other subjects in order to cluster their voxels for the purpose of feature abstraction.

For each subject, a handful of ROIs are available. We could run our experiments on each of them separately or on a subset of ROIs. Since our previous classification results for this picture-sentence study were very accurate when *CALC* was used, we decided to use this ROI for the experiments in this paper. However, for other studies we may need to try our method on other ROIs or even the full brain in order to obtain decent classification accuracy.

In our experiments we trained neural networks using backpropagation with a learning rate of 0.1. We varied the number of hidden units (clusters) from 2 to 5. Only one output unit was used and an output greater than 0.5 was predicted in *class 1* while an output between 0 and 0.5 was predicted in *class 0*. The weights of the network were randomly initialized with values of magnitude less than 0.1.

As stated in section 3.1, our goal is to distinguish whether a subject is looking at a picture or is reading a sentence. Remember that for our subject we have 20 trials when the sentence is presented first. Each such trial consists of 32 time snapshots, one every half second. The snapshots $1-16$ correspond to the period of sentence processing while the last 16 correspond to the picture segment. Previous studies showed that time slices 11(sentence) and 27(picture) yield the best accuracies when used to discriminate between sentences and pictures on the joint dataset (20 trials with sentence presented first and 20 trials with picture presented first). Therefore, for our first experiment, we decided to use these two time slices in the 20 trials when the sentence was presented first. This leaves us with 40 examples (20

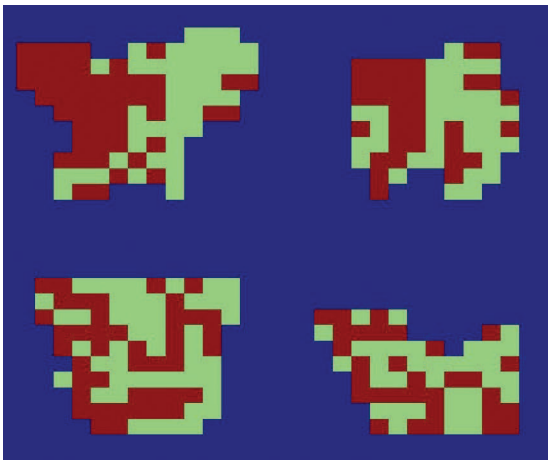

Figure 1: Clustering of CALC (318 voxels) in two clusters using 320 examples. The accuracy was approximately 94%.

sentences and 20 pictures). In order to train the neural network, 30 examples were used and the validation set consisted of 10 examples. We obtained perfect accuracy when we learn at least two clusters.

In our second experiment, each trial supplied 8 sentences (slices 5-12) and 8 pictures (slices 20-278). This leaves us with 320 examples (160 sentences and 160 pictures). Out of these, 240 were used as a training set and the rest as a validation set. The classification accuracies are at least 90% in this case and we observed a decrease in performance with increasing number of clusters. This may seem wrong at a first look because one may argue that a neural net with three hidden units can learn what one with two hidden units can. However, given the increase in number of parameters, there is a higher chance of overfitting and therefore the accuracy on the validation set does not improve with increasing number of hidden units.

Classifier \ Dataset	*40 Examples*	*320 Examples*
ANN (2 clusters)	1.00	0.94
ANN (3 clusters)	1.00	0.93
ANN (4 clusters)	1.00	0.90
ANN (5 clusters)	1.00	0.90
GNB	0.90	0.875
SVM	0.875	0.83
3NN	0.875	0.77

Table 1: Accuracies of different classification methods.

We also compare our results with other classic machine learning techniques: Gaussian Naive Bayes, Nearest Neighbor, Support Vector Machines. In Table 1 we report the accuracies of four fold cross validation for these classifiers. These results show that the clustering of the voxels helped neural networks outperform all other classifiers in terms of accuracy. In addition, the abstraction found at the level of each hidden unit provides the basis for automated feature reduction that extracts the best summary of a cluster of voxels.

Figure 1 presents the results of such a clustering of the visual cortex using a neural network with two hidden units and a number of 320 examples (240 in the training set, 80 in the

validation set). The two clusters are colored in green and brown. The figure contains the four brain slices corresponding to CALC for the subject in our study.

5 Future Work

In this study we experimented with only one dataset, one subject and only one classification task. One direction for future work would be to run our method on other subjects in the StarPlus study and check if the clusterings we obtained are similar. Similar clusterings across subjects may point to the fact that there are hidden cognitive states that everybody goes through when performing a more complex cognitive state. However, it is a challenge to decide if clusterings for different subjects are similar.

It would also be interesting to try other classification tasks on other datasets. Currently we also have access to a dataset about semantic categories and to a syntactic ambiguity dataset. In the semantic categories study we have the opportunity to try a twelve way classification task as opposed to a binary task.

6 Conclusions

In this paper we studied the applicability of a new neural network clustering method to the task of automatically finding abstractions of the fMRI data that maximize the accuracy of a given classifier model. We showed that our method outperforms classical machine learning methods like SVM, GNB and kNN in terms of accuracy. In addition, the abstraction found at the hidden layer level provides an efficient way of doing feature reduction that allows us to accurately discriminate between two different cognitive states.

7 Acknowledgement

While a student at Carnegie Mellon University, Radu Stefan Niculescu was supported by a Graduate Fellowship from the Merck Computational Biology and Chemistry Program at Carnegie Mellon University established by the Merck Company Foundation and by National Science Foundation (NSF) grant no. CCR-0122581.

References

[1] Duda R. Hart P. (1973). Pattern classification and scene analysis. John Wiley & Sons.

[2] Pelleg D. Moore A. (2000). X-means: Extending K-means with Efficient Estimation of the Number of Clusters

[3] Rose K. (1998) Determenistic annealing for Clustering, Compresion, Classification, Regression and related optimization problems. Proceedings of IEEE,86(11), 1998

[4] Hoffmann T. Buchmann (1997) Pairwise data clustering by deterministic annealing. IEEE Transactions of PAMI, 19(1):1-14, 97

[5] Slonim N. Tishby N. (1999) Agglomerative Information Bottleneck. In Proc. of Neural Information Processing Systems

[6] Pereira F. Bialek W. Tishby N. (1999) The information bottleneck method. In Proc. of the 37-th Allerton Conference on communication and Computation

[7] Slonim N. Tishby N. (1999) Document clustering using word clusters via the information bottleneck method.

[8] Bradley P. S. Fayyad U.M. (1998) Refining initial points for K-Means clustering. Proceedings of the 15-th International Conference on Machine Learning.

[9] Jain A.K., Murty M.N., Flynn P.J. (1999) Data Clustering: A Review. In ACM Computing Surveys, Vol 31, No 3

[10] Focardi S.M. (2001) Clustering economic and financial time series: Exploring the existence of stable correlation conditions.

[11] Oates T., Firoiu L., Cohen P.R. (1999) Clustering time series with Hidded Markov Models and Dynamic Time Warping. Presented at IJCAI-99 Workshop on Sequence Learning

[12] Smyth P. (1997) Clustering sequences with Hidden Markov Models. In Advances in Neural Information Processing Vol 9 (1997)

[13] Mitchell T. (1997) Machine Learning. McGraw-Hill.