fMRI

Josh Willis
University of Tennessee
jwill221@vols.utk.edu

Sadra Hemmati
University of Tennessee
shemmati@vols.utk.edu

William Halsey
University of Tennessee
whalsey@vols.utk.edu

Arvind Ramanathan
Oak Ridge National Lab
University of Tennessee
ramanathana@ornl.gov

Abstract

Functional magnetic resonance imaging (fMRI) is an imaging technique that provides valuable insights on the structure and functionality of the brain. The images resulting from fMRI scans are usually analyzed to gain insight into brain activity as a subject performs a task. This information can be analyzed to potentially classify patients or find causality structures in the brain. We have used SVM to classify patients as either schizophrenic or non-schizophrenic using their fMRI images. We show that the analysis of complex fMRI data on a large scale is possible using Apache Spark and Amazon Web Services.

1 Introduction

Functional magnetic resonance imaging is a non-invasive technique for studying brain activity. In a fMRI scan, a subject performs a task experiment while images are taken. The task experiment is designed to activate neurons in the brain. For example, the subject might tap their fingers or perform an auditory test. Neuron activity can be seen by examining the oxygenation of the blood. Each scan consists of hundreds of images over time and each image can consist of 100,000 voxels. Hence, the amount of data for a single patient is large and is on the big data scale for multiple patients.

The process of task experiment and blood oxygenation analysis is known as blood-oxygen-level dependent (BOLD). It is important to note that there is a delay between when a task occurs and when the oxygenation of the blood can be see. For example, 1 shows the delay of blood oxygenation levels after a task is performed.

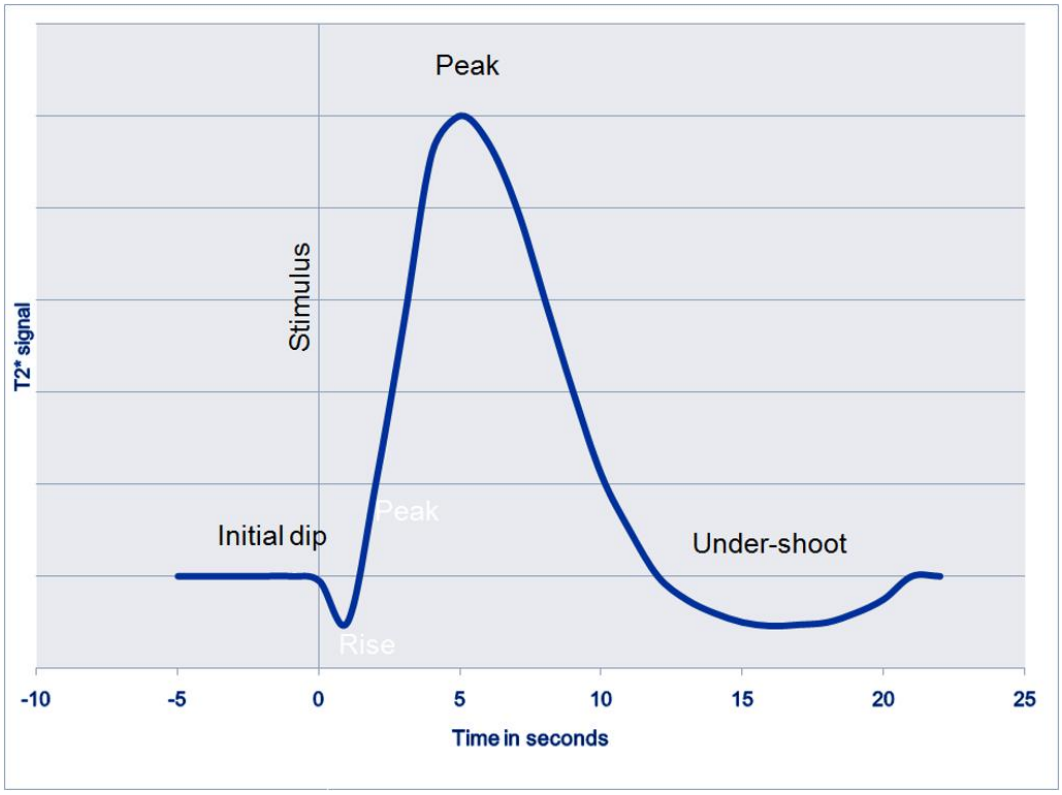

Figure 1: You can see here that the activity response in BOLD occurs approximately 4-6 seconds after the task experiment. Case courtesy of Dr Frank Gaillard, Radiopaedia.org.

Typically fMRI data is analyzed to either find regions of connectivity in the brain or classify patients. An analysis of regions of connectivity helps identify what regions of the brain are correlated. For example, methods such as Bayesian networks, LinGAM, and Independent Component Analysis (ICA) have been used on a smaller scale to analyze the connectivity of the brain. Classifying patients can be useful when identifying diseases and features of the brain between patients. For example, one might be interested in identifying a disease in a patient based on previous patients fMRI scans. From our research, little to no analysis has been performed on brain connectivity or classification for fMRI data on a big data scale.

The analysis of fMRI scans is important because it can identity correlated regions in the brain and can classify patients based on brain activity. There is growing interest in using fMRI data for classification of mental disorders and predicting early onset diseases such as autism, schizophrenia, and Parkinsons disease. In addition, gigabytes of fMRI datasets are openly available from OpenfMRI and terabytes of fMRI datasets are available at Oak Ridge National Laboratory (ORNL). These large repositories of datasets are waiting to be explored and researched. By proving that the analysis of this data can be performed on a large scale, we hope to influence others into analyzing fMRI data for disease classification and brain study. We are interested in using the BOLD information from a large amount of fMRI scans to classify patients as schizophrenic or non-schizophrenic.

2 Data

We gathered our data from an OpenfMRI schizophrenic dataset. OpenfMRIs datasets are broken into compressed groups of patients. Each patients dataset includes BOLD fMRI images for three letter-n-back task experiments. For our analysis, we used the first task experiment from each patient. In addition, the label of the patients could be found in the metadata of the OpenfMRI dataset.

OpenfMRIs schizophrenic dataset holds fMRI scans for 102 patients. Each patients BOLD NIfTI file is approximately 19MB. Therefore, the total compressed dataset is approximately 1.9GB. We examined 27 patients data; hence, our compressed dataset is approximately 513MB. However, the amount of data we dealt with grew after we preprocessed our data.

2.1 Preprocessing

Apache Spark and, in general, MapReduce requires that data be present in an ASCII format. Therefore, before we analyzed the fMRI images, we converted the binary compressed NIfTI file format to a comma separated value (CSV) file format. We used the Python module, NiBabel, to open and extract our desired information from the NIfTI file. Specifically, we removed the time step, x, y, and z coordinates of each voxel, and voxel intensity from the fMRI images and prepended the class of the patient to each voxels information. The class of the patient was passed into our preprocessor after it was retrieved from the OpenfMRI metadata file.

It is important to note that an ASCII representation of a dataset is exponentially larger than the binary representation of the dataset. After a BOLD NIfTI file is converted to a CSV it is approximately 400MB. Because we are working with 27 patients, we dealt with approximately 11GB worth of data. If we had processed the entire OpenfMRI dataset, we would be dealing with approximately 41GB worth of data.

In addition to readying the data for the data processing platform, one would also be interested in removing noise in the data. Because patients could move during scans, movement noise could be present in the OpenfMRI dataset. In order to alleviate this movement noise, one might normalize the dataset to remove the abnormalities.

3 Analysis

We decided to analyze our fMRI data collected from OpenfMRI to classify fMRI patients as schizophrenic or non-schizophrenic. Support vector machines (SVM) are binary classifiers; hence, the SVM model naturally applies to our problem.

SVM is a supervised learning algorithm used for classification and regression. Given a set of training set with binary labels, an SVM algorithm builds a binary classification model. The model is built so that a hyperplane will separate the data its appropriate class with the least amount of error. SVM is a low bias learning method and has few assumptions on the distribution of the data. Due to its use of kernel, it can perform a nonlinear data separation and can, hence, be used on datasets that are not linearly separable.

4 System

To process the large amount of fMRI data and execute an SVM analysis, we used a system consisting of Apache Spark and Amazon Web Services.

4.1 Apache Spark

To process our fMRI data on a large scale, we decided to take advantage of Apache Spark, a popular distributed data processing platform, because of its quick performance and trivial cluster creation. Apache Spark is an abstraction layer on top of MapReduce, MPI, or another distributed parallel system; it allows for the parallel computation of programs across multiple computers in a cluster. Where MapReduce struggles with algorithms that iterate, Apache Spark shines. Apache Spark allows the application programmer to specify when data persists. In other words, the application can specify for data to reside in distributed memory after it is used so that it can be used by further iterations. In MapReduce, the data would have to reloaded into distributed memory.

In addition to Apache Sparks performance, we also wanted to exploit its cluster creation. Apache Spark provides an executable that allows the user to create a Amazon Web Services (AWS) cluster from the command line. Therefore, we decided to host our cluster on AWS.

4.1.1 Apache Spark MLlib

Apache Spark MLlib is Apache Spark's machine learning library. It supports common machine learning algorithm types, such as: classification, regression, and clustering. It supports the SVM algorithm that we used for analysis.

4.2 Amazon Web Services (AWS)

AWS is Amazons web service system. It allows the user to rent computing resources such as servers, cloud storage, databases, network, and more. Amazons Elastic Compute Cloud (EC2) service provides server instances, which we used for our clusters nodes. Their cloud storage service, Amazon Simple Storage Service (S3) was used for the storage of the ASCII (CSV) formatted fMRI data. Amazons S3 integration, the AWS CLI, with EC2 server instances made it easier to get the data onto the cluster.

We had five server instances in our cluster, one master node and four slaves as can be seen in Figure 4.2. The server instance type used in our project, m3.xlarge cost $0.28 an hour. We had 11GB worth of data to store in our cluster. Therefore, we stored the data in S3 and transitioned it over to the cluster. S3 is relatively cheap and only costs $0.03 per GB per month.

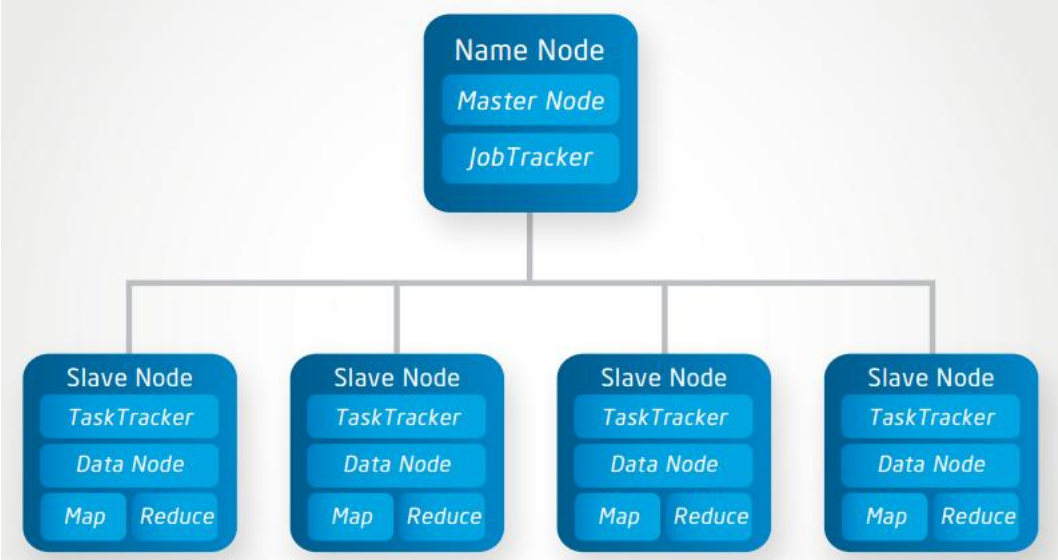

Figure 2: We have four slaves and a master node in our cluster. Picture courtesy of rosebt.

4.3 Troubleshooting

We had several issues while trying to setup our system. We hope by highlighting them here, future readers will be aware and can avoid the same issues.

In order to create our cluster on AWS, we used Apache Spark's EC2 cluster creation script. The cluster creation script defaults the region for the cluster to be us-east. However, our AWS account was located in the us-west region. Therefore, when we tried launch our cluster it unexpectedly failed.

After launching our cluster, we had to move the data to the cluster. Apache Spark's EC2 cluster creation script named our master nodes and slave nodes appropriately, so we could find our master node in the web EC2 Dashboard Interface. When we tried to ssh into the master node from the ssh command given in the dashboard, we were unable to. This meant we were not able to sftp into the machine to transfer the data over. Therefore, we had to circumvent ssh; we stored the data which is on Amazon's S3 cloud storage and were able to retrieve it from there.

We found that Apache Spark took a considerable amount of time to place our data across the cluster. In our first attempts, Apache Spark was unable to place our data in the cluster. Therefore, we kept upgrading our server instances until it was able to place the data on the cluster.

5 Results and Cost

After moving the data to the cluster, we separated each patient into either the training or testing set. For simplicity, we declared the first 20 patients the training set and the last 7 as the testing set. However, the testing set did not contain any non-schizophrenic patients. In the future, we would use k-fold cross validation to select our training and testing sets.

Once our AWS cluster was configured and setup, we were able to run our SVM classification on our 11GB worth of fMRI data. Classification took over 24 hours, but we were able to achieve 99% accuracy on a voxel basis. We find this accuracy to be misleading for several reasons: we classified on a per voxel basis; our testing set mistakenly did not include any non-schizophrenic patients; and, we had a small number of patients, samples, in our dataset.

Based on the computation time and cost of our server instance type, we determined that we spent approximately $22 on our classification.

6 Conclusion

We have shown that fMRI data can be analyzed at a big data scale using Apache Spark and Amazon Web Services. We discussed our results, their validity, the financial cost of using AWS, and the computation time required for analysis. Finally, we discussed the importance of large scale fMRI analysis.

References

[1] Ramanathan, Arvind. *COSC 526 Lectures*. University of Tennessee, 2015. Web. 23 Apr. 2015. `http://web.eecs.utk.edu/~aramanat/projects.html`.

[2] Lindquist, Martin. *Statistical Analysis of fMRI data, coursera online video lectures*, `http://coursera.org`.

[3] Shimizu, Sohei. *LiNGAM method* `https://sites.google.com/site/sshimizu06/lingam/~`.

[4] OpenfMRI.org. *Schizophrenic and Normal sunjects pictures* `https://openfmri.org/m/~`.