

# Decoding High-Dimensional Brain Imaging Data

Rebecca Davidson, Graham Lowe, Patrick Winters

May 12, 2008

## 1 Introduction

The inspiration for our final project comes from work done at CMU between their neuroscience and machine learning labs. Specifically, Tom Mitchell et al.[1], published publicly available fMRI data[4] for classification experiments. Their initial results confirmed that it's possible to classify a subject's cognitive processes from his or her respective fMRI data, and suggested the feasibility of training cross subject classifiers.

### 1.1 Outline

fMRI data suffers from issues known to cause problems for machine learning algorithms. A review of these issues is found in Section 2. A more in depth look into our dataset and its complications is given in Section 3. An outline of our experiments and procedures can be found in Section 4. For the results of our experiments see Section 5. We provide our conclusions in Section 6, and, finally, we offer some thoughts on the applications and use of these experimental tools in Section 7.

## 2 Background

### 2.1 fMRI and Machine Learning

Machine learning applied to fMRI data seems to be a relatively new application of this burgeoning field. Typically neuroscientists use a variety of statistical methods as tools for analysis, but machine learning techniques offer a greater degree of accuracy.

### 2.2 High-Dimensional Data

A review of feature selection and extraction methods is beyond the scope of this paper, but we read and experimented with a number of algorithms. Without a large amount of examples to cover the decision surface, methods like SVM's suffer performance losses with extremely high-dimensional data.

Machine learning techniques for performing feature extraction and selection in image, video, and gene processing abound. It's been shown that these methods are useful in improving classifier accuracy for some tasks, but can serve useful functions in their own right. For instance, feature selection methods are often employed to discover correlations in genetic markers for disease risk prediction[3]. Similarly, information about correlations between areas of brain activity should prove useful for neuroscience research.

## 3 Data

The experiments described by the data consist of a number of trials and observations. Subjects were asked to lie in an fMRI machine and perform two tasks: reading a sentence and viewing an image. The order of the tasks was varied, and each trial lasted a number of seconds. Rest periods between tasks ensured latent activity didn't affect any subsequent observations. The subject would view a sentence, rest for a moment, and then be shown a related image, or the order would be reversed. Data is provided for six human subjects.

### 3.1 Voxels

The fMRI data consists of a video of a number of activation levels for points in the brain called voxels. The voxels in a typical fMRI study have a volume of a few tens of cubic millimeters. Each three-dimensional “image<sup>1</sup>” or “snapshot” of a video sequence consists then of approximately 5,000 voxels, although the number varies per subject. This is already a high-dimensional feature vector, but when video sequences are used as examples instead of instantaneous images, the amount becomes troublesome. Furthermore, it’s assumed that a good deal of noise occurs if a subject shifts or moves during a trial. Complicated noise and correlations between features is assumed, however, the data is heavily preprocessed and normalized by standard fMRI software. A two-dimensional representation of an instantaneous snapshot is given in Figure 1.

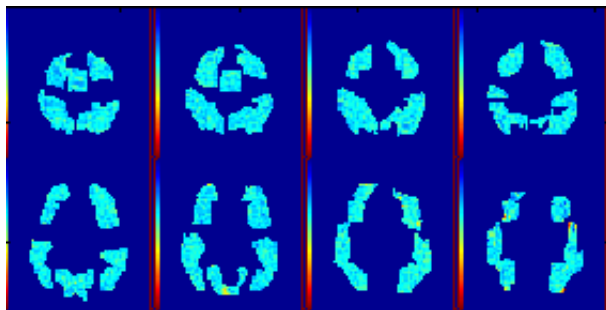


Figure 1: Visualization of a Single Instantaneous Snapshot.

### 3.2 Examples

Each subject was observed performing two tasks for a number of different trials. The data can be thought of as a video sequence of a 16 second trial split in half when the task switched. Thus, each labeled sequence is 8 seconds long, with a temporal resolution of 0.5 seconds; each labeled example is 16 images long and the classes are split evenly. Mitchell et al.[1] flattened 8 second sequences into single examples which

<sup>1</sup>Each image is not square, and is represented with activation values and associated 3-dimensional coordinates

# Images per Example	Length of Feature Vector	Total # of Examples
1 image	4949	1280
4 image	19796	320
16 image	79184	80

Table 1: Different Example Sizes and Data Specifics for Subject 1.

resulted in an extremely high feature vector (approximately 80,000 columns).

We experimented with the ability to split each video sequence into smaller parts, which multiplies our number of examples and reduces the dimension of our feature vectors at the cost of temporal information. “Example size” will from here forward refer to the number of images put into one example.

One can see in Table 1 that the number of examples is quite low, while the size of the feature vector (the # of Voxels per example) is quite high. By reducing the length of each video sequence, we can increase the number of examples at the cost of temporal information.

It was possible to split the parts such that images were reproduced in different orders within multiple examples, but we abandoned this method when we found this approach self-defeating. With a limited number of examples, the highly correlated features cause poor performance. It may be interesting to consider this approach if a larger dataset is available, as it shares justification with the virtual support vector methods proposed to handle invariances in data[2]. It should be noted that the virtual support vector or kernel jittering approach was considered to handle noise and translation invariance, but the complicated structure of the data made this approach infeasible.

## 4 Experiment

The initial purpose of our experiment was to improve classification accuracy over Mitchell et al.[1], but lacking sufficient details about their experiment we avoided comparisons. After performing standard kernel and example size tests, our early results steered the project toward feature selection methods. In or-

der to improve classification accuracy we looked to dimensionality reduction methods, devising our own strategies and benchmarking the standard and simple Fischer Score algorithm.

## 4.1 Single Subject

We began our experiment training classifiers and devising algorithms to work with a single subject's data. We trained classifiers and devised feature selection methods. Our methods follow.

### 4.1.1 Classifiers

Mitchell et al.[1] found strong classifiers such as SVM's to work best on average for the classification task. We decided to focus heavily on using these. In addition, we attempted separating examples into smaller video sequences: 1 image, 4 images, and 16 images. Justification for this was offered in Section 3.2.

We performed searches for optimal parameters on all subjects and example sizes for three SVM kernel types: linear, polynomial, and rbf. The results are described in Section 5.1.

### 4.1.2 Feature Selection

In an effort to reduce the dimension of our feature vectors, we employed methods to select the important voxels for task discrimination. Previous work by Mitchell et al.[1] showed marked improvement using a handful of different feature selection and averaging schemes. The feature selection results are described in Section 5.2, and our methods are described below.

We tested feature selection accuracies by selecting the  $n$  best features and performing cross-validation on examples produced with only those features present. We ranged  $n$  on a log-scale, and plotted accuracies for each size. We also performed this for a range of example sizes.

### Boosting

We performed rankings using boosting with an adaptation of adaboost. This method, while computationally more expensive, offered the ability to improve

rankings by running more iterations additively. The algorithm loops the following for any number of iterations:

1. Randomly Split the features into two sets.
2. Train a classifier on each set of features.
3. Boost the classifiers for a fixed number of iterations.
4. For each feature in a set, add to its score the final weight of its boosted classifier.

Modification of the algorithm could involve splitting the features into more than two sets. It's not clear that it would afford any speed or ranking improvement, but splits could be generated with bias.

### Cross-Validation

This algorithm is similar to feature selection with boosting, but uses a different scoring method. The following steps are modified.

3. Perform cross-validation on the classifiers.
4. For each feature in a set, add to its score the cv accuracy of its classifier.

### Fischer Score

We discovered and applied F-Score to the data-set, with a few reservations. F-Score attempts to rank a feature's discriminative ability independent of other features. This denies us the ability to take into account correlations between features, but has the advantage of being extremely fast.

### Most-Discriminative

As with F-Score, this method ignores feature correlations. We trained SVM's on each individual feature and ranked them according to their independent cross-validation accuracy.

While the boosting and cross-validation methods were computationally expensive, they afford additive scoring. Since the feature splitting is performed randomly at each iteration, the rankings can be updated

with additional scores as time and computation permits. We attempted to account for correlations between features with these approaches, but with high-dimensional data, attempting to test all enumerations of features would be impossible. Therefore, we present these heuristics.

## 4.2 Cross Subject

For cross-subject classification it was essential we discover an informative feature space for all subjects. Since the number of voxels per subject varied and the voxel spatial coordinates weren't the same, it was necessary to transform each subject's examples into a common space. We decided to use our feature selection results to select the  $n$ -most discriminative features per subject, in order of rank, to project all examples onto a space in  $\mathbb{R}^n$ . The results of this experiment are explained in Section 5.3.

For cross-subject testing we trained on all but one subject's data, holding the final subject for testing. We performed this allowing each subject to participate as test set and averaged the results. We refer to this as leave one subject out error.

## 5 Results

We've decided to leave out in-depth comparisons with Mitchell et al.[1] because they worked with different amounts of data<sup>2</sup> and we don't have adequate detail about their experiments. But, our classifiers and feature selection methods seem to have outperformed their initial results.

Experiments with example size found single images to work very well in comparison with full video sequences. We posit this is the result of having more and lower dimensional samples.

Our custom feature selection methods do not show an improvement in error rate, and we hypothesize that they are simply too computationally expensive to be run for the necessary time to see positive results. But, one feature selection method did perform incredibly well and that was F-Score. With F-Score

<sup>2</sup>Mitchell provided data for only the best performing subjects.

we managed to achieve bewildering speed and accuracy in our predictions.

### 5.1 Classifiers

Table 2 details the average single subject error rates and support vector numbers for the family of kernels we experimented with. All kernels performed better with single image examples, but our results with feature selection suggest this is only due to the greater number of examples.

SVM Kernel	# Images	Error	SV
Linear	1	0.24±0.03	1034±31
RBF	1	0.21±0.03	1272±7
Polynomial	1	0.22±0.04	1195±124
Linear	4	0.27±0.04	321±1
RBF	4	0.45±0.04	321±0
Polynomial	4	0.25±0.04	321±1
Linear	16	0.31±0.07	80±0
RBF	16	0.48±0.03	80±0
Polynomial	16	0.25±0.14	80±0

Table 2: Average Classifier Details with 5-Fold CV

It can be seen that the number of support vectors is quite high, almost the number of examples. Due to the high-dimensionality of the data, and the surplus of noisy, useless features, SVM's require almost all of the examples to optimize.

### 5.2 Feature Selection

A comparison of feature selection methods over examples of size 1 is shown in Figure 2. The average accuracy is plotted for a linear kernel using the first  $2^n$  features ranked by each method. It's clear that F-Score performs best for our task, but our tinkering suggested that at least the boosting method's performance improved slightly with additional iterations. For Figure 2 we ran the CV feature ranking for 100 iterations and the Boosting feature ranking for 100 iterations. Each took at least a day to compute. After boosting for an additional 400 iterations, F-Score and our Boost rankings agreed on 2 of the top 5 features, and interestingly ranked the same feature as

number one<sup>3</sup>.

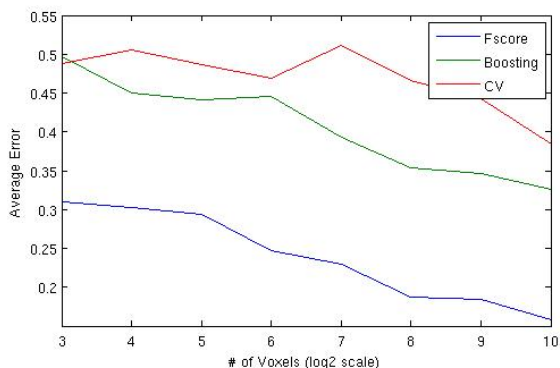


Figure 2: Feature Selection Comparison on Examples of Size 1.

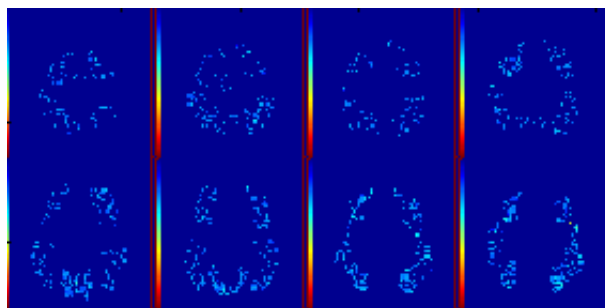


Figure 3: Visualization of a Single Instantaneous Snapshot with F-Score's Top 1024 Voxels.

Focusing more heavily on F-Score we plotted the feature selection results for different example sizes with a linear kernel. To our surprise, our classifiers performed best on the large video samples with a small number of selected features. This implies the classification task is simple, and our SVM's are capable of distinguishing between the two states with a minimal amount of information. We assumed there must be a "sweet spot" of the brain that easily distinguishes between the two cognitive states, but a visualization of the selected features in Figure 3 shows otherwise. The selected voxels appear randomly distributed across the brain, which may indicate classifier accuracy is only improved by the dimensionality reduction of the selections.

The benefit of using long video sequences with few features indicates the classifier is learning the fMRI experiment's temporal properties very well. This is an unintended and perhaps negative side effect, but still highlights the usefulness of feature selection. A plot of F-Score's results using different example sizes is in Figure 4.

While the classifiers still perform better than average on single image (instantaneous) examples, the feature selection methods aren't as useful. We hypothesize that instantaneous examples have a higher

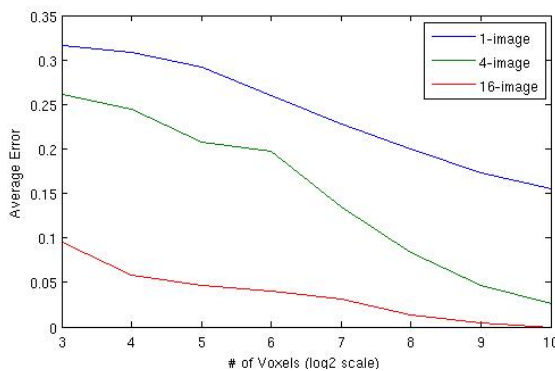


Figure 4: FScore Results for Different Example Sizes.

<sup>3</sup>But those were the only two in the top 50

degree of correlation between features that is not accounted for by F-Score’s independent feature rankings. Although improvements can be made in classification accuracy, naive feature selection methods are confused by the large range of in-class values for a given feature.

### 5.3 Cross-Subject

In section 4.2 we described our rationale for using feature selection to produce a common feature space for all subjects, and our results elected F-Score to select our features. We used F-Score to select features, and combined subject’s examples to perform classification. Using a leave one subject out training and testing method, we describe the error rates in Figure 5.

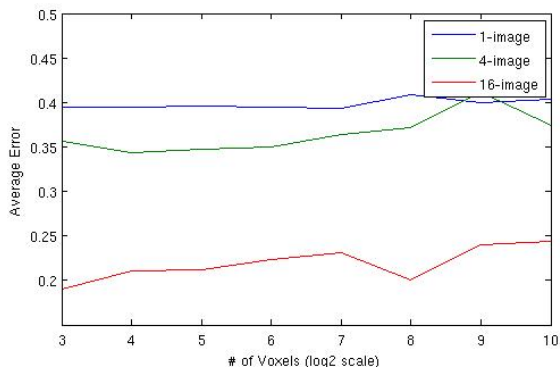


Figure 5: Cross-subject Classification

## 6 Conclusions

The difficulty of our project centered on the “curse of dimensionality” argument coined by Richard Bellman. Our dataset provided a paltry number of examples and subjects and a huge feature space. We initially assumed that this problem would be extremely difficult, however our findings suggest that reasonable classifier accuracy rates can be achieved with a concerted effort to reduce feature space. Machine learning algorithms are applied to a wide variety of

data and spaces, some of which may be extremely complicated. For these algorithms to be robust and multi-purposed, they need to perform reasonably well on these outliers.

Our best single-subject and cross-subject accuracies came from 16-image examples, of which our dataset contained no more than 80 examples per subject. Furthermore, each example contained almost 80,000 features. While our classifiers performed reasonably well on these unadulterated examples, we substantially improved their classification accuracy by focusing them on a smaller set of relevant features. This implies that the important information for our task was hidden within a huge feature space, and heavily suggests the importance of feature extraction and selection methods when dealing with limited amounts of data.

## 7 Applications

Proving that cognitive states are distinguishable with fMRI data is interesting, but if neuroscientists are to use these techniques as tools they must be able to extract useful information about brain activities. Pinpointing the important and distinguishing voxels between cognitive states could unveil meaningful clues to decoding the neural code.

Feature selection and ranking methods based on machine learning classifiers can also be used for a number of different fields, in particular genetic research[3]. Most of the discussion we encountered about learning feature correlations stemmed from the genetics based machine learning field. Additionally, discovering new and better dimensionality reduction methods based on strong classifiers could be fruitful for many applications of machine learning.

## References

- [1] ”Learning to Decode Cognitive States from Brain Images,” T.M. Mitchell, R. Hutchinson, R.S. Niculescu, F.Pereira, X. Wang, M. Just, and S. Newman, *Machine Learning*, Vol. 57, Issue 1-2, pp. 145-175. October 2004.

- [2] Decoste, D., & Scholkopf, B. (2002). Training invariant support vector machines. *Machine Learning*, 46, 161-190.
- [3] Isabelle Guyon , Jason Weston , Stephen Barnhill , Vladimir Vapnik, Gene Selection for Cancer Classification using Support Vector Machines, *Machine Learning*, v.46 n.1-3, p.389-422, 2002
- [4] <http://www-2.cs.cmu.edu/afs/cs.cmu.edu/project/theo-81/www/>