Classification of Functional Magnetic Resonance Imaging Data using Informative Pattern Features

Francisco Pereira and Matthew Botvinick
Princeton Neuroscience Institute and Psychology Department
Princeton University, Princeton NJ 08542
{fpereira,matthewb}@princeton.edu

ABSTRACT

The canonical technique for analyzing functional magnetic resonance imaging (fMRI) data, statistical parametric mapping, produces maps of brain locations that are more active during performance of a task than during a control condition. In recent years, there has been increasing awareness of the fact that there is information in the entire pattern of brain activation and not just in saliently active locations. Classifiers have been the tool of choice for capturing this information and used to make predictions ranging from what kind of object a subject is thinking about to what decision they will make. Such classifiers are usually trained on a selection of voxels from the 3D grid that makes up the activation pattern; often this means the best accuracy is obtained using few voxels, from all across the brain, and that different voxels will be chosen in different cross-validation folds, making the classifiers hard to interpret. The increasing commonality of datasets with tens to hundreds of classes makes this problem even more acute. In this paper we introduce a method for identifying informative subsets of adjacent voxels, corresponding to brain patches that distinguish subsets of classes. These patches can then be used to train classifiers for the distinctions they support and used as "pattern features" for a meta-classifier. We show that this method permits classification at a higher accuracy than that obtained with traditional voxel selection, and that the sets of voxels used are more reproducible across cross-validation folds than those identified with voxel selection, and lie in plausible brain locations.

Categories and Subject Descriptors

I.5 [**Computing Methodologies**]: Pattern Recognition

General Terms

Algorithms, Experimentation

KDD'11, August 21–24, 2011, San Diego, California, USA.

Keywords

functional MRI, classification, feature synthesis, clustering, neuroscience

1. INTRODUCTION

Functional Magnetic Resonance Imaging (fMRI) is a technique used in psychological experiments to measure the blood oxygenation level throughout the brain, which is a proxy for neural activity; this measurement is called *brain activation.* The data resulting from such an experiment is a 3D grid of cells named *voxels* covering the brain (on the order of tens of thousands, usually), measured over time as tasks are performed and thus yielding one time series per voxel (collected every 1-2 seconds and yielding hundreds to thousands of points).

Traditionally, this has been used to contrast brain activation during a task of interest, e.g. reading words, with activation during a related control condition, e.g. reading nonsense words, with the goal of identifying brain locations where the two differ. The most common analysis technique for doing this – statistical parametric mapping [3] – tests each voxel individually by regressing its time series on a predicted time series determined by the task contrast of interest. This fit is scored and thresholded at a given statistical significance level to yield a brain image with clusters of voxels that respond differently to the two tasks (colloquially, these are the images that show parts of the brain that "light up"). Note also that, in tandem with this task-contrasting activation, there are many other processes taking place in the brain that will happen for both tasks : visual processing to read the words, attentional processing due to task demands, etc. The output of this process for a given experiment is a set of 3D coordinates of all the voxel clusters that appear reliably across all the subjects in a study. This result is easy to interpret, since there is a lot of information about what processes each brain area may be involved in. The coordinates are comparable across studies, and thus result reproduciblity is also an expectation.

In recent years, there has been increasing awareness of the fact that there is information in the entire pattern of brain activation and not just in saliently active locations. Classifiers have been the tool of choice for capturing this information and used to make predictions ranging from what stimulus a subject is seeing, what kind of object they are thinking about or what decision they will make [11] [13] [7]. Such classifiers are usually trained on a selection of voxels from the 3D grid that makes up the activation pattern; often this means the best accuracy is obtained using few voxels,

from all across the brain, and that different voxels will be chosen in different cross-validation folds. Domain experts will generally raise two types of concern in face of this. The first is that the classifiers can be hard to interpret, e.g. does weight placed in a voxel in one area mean that that area is used? The second is the lack of reproducibility in face of the choice of different voxels in different folds. Even though there might be redundant information which would explain this, the results can still be perceived as dubious.

One approach to this problem is to try and regularize classifiers so that they include as many informative voxels as possible [1], thus identifying localizable clusters of voxels that may overlap across folds. A different approach is to cross-validate classifiers over small sections of the grid covering the brain, known as *searchlights* [10] [14]. This can be used to produce a map of the accuracy in the searchlight around each voxel, taking advantage of the pattern of activation across all the voxels contained in it. Such a map can then be thresholded to leave only locations where accuracy is significantly above chance.

A different issue that affects both of these approaches stems from the increasing commonality of datasets with tens to hundreds of tasks or stimuli. Knowing the location of a voxel does not suffice to interpret what it is doing, as it could be very different from stimulus to stimulus (rather than just active or not, as in the two task situation). It's also possible for neighbouring voxels to be doing different things relatex to the same task, thus barring any attempt to cluster them by their time series. Few brain locations will differentiate between all possible stimuli, say, at the spatial resolution of fMRI, and hence defining a searchlight size or shape is a trade-off between including voxels and making it harder to learn a classifier and excluding voxels and thus the number of distinctions that can be made.

Our goal is to attempt to address all of these issues by building a classifier that works in terms of *the presence or absence of patterns of activation across certain regions*, rather than the level of activation in individual voxels. The procedure we will describe involves building searchlights of various sizes and shapes in a data-driven manner, and learning pattern detectors operating over those searchlights. The outputs of these can then be used as the inputs to the classifier.

2. DATA AND METHODS

2.1 Data

As we described earlier, the grid covering the brain contains on the order of tens of thousands voxels, measured over time as tasks are performed, every 1-2 seconds, yielding hundreds to thousands of 3D images per experiment. During an experiment a given task is performed a certain number of times – trials – and often the images collected during one trial are collapsed or averaged together, giving us one 3D image that can be clearly labelled with what happened in that trial, e.g. what stimulus was being seen or what decision a subject made. Although the grid covers the entire head, only a fraction of its voxels contain cortex in a typical subject; hence we only consider these voxels as features.

A *searchlight* is a small section of the 3D grid [10], in our case a $27 = 3 \times 3 \times 3$ voxel cube. Analyses using searchlights generally entail computing a statistic [10] or cross-validating a classifier over the dataset containing just those voxels [14], and do so for the searchlights centered around each voxel in the brain. The intuition for this is that individual voxels are very noisy features, and an effect observed across a group of voxels is more trustworthy.

In the experiment performed to obtain our dataset [1] [12], subjects observed a word and a line drawing of an item, displayed on a screen for 3 seconds and followed by 8 seconds of a blank screen. The items named/depicted belonged to one of 12 categories: animals, body parts, buildings, building parts, clothing, furniture, insects, kitchen, man-made objects, tools, vegetables and vehicles. The task was to think about the item and its properties while it was displayed. There were 5 different exemplars of each of the 12 categories and 6 experimental epochs. In each epoch all 60 exemplars were shown in random order without repetition, and all epochs had the same exemplars. During an experiment the task repeated a total of 360 times, and a 3D image of the fMRI-measured brain activation acquired every second. Each example for classification purposes was the average image during a 4 second span while the subject was thinking about the item shown a few seconds earlier (a period which contains the peak of the signal during the trial); *the dataset thus contains 360 examples*, as many as trial tasks. The voxel size was $3 \times 3 \times 5$ mm, with the number of voxels being between 20000 and 21000 depending on which of the 9 subjects is considered.

2.2 Algorithm

This section describes the entire procedure that takes place given a training set and test set, which would appear in the context of cross-validation. In our experiments this is performed by using even or odd epochs as the two folds, with 180 examples in each. We do this – rather than a leave-one-epoch-out, for instance – because we are interested in estimating the reproducibility of models learned on two datasets from the same subject, and using training sets with overlapping examples would give an inflated estimate of this. The procedure has two parts, described separately for clarity.

2.2.1 *Identifying informative voxel sets*

The first goal is to identify different kinds of information present throughout the brain. Ultimately, we would like to understand how a certain semantic category is represented throughout the brain (e.g. do "Insects" and "Animals" share part of their representation because both kinds of things are alive?). Intuitively, there is information in a given location if at least two categories can be distinguished looking at their respective patterns of activation there (or otherwise the pattern of activation is noise or common to all categories). A natural starting point is to consider whether each pair of categories can be distinguished in each of the thousands of searchlights covering the brain, and to do this for all pairs.

The second goal is to assemble as large a region as possible containing one kind of information. Examining each small searchlight makes sense if we consider that, a priori, we don't know where the information is or how big a pattern of activation would have to be considered (with some exceptions, notably areas that respond to faces, houses or body parts see [8] for a review). That said, if the same categories are distinguishable in two adjacent searchlights (which overlap) then it is reasonable to assume that all their voxels put together would still be able to make the same distinctions.

[1]The data were kindly shared with us by Tom Mitchell and Marcel Just, from Carnegie Mellon University.

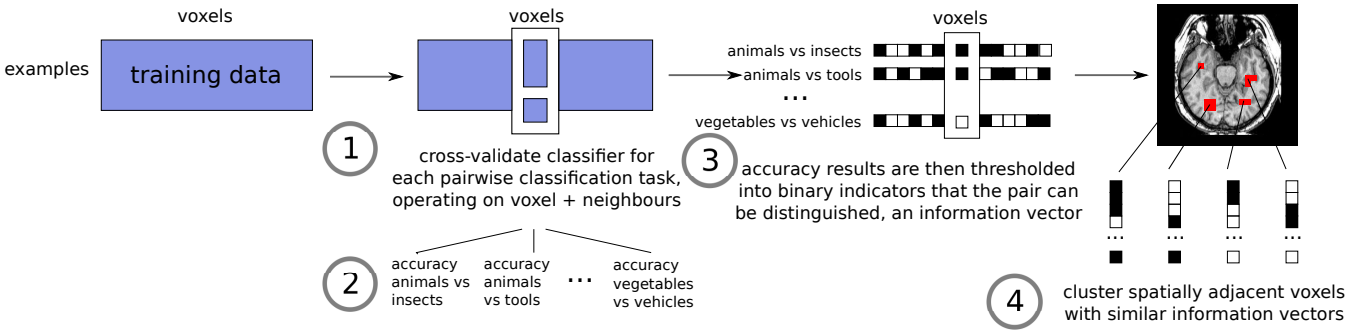

Figure 1: Identifying informative voxel sets

Doing this repeatedly allows us to find a kind of natural searchlight, not bound by shape or size assumptions, which we call an *informative voxel set*.

Figure 1 depicts the steps in the procedure to identify informative voxel sets. Within a training set, and for each pairwise classification task (e.g. animals-vs-insects, tools-vs-buildings, etc):

1. Cross-validate a classifier in the searchlight centered around each voxel and obtain an accuracy value.

2. Assign that value to the voxel at the center of the searchlight, yielding an accuracy brain image.

3. Transform the accuracy brain image into a p-value brain image (of obtaining accuracy as high or higher under the null hypothesis that the classes are not distinguishable, see Section 2.2.4 for details). Threshold this at 0.01 (uncorrected) to get a binary brain image with candidate locations for whether this class distinction can be made.

After these steps we have as many binary images as there are pairs of classes (66 pairs total). These can also be seen as a new dataset, with $\#pairs$ examples and $\#voxels$ features, where each voxel has a $\#pairs$ *information vector* summarizing what distinctions can be made inside its searchlight. We can now use agglomerative clustering of adjacent voxels with similar information vectors to build informative voxel sets (step 4 in Figure 1, see Section 2.2.4 for details of the clustering algorithm). The final result is an assignment of each voxel to a cluster, and an information vectors for each cluster. Note that there are typically only a few tens of clusters with 2 or more voxels.

We do not correct for multiple comparisons at the binary brain image stage because we want to retain as many reasonable, low p-value candidate voxels as possible. In our experience noise voxels that have low p-values don't have too many neighbours with low p-values as well, so they should be eliminated during the clustering stage.

2.2.2 Constructing pattern features for classification

The goal of the second part of the procedure is use the informative voxel sets to create features that correspond to indicators of a certain pattern being present in each set, and convert train and test examples into this new feature space. Given the informative sets identified in the training data in the first part, Figure 2 shows the steps taken:

1. For each informative set, train as many classifiers as there are class distinctions in the corresponding information vector (the classifier is trained on all the voxels in the informative set). Any classifier will do, as long as it can be trained to output the probability of one of the two classes given an example.

2. For each informative set, apply the classifiers learned to the voxels in the set across *all* examples of the training data. The output of a classifier over all the examples becomes a new *pattern feature*, taking values in the $[0, 1]$ range; extreme values mean that the classifier detects the pattern corresponding to one of the two classes. Each set gives rise to as many pattern features are there distinctions in its information vector. Note that each classifier will be applied to examples of classes other than the two used to train it; this is deliberate, as the classifier is being used solely as pattern detector.

3. The same feature construction process is done for the test set, applying the classifiers learned on the training set.

2.2.3 Learn a classifier over pattern features

After the construction of training and test pattern feature datasets, a classifier can be trained and tested, as shown in step 4 in Figure 2. There is no need to use the same kind of classifier as that used for pattern feature construction; in our case, we need only a classifier that is interpretable in the sense that we can ascertain the sensitivity of its predictions to each of the features.

2.2.4 Further Details

Classifiers on searchlights.

The classifier cross-validated in each searchlight, for each pair of classes, was LDA (Linear Discriminant Analys, [5]) with a shrinkage estimator of the covariance matrix that sets the shrinkage parameter automatically [16]. We used it because it could capture the covariance structure of the voxels inside a searchlight, given there were at most 27 voxels in it. The cross-validation was done inside the training set (3 epochs, containing 180 examples belonging to 12 classes), leaving one epoch out. There were thus 10 examples of each class for each pairwise discrimination. The resulting accuracy for each pairwise discrimination was converted into a p-value, by computing the probability of classifying that many

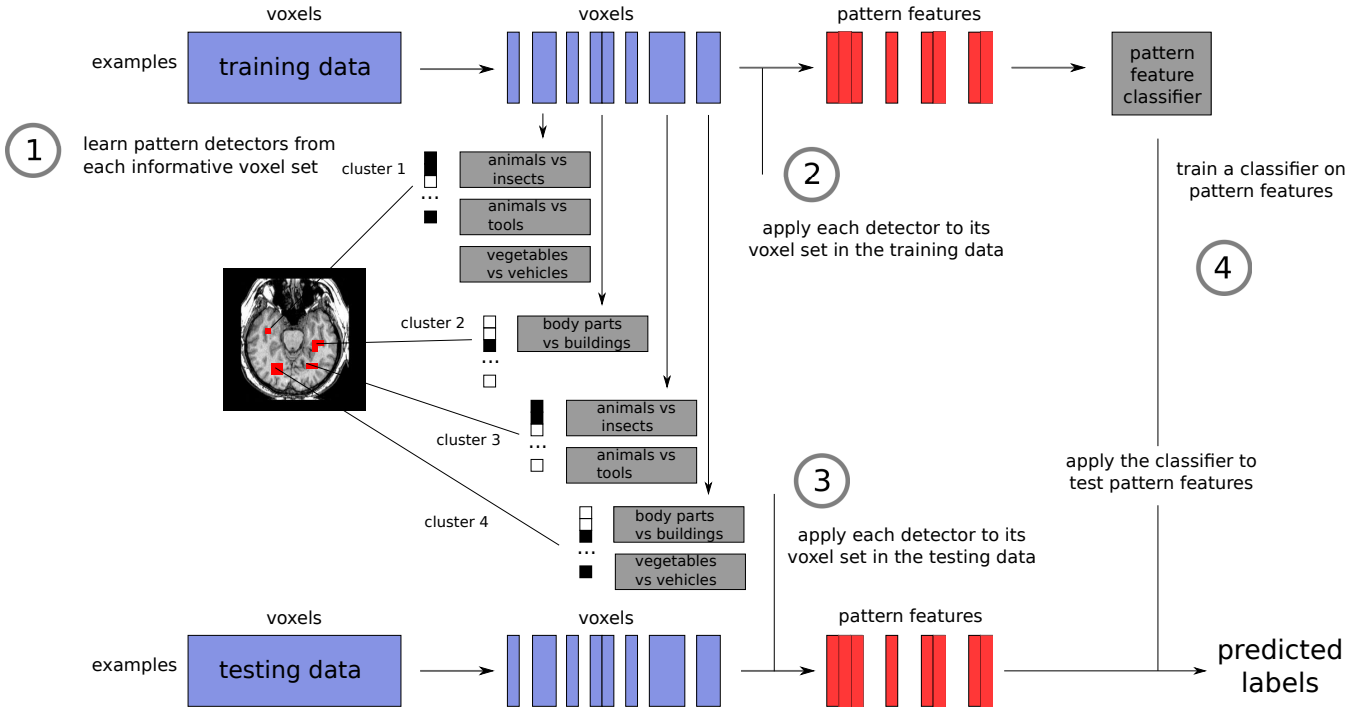

Figure 2: Constructing pattern features

or more examples correctly if the classifier were performing at chance level. Under the null hypothesis the number of correctly labelled examples has a binomial distribution with $p = 0.5$ and $n = \#examples$. For more details see [15].

Clustering of information vectors.

At the start of the agglomerative clustering process, each voxel is a cluster by itself and has an associated binary information vector with 66 entries corresponding to which pairs of classes can be distinguished in its surrounding searchlight. For each voxel we compute the similarity of its information vector with those of all its neighbours, which takes a few tens of seconds.

The similarity measure between two vectors $\mathbf{v}_i$ and $\mathbf{v}_j$ is obtained by computing the number of 1-entries present in both vectors, $\sum_{pairs} \texttt{AND}(\mathbf{v}_i, \mathbf{v}_j)$, the number of 1-entries present in only one of them, $\sum_{pairs} \texttt{XOR}(\mathbf{v}_i, \mathbf{v}_j)$ and then the measure

$$\text{similarity}(\mathbf{v}_i, \mathbf{v}_j) = \frac{\sum_{pairs} \texttt{AND}(\mathbf{v}_i, \mathbf{v}_j) - \frac{\sum_{pairs} \texttt{XOR}(\mathbf{v}_i, \mathbf{v}_j)}{2}}{\sum_{pairs} \texttt{AND}(\mathbf{v}_i, \mathbf{v}_j)}$$

The measure was chosen with several properties in mind. The first is that it peaks at 1 if the two vectors match exactly, and decreases – possibly into negative values – if there are mismatches; it will tolerate more mismatches if there are more distinctions being made. The second is that it will cluster sparse vectors as readily as dense vectors, as long as there are vew few mismatches. The number of entries present in only one is divided by 2 so that the differences do not get twice the weight of the similarities.

We then iterate the following steps:

1. find the two adjacent clusters (voxels, in the first iteration) whose information vectors are the most similar
2. merge the clusters, obtaining a new information vector that will represent the cluster (this is obtained by computing a soft-AND function of the initial information vectors of all voxels in both clusters, which makes an entry 1 if 90% of voxels have it)
3. update the similarity between that cluster and all its neigbours (a very small fraction of the number of similarities computed during initialization)

stopping when similarity drops below very conservative threshold for the measure (0.9, ensures that only voxels with almost identical profiles get merged). The soft-AND appears to produce reasonable results for the amount of noise present, in that it is robust to small differences (e.g. if a single voxel in a large cluster has 0 entry in which all others have a 1, then the cluster prototype should still have a 1 there, and vice versa). This runs in a few minutes for all our subjects.

Generation of pattern features from informative sets.

After clustering stops there are typically several thousand clusters left, given the high similarity threshold we use, hence we have to select which ones will give rise to pattern features. For each cluster, we know the number of voxels in it and also its information vector, which is an indication of how many class pairs can be distinguished there. Regardless of size, we use clusters that support at least 6 distinctions ($\frac{1}{2}$ of the number of classes), as we are interested in clusters that encode semantic representations and thus are likely to have patterns shared across several classes.

This is the main parameter of the algorithm. It could be set to a value we know we want a priori or be chosen

by cross-validation inside the training set if accuracy is the main goal. We report the results for various settings to examine the sensitivity of results – both qualitatively and quantitatively – to this setting.

For each cluster selected we then train as many classifiers as there are distinctions in its information vector, and use their output when applied to train and test data to produce pattern features, as described earlier. We use a linear SVM (LIBSVM [2], $\lambda = 1$) for this purpose.

Classifier on pattern features.

The final step in the procedure is to train a classifier on the pattern features produced from the training set. This classifier can be chosen so as to produce a model relating pattern features to class labels in a manner is intuitive to a domain expert. One example might be a decision tree where the first decision would depend on the presence of a certain pattern in a particular voxel cluster. Presence would indicate the subject was thinking of a living thing, whereas a different pattern would be present for inanimate objects. Expanding further would essentially provide a representation for examples of a particular category composed of the presence and absence of a combination of patterns. For our experiments we use a linear SVM (LIBSVM [2], λ=1), as our primary interest is determining how the method performs as gauged by a number of quantitative measures.

3. EXPERIMENTS

3.1 Baseline

We will contrast experimental results obtained with our method with a baseline of classification using voxel selection. The scoring criterion for each voxel is the accuracy of a LDA searchlight classifier doing 12 way classification. The number of voxels to use was selected by nested cross-validation inside the training set [2]. The results are shown in the first line of Table 1. Whereas the accuracy is above chance (0.08) for all subjects, it is rather low for some. There are at least two factors responsible for this. The first is that some classes give rise to very similar patterns of activation (e.g. "buildings" and "building parts"), and hence examples in these classes are confusable (confusion matrices bear this out). The second factor is that subjects vary in their ability to stay focused on the task and avoid stray thoughts or remembering other parts of the experiment, hence examples may not belong to the class corresponding to the label or even any class at all. [12] also points out that accuracy is correlated with a subject's ability to stay still during the experiment.

3.2 Results

Table 1 shows the classification accuracy results obtained with our method, varying the threshold used for determining which clusters to draw pattern features from. Highlighted in bold is the threshold we would like to use, since we would like to have clusters that distinguish at least a few classes from others, but do not wish to overly restrict the choice to clusters that distinguish several. The results indicate that the threshold choice might not make much difference for any but the subjects where classification performance was worse (P5 and P9, where voxel selection did not improve results relative to using the whole brain).

The results also show that there is no loss of information in switching from using voxels to using pattern detectors as features. Depending on the subject and set of results, there are generally thousands of voxel clusters being used, *each of which gives rise to as many pattern detector features as there are class distinctions supported by that cluster*. Hence we typically have many thousands of features and yet the accuracy is comparable or better than that obtained with voxel selection; one possible explanation for this is the existence of many correlated detector features.

Note that classification happens in a 2-fold cross-validation procedure, and that that the total number of examples is 360. This is the reason why we opted not to include error bars on classification results or perform paired t-tests to compare pairs of them. Instead, we performed sign tests that look at whether our method outperformed the baseline voxel selection method for each subject, yielding a p-value < 0.01 at the various thresholds.

The vast majority of the clusters are just single voxels and their searchlights. Given this, it's legitimate to ask to what extent our method suffers from the lack of overlap in voxels used in the two halves of the dataset that tends to affect voxel selection. Table 2 shows the overlap, computed as $\frac{\#\text{voxels selected in both folds}}{\#\text{voxels selected in either fold}}$. In the baseline case the overlap is between selected voxel subsets, for our method it's between the voxels that belong to sets used to produce patern detectors. As expected, overlap decreases as we raise the threshold for using clusters to produce pattern detectors. The ranges of numbers of clusters used are given in the last column of the table, and the lower and upper limits correspond to the worst and best subject in terms of accuracy. Regardless of threshold choice, our procedure for identifying informative voxel sets using pairwise classification profiles seems to also provide more reliable voxel selection (considering the number of single searchlight clusters). The results suggest that, for this dataset, most searchlights support at least 9-12 category distinctions.

Finally, we can examine the locations of the clusters identified, as the dataset has been annotated with anatomical labels for each voxel [17]. For the various subjects, the overwhelming majority of clusters lie in various locations inside either occipital cortex (the various visual cortices, where visual representations are assembled, see for instance [9]) or temporal cortex (e.g. the various temporal gyri, the fusiform gyrus, the lingual gyrus, and other locations connected with representations of semantic information [6], [4], [12]).

4. CONCLUSIONS AND FURTHER WORK

We have introduced a method to synthesize features corresponding to the presence/absence of certain patterns of activation across small brain regions, identified in a data-driven way. Such features appear to preserve or enhance the information contained in the levels of activation in individual voxels. We conclude this from results showing that they permit classification at a higher accuracy than that obtained with a classifier working over voxels selected using a competitive method. Moreover, the sets of voxels used are more reproducible across cross-validation folds than those identified with voxel selection, and lie in plausible brain locations.

[2]Possible choices were 50, 100, 200, 400, 800, 1200, 1600, 2000, 4000, 8000, 16000 or all voxels.

We are continuing to develop this method, with the next stage being identifying correlated pattern feature detectors and use that to further condense small clusters into larger ones, if possible, or non-contiguous units. The ultimate goal is to be able to robustly identify the various types of information present in a pattern of brain activation, as defined by the ability to distinguish some subset of the classes from another subset. Given this, it will be feasible to try and related such distinctions to existing theories about how semantic information is represented across the brain. A second reason for doing this is to reduce the number of clusters considered in the final classifier. The desired outcome is one where the pattern of brain activation for a given class can be succintly described in terms of cluster subpatterns being present or absent, and classes related by which subpatterns they share.

5. REFERENCES

[1] M. K. Carroll, G. a. Cecchi, I. Rish, R. Garg, and a. R. Rao. Prediction and interpretation of distributed neural activity with sparse models. *NeuroImage*, 44(1):112–22, Jan. 2009.

[2] C. Chang and C. Lin. LIBSVM: a library for support vector machines. Technical report, 2001.

[3] K. J. Friston, J. Ashburner, S. J. Kiebel, T. E. Nichols, and W. D. Penny. *Statistical Parametric Mapping: The Analysis of Functional Brain Images*. Academic Press, 2006.

[4] S. J. Hanson, T. Matsuka, and J. V. Haxby. Combinatorial codes in ventral temporal lobe for object recognition: Haxby (2001) revisited: is there a "face" area? *NeuroImage*, 23(1):156–66, 2004.

[5] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: data mining, inference and prediction*. Springer-Verlag, 2001.

[6] J. Haxby, M. Gobbini, M. Furey, A. Ishai, J. Schouten, and P. Pietrini. Distributed and overlapping representations of faces and objects in ventral temporal cortex. *Science*, 293(5539):2425, 2001.

[7] J. Haynes and G. Rees. Decoding mental states from brain activity in humans. *Nature Reviews Neuroscience*, 7(7):523–34, 2006.

[8] M. A. Just, V. L. Cherkassky, S. Aryal, and T. M. Mitchell. A neurosemantic theory of concrete noun representation based on the underlying brain codes. *PloS one*, 5(1):e8622, 2010.

[9] K. N. Kay, T. Naselaris, R. J. Prenger, and J. L. Gallant. Identifying natural images from human brain activity. *Nature*, 452(7185):352–5, 2008.

[10] N. Kriegeskorte, R. Goebel, and P. Bandettini. Information-based functional brain mapping. *Proceedings of the National Academy of Sciences*, 103(10):3863, 2006.

[11] T. M. Mitchell, R. Hutchinson, R. S. Niculescu, F. Pereira, X. Wang, M. Just, and S. Newman. Learning to Decode Cognitive States from Brain Images. *Machine Learning*, 57(1/2):145–175, Oct. 2004.

[12] T. M. Mitchell, S. V. Shinkareva, A. Carlson, K. Chang, V. L. Malave, R. A. Mason, and M. A. Just. Predicting human brain activity associated with the meanings of nouns. *Science (New York, N.Y.)*, 320(5880):1191–5, 2008.

[13] K. A. Norman, S. M. Polyn, G. J. Detre, and J. V. Haxby. Beyond mind-reading: multi-voxel pattern analysis of fMRI data. *Trends in cognitive sciences*, 10(9):424–30, 2006.

[14] F. Pereira and M. Botvinick. Information mapping with pattern classifiers: a comparative study. *NeuroImage (to appear)*, 2010.

[15] F. Pereira, T. M. Mitchell, and M. Botvinick. Machine learning classifiers and fMRI: a tutorial overview. *NeuroImage*, 45(1 Suppl):S199–209, Mar. 2009.

[16] J. Schäfer and K. Strimmer. A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statistical applications in genetics and molecular biology*, 4:Article32, Jan. 2005.

[17] N. Tzourio-Mazoyer, B. Landeau, D. Papathanassiou, F. Crivello, O. Etard, N. Delcroix, B. Mazoyer, and M. Joliot. Automated anatomical labeling of activations in SPM using a macroscopic anatomical parcellation of the MNI MRI single-subject brain. *NeuroImage*, 15(1):273–89, 2002.

Table 1: Classification accuracy for the 9 subjects. The p-value of the sign-test for the number of subjects where our method outperforms voxel selection is < 0.01 for the various thresholds.

	P1	P2	P3	P4	P5	P6	P7	P8	P9
baseline (using all voxels)	0.31	0.21	0.19	0.27	0.13	0.09	0.14	0.13	0.15
baseline (voxel selection)	0.53	0.33	0.24	0.34	0.14	0.16	0.21	0.20	0.15
#voxels selected (fold 1)	1200	400	200	1600	800	800	800	400	2000
#voxels selected (fold 2)	800	200	100	800	50	8000	100	1200	100
inclusion threshold 3	0.57	0.34	0.33	0.42	0.16	0.21	0.24	0.16	0.18
inclusion threshold 6	**0.57**	**0.34**	**0.33**	**0.42**	**0.17**	**0.28**	**0.23**	**0.20**	**0.17**
inclusion threshold 9	0.58	0.34	0.33	0.43	0.16	0.31	0.22	0.20	0.19
inclusion threshold 12	0.56	0.36	0.35	0.41	0.17	0.30	0.21	0.16	0.14

Table 2: Cross-fold overlap in voxels used for the 9 subjects.

	P1	P2	P3	P4	P5	P6	P7	P8	P9	#clusters used
baseline (voxel selection)	0.24	0.12	0.05	0.08	0.01	0.06	0.02	0.04	0.01	-
inclusion threshold 3	0.29	0.22	0.24	0.35	0.44	0.46	0.45	0.43	0.43	approx. 3000-8500
inclusion threshold 6	**0.29**	**0.22**	**0.24**	**0.35**	**0.17**	**0.20**	**0.19**	**0.17**	**0.16**	approx. 1000-4500
inclusion threshold 9	0.26	0.15	0.19	0.23	0.07	0.13	0.11	0.09	0.06	approx. 250-2500
inclusion threshold 12	0.30	0.15	0.19	0.21	0.05	0.13	0.12	0.05	0.03	approx. 50-1500