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. 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 are the following: 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 this hidden nodes will summarize the activity in a cluster of neighboring voxels in the brain.

2 Approach

In this paper 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 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 Results

We compare our algorithm with other classic machine learning techniques (Gaussian Naive Bayes, Nearest Neighbor, Support Vector Machines) on a dataset describing a cognitive task involving reading sentences and looking at pictures. Each example is a snapshot of the brain of a given subject. In Table 1 we report the 4 fold cross-validation results for the visual cortex. Our algorithm performs better presumably because it is able to incorporate the domain knowledge assumption that there are clusters of voxels that act together to perform a common cognitive function. None of the other models makes this assumption.

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
GNB	0.90	0.875
SVM	0.875	0.83
3NN	0.875	0.77

Table 1: Accuracies of different classification methods.