Applying feature selection methods on fMRI data

Raoul Harel Selman Ercan Elgar de Groot Stijn van Schooten

March 29, 2014

Abstract

In neuroscience, the ability to correlate and classify certain activity patterns of the brain to different physical and mental states of the subject is of high importance. Analysis of fMRI data is one of the venues in which this objective is being pursued. However data produced using fMRI technology is highly complex. To this end, machine learning becomes relevant. Prominent hurdles facing fMRI data analysis are their high-dimensionality (thousands of features per instance), low signal-to-noise ratio and interdependency. This motivates the use of feature selection methods in order to consolidate relevant information and discard noise. Many feature selection methods exist but only a few have been applied in the fMRI domain. In this paper we identify positive characteristics of feature selection algorithms that are beneficial when dealing with fMRI datasets. To do this, we evaluate representatives from each of the three main feature selection classes: Filters, wrappers and embedded methods. We have found probabilistic embedded methods to be the most suitable for fMRI data. We would therefore recommend using these (or similar) methods to process data with fMRI-like characteristics.

1 | Introduction

Functional Magnetic Resonance Imaging (fMRI) is an important technique used for brain mapping. It maps neural activity by measuring the amount of blood-flow in the brain. It has been subject to a lot of scientific research, and has a lot of (potential) uses. For example, a recent study has shown fMRI analysis can assist in the diagnosis of autism spectrum disorder [11]. Another study has discovered certain regions in the brain that show altered activity in early stages of Alzheimer's disease [42][39]. This might help in the early discovery of this disease, with resulting benefits. Another study has proved that it is possible to detect drug abuse in subjects using fMRI technology [45]. There has even been research into the possibility of lie detection [7] and diagnosis of other related psychological diseases [22]. Apart from clinical use, fMRI research has scientific value as well. By finding certain regions in the brain that activate when a subject is doing a specific task, we can learn a lot about the brain. For example, a study by Hanson *et al.* [18] shows that there is no single area in the brain that is responsible for face recognition. This kind of research may change the way we think about how the brain works.

A lot of this research is about classification problems: distinguishing between two states of the brain. In other words; *classifying* brain states. Learning a computer to do this (machine learning) is essential. This subject of study lies way beyond the scopes of just fMRI research, and even in bioinformatics it has a multitude of applications: gene expression, protein function, biomarker analysis and many more. It may not be surprising that a lot of research has gone into the optimization of machine learning techniques.

One aspect of machine learning that is receiving more and more attention is feature selection. Feature selection is the art of selecting those aspects of the data that distinguish different classes best. In fMRI in particular, this often means selecting certain brain regions. This reduces the amount of data the classifier has to consider, but can also result in insights about the data, like in the face recognition example above.

In this paper we aim to give insight in the field of feature selection on fMRI data. In the sea of available feature selection methods we try to find the ones that show potential, and of those

methods we try to select the ones that are the most useful for fMRI data analysis.
In the next chapter we go more in depth on fMRI data, what it is and what the properties are. In third chapter we will talk more about feature selection, give a general overview of what it is and further explain why it is useful when analysing fMRI data and in the fourth chapter we lay down a multitude of feature selection methods, explain how they work and discuss their usefulness in fMRI research.

2 | On fMRI

In this section we will introduce MRI, fMRI and its accompanying techniques as well as explaining how the measurements are made and what can be done with its data. Firstly, fMRI will be elaborated. Next, we will explain some aspects of the measuring environment. Afterwards, the properties of fMRI data will be listed in order to give a better insight into why certain feature selection methods might work better for this type of data. The purpose of this section is to introduce the reader into the world of neuroscience, and to fMRI data in particular.

2.1 What is fMRI data?

Magnetic Resonance Imaging (MRI) is a technique used to measure hydrogen atom density differences in a body by rotating and pulsating a large and powerful magnet around it. By doing this at the right frequency, the resulting energy will push the hydrogen atoms in an excited state and they will start to emit a radio frequency signal. The contrasts in de MRI image is determined by the rate at which the excited atoms return to their "relaxed" stated. [28].
These measurements actually are two-dimensional slices, and can be interpolated into a complete 3D image, which can later be evaluated using contrasting methods to produce typical MRI images. Since differences in hydrogen concentrations are depicted, the boundaries between different types of tissue can be shown (under assumption that tissue is uniform). Certain properties can be read from these hydrogen densities, for instance whether a swelling is a tumor or a cyst [28].
fMRI is a relatively new procedure to measure brain activity by detecting changes in blood flow utilizing MRI. More blood flow is associated with a higher firing rate of the neurons in that part of the brain, thus measuring blood flow amounts to measuring activity of that specific part of the brain. This is made possible by the fact that the brain does not store energy in any form, so when more is required (when more neurons are firing or when they fire in a higher frequency), the blood flow to that area must increase.
fMRI uses a technique known as Blood-Oxygen-Level Dependent (BOLD) contrast [29]. The resulting activity map is usually represented graphically by colour coding the rate of activity, resulting in an image similar to the one in figure 2.1. This method can achieve a very high resolution of less than a millimetre (for smaller apparati, not fit for human measurement), but the measurement window is usually less than a second long. By measuring multiple times over several minutes, data is given a temporal dimension in addition to its spatial and intensity dimensions.
fMRI measurements are used for medical and research purposes, since they offers a much higher resolution than EEG (Electroencephalography, measuring the electrical activity along the scalp) and MEG (Magnetoencephalography, measuring the magnetic fields along the scalp produced by the firing of neurons, thus creating small electrical currents), making it possible to model brain activity much better within a limited window of time. Most research is done in the behavioural field, but tests are also done in relation to biometrics or automated prosthesis.

2.2 How is fMRI data obtained?

Since fMRI data represents the activity level in specific parts of the brain, certain activity models can be linked to certain activities of the subject. Usually, the measurements are done while the subject is undergoing a specified test. This test can be a physical test, like moving an arm, more psychological, like listening to certain sounds, or a combination, like solving a puzzle. A

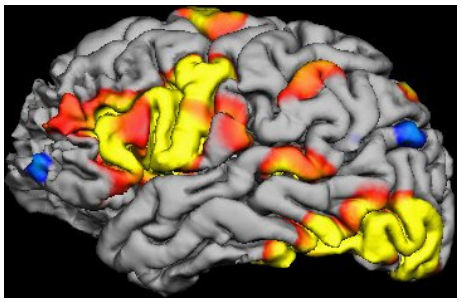

Figure 1: A typical fMRI measurement, mapped with BOLD contrast. The changes in blood flow are color-coded in order to easily identify which parts of the brain are more active. Red indicates an increase and blue a decrease in blood flow. (image by Ned T. Sahin)

measurement consists of a lot of images that are pooled together, each image made within a short timespan of the previous one. These images give a spatial and temporal image of the brain activity, usually with intervals of less than a second. For standard intensity mapping, the first measurements (subject is in rest and tries to minimize brain activity) are used as a baseline. Afterwards the subject performs the test routine and while or after this, another measurement is done. By looking at the differences between the baseline images and the new image, a difference in blood flow can be mapped. But for a relative difference between tasks, images can be compared to each other.
After making the measurements, the data is collected and preprocessed for further use. This means removing spatial shifts (the subject can move) and normalising between all images. Afterwards, the images (now still slices) are aggregated and interpolated into a 3D image and contrasted.
Because a physical task activates other parts of the brain than psychological tasks, the task will be designed to try to create a large as possible difference between neural images. By chaining tasks, more interesting data can be collected, like what part of the brain is activated for the left arm versus the right arm. By putting subtle differences in the tasks, the areas activated for more specific tasks can be identified, but deteriorate the accuracy of the produced model.
Also a lot of the acquired data is cross-referenced with existing data, since complete fMRI datasets are relatively sparse. One example of a widely used dataset is the Haxby dataset. This dataset was created in 2001 by Haxby *et al.* and is freely accessible [19]. A processed sample of this dataset can be seen in figure 2.2.

2.3 Properties of fMRI data

When measuring, the fMRI data comes out of the machine as slices, which are stacked to go from the 2D slices to a 3D representation of the brain activity. The 3D image consists of voxels (volumetric pixels: 3D pixels): blocks of 2mm x 2mm x 2mm [9] covering millions of neurons. Since fMRI data consists of measurements from many different locations in the brain, this leads to large (a few GB) datasets and considerable "noise" (noise being the firing of neurons that is not related to the test). The great number of voxels means that the feature-to-instance ratio is usually very large (around 5000:1, with every 3D image being an instance and every voxel a feature[24]) and that a lot of points are measured within one measurement.
While having a relatively high resolution (usually more than 100.000 voxels per image), one voxel still represents millions of neurons, which means that (while making a mapping of the brain activity) only an approximate model based on clustered average brain activities can be made.
Additionally, within fMRI data, voxels that are less informative by themselves can be useful when considered together. This means that when analysing this data, methods that can capture this interdependency are preferable to methods that would consider each voxel in isolation.

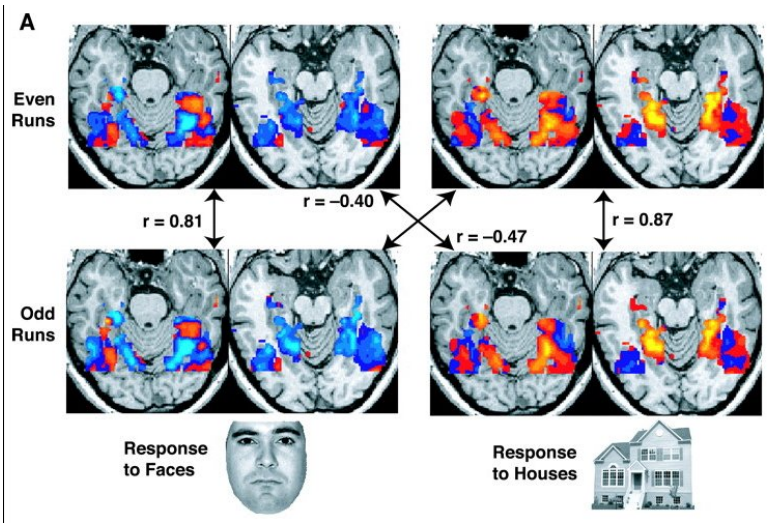

Figure 2: A processed sample of a part of the Haxby dataset. The subjects were shown faces and houses, and the parts that are used are coloured. The 'r' factors are correlations, the higher the absolute value, the better the match. (image by NIMH)

3 | Feature selection

This section introduces the concept of feature selection and motivates its use in the context of fMRI. First the underlying idea is explained and finally the principal motives supporting the incorporation of feature selection in fMRI data analysis will be mentioned.

3.1 Introduction to feature selection

Feature selection is the process of selecting a subset of features to be used in the construction of a mathematical model. However in this paper to limit ourselves to just classifiers as the model. Feature selection is often mentioned in the context of machine-learning, where it is essential to rank properties of (abstract) objects (features) on a scale of "good" to "bad" . Good features are those that help interpret the results of a classification process. Bad features are those that do not contribute to the results' interpretability. Feature selection methods aim to measure and rank the usefulness of features.

The use of feature selection methods is motivated by both theoretical as well as practical reasons. Theoretically, incorporation of feature selection is desired when it is thought that elimination of certain features off the dataset will improve a classifier's accuracy. Greater accuracy is desired as it means the classifier is more reliable when used on new data, which ultimately is the end-goal of any machine-learning process. Practically, the use of feature selection is sometimes necessary due to a lack of processing power which inhibits a classifier's ability to include all features in its formulation [16].

Feature selection is in fact a member of the more broad feature extraction class of methods [17]. However it should not be confused with other feature extraction methods. For example, while Principal Component Analysis (one of the feature extraction methods, abbreviated PCA) reduces the feature-set's size, it does so by merging features, not by omitting them [20]. PCA therefore may include irrelevant or even harmful features (such as ones with an extremely high variance) in its output, undermining the classifier's accuracy. Another drawback of merging features is that the results can no longer be directly interpreted using the original instance, since the initial features no longer exist in the output. Feature selection methods are less prone (but not immune) to these pitfalls.

Many classes and variants of feature selection methods exist. They can be divided into three main groups: Filters, wrappers and embedded methods [16]. Each of these uses a different way of assessing a feature's merit. Furthermore, each method can fall into one of two mutually exclusive classes: Univariate and multivariate methods. A diagram depicting the inter-class relationships in more detail can be seen in figure 3.1. A description of all these classes follows.

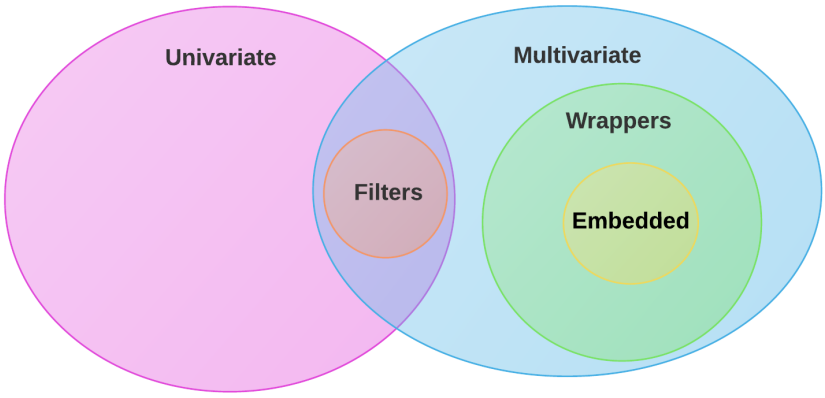

Figure 3: Venn diagram depicting the relationship between the univariate and multivariate classes, filters, wrappers and embedded methods.

Univariate and Multivariate Methods Univariate methods use per-feature computed statistics to determine an optimal feature subset. In contrast to that, multivariate methods are allowed to use statistics that combine a multitude of features to reach an optimal subset [20].

Wrappers Wrappers use the classifier itself to gauge a certain subset's merit. The classifier is then treated as a black-box if you will. Subsets of features from the training set are run through it and the classifier's accuracy is measured. Since the amount of available subsets to assess grows exponentially with the amount of features, an exhaustive search of the optimal subset is beyond a computer's processing power and is thus impractical. As a solution to this problem, these methods employ customizable heuristics (search algorithms) and termination-conditions that traverse the search-space in such a way that attempts to estimate the optimal subset as efficiently as possible. The search-process starts at some predefined initial state and subset. Then, the classifier's accuracy with the given subset is tested against a training set. The results of the test are used in order to construct the next candidate subset. This process repeats itself until certain termination-conditions are met [21].

Filters Filters, in contrast to wrappers, are classifier-independent and rely on the internal relations of the data itself to estimate the usefulness of features and is mostly suitable as a preprocessing step prior to the application of further analysis. An advantage of these methods compared to wrappers is their efficiency, as they often employ relatively simple operations compared to wrappers [16]. A drawback is the fact that the resulting feature subset is not made to suit any specific classifier.

Embedded Methods Embedded methods are similar to wrappers in the sense that they incorporate the classifier in the decision process, but differ in the fact that they are not universal

- each is custom-made for a specific classifier [25]. Embedded methods are therefore much faster than wrappers but pay the price in the form of their non-flexible architecture.

3.2 Why apply feature selection on fMRI data?

While feature selection is important in machine learning in general, a lot of the properties of fMRI data make it especially important in this context. An fMRI scan of a brain contains hundreds of thousands of voxels, and when the resolution of fMRI scanners improves, this amount will only go up. Relevant regions of the brain are often only a small portion of all these voxels. While we could use all the voxels in a machine learning algorithm, it is known that these algorithms perform worse when a lot of unnecessary features are included in the training set [21][16][36].
This problem is called "overfitting". Overfitting means that the algorithm tries to fit the classifier variables to every feature, even the irrelevant ones. This results in classification variables that fit the training set very well, but perform poorly on data that is not in the training set. This becomes an even bigger problem when taking into account the fact that the amount of training samples is small, and the amount of noise in the samples is high. So it is important that a trained machine learning algorithm generalizes well.
There are other methods that reduce the amount of features and select more relevant ones, called feature extraction methods. Two examples are Principal Component Analysis (PCA) and Independent Component Analysis (ICA). These methods work by combining features into new ones and that way store more information in less dimensions [6]. The resulting features however, do not represent voxels anymore, so patterns between voxels can not be analysed (also referred to as multi-voxel pattern analysis, or MVPA). The fact that features in fMRI data represent parts of the brain brings us to another argument for using feature selection rather than extraction.
Every application of fMRI has different features that are important. If you are trying to determine if a person lies you are probably looking for very different features than if, say, you are trying to determine if a patient is depressed. Automated feature selection can be a big help in selecting the right brain regions for a certain application. Consequently, feature selection has the potential to be used in finding the brain regions responsible for a particular task, and not only make a classifier be more efficient. This is also another point against feature extraction methods [44].
It may be clear that finding a good and small subset of all the available voxels is important in efficient classification of fMRI data. In the next chapter some well-known and some more recent potential feature selection methods will be discussed.

4 | Evaluation of different feature selection methods on fMRI data

Now that we understand the need for feature selection in machine learning applications on fMRI data, we are going to cover some feature selection methods and evaluate whether or not they have potential in the application of fMRI. We cover a broad selection of methods from different classes to get a good view of the different methods that are available, and some recent methods to see where this research topic is heading. We start off with the simpler filter methods, after which some wrapper methods will be discussed and finally a few embedded methods will be evaluated. This order is chosen to steadily increase the readers understanding of more specific feature selection methods without throwing them in the deep immediately.
Every method will be elaborated, the idea and underlying techniques will be explained, after which a short discussion will follow about that specific method in relation to fMRI data. This will also include a mini-conclusion that will only have the current method in scope. All these conclusions are aggregated and discussed in the final discussion and conclusion.

4.1 Univariate Methods

Univariate methods are probably the simplest feature selection methods. These methods are univariate filters. This means that they rank each feature individually according to some statistic, and then select the desired amount (let us call that n) of most favourably ranked features. Because of this single variate nature, they fail to recognize any correlations or dependencies between features. This can result in selecting redundant features, for example when two neighbouring voxels both have a high ranking, but are showing very similar patterns of activation, then selecting one of them might be just as useful as selecting both. Or not selecting features that have no meaning on their own but might have in combination with other features (see [17] for a more thorough overview of what information might be missed when using univariate methods.) On the other hand, the fact that each feature only has to be considered once means that the search space is very small compared to multivariate methods. This makes that these methods are the fastest running feature selection methods that are covered in this paper (see [40] for a comparison of running times for different classes of feature selection methods).
In this chapter we cover two univariate filters: discriminability based and activity based feature selection. Despite their simplicity we have included them in this review because they have been used a lot in earlier research in machine learning, they provide a basic understanding of what feature selection in fMRI data envelops and they may still have a meaning in future research on this topic.

Discriminability based feature selection: The most common applications that use feature selection are classification problems. In fMRI that translates to distinguishing between different brain states. For example, whether a subject is reading a sentence about houses or reading a sentence about mice. To find relevant features for such a problem, we could select those that best *discriminate* the two classes. To achieve this, a separate classifier is trained for each voxel. The accuracy of every classifier on the training data is used as a measure of the discriminating power of the voxel. The n voxels that have the best scores according to this measure are selected[31].

Activity based feature selection: Most often when fMRI data is achieved, not only measurements are obtained of a brain that is busy with some task, but also of the brain in some "resting state". For example fixating on a specified place on a wall. To distinguish i activation classes y, we would perform a t-test on every voxel and every class y_i, to compare the voxel's activity in examples when it is in class y_i and when it is in resting state. Then, for each class, the voxels with the highest t-value are selected, then the second highest and so on until n voxels are chosen[31].

Discussion: We now have two basic methods for selecting features for fMRI data: based on the discriminability of features between classes and based on the activity of features compared to a resting state. Despite their similarity, research has shown that the activity based method has a considerable edge over the discriminability based method[37][31].
Overall, these methods are simple and fail to take into account a lot of the (usefull) properties of fMRI data that other, more advanced, methods do. This is why most of the more recent research has gone into the so-called multi-voxel pattern analysis (MVPA)[36].
Despite this, these methods still have meaning even with far more complex methods available. Because the more advanced MVPA methods are usually far more computationally expensive and have a much bigger search space, preprocessing the features with a univariate method can significantly benefit the classifier performance[9].

4.2 Information based feature selection

As an alternative to activation based feature selection described above, Kriegeskorte *et al.* [23] proposed in 2006 a multivariate filter method that makes good use of the properties of fMRI data. This methods scans the brain with a "searchlight", multivariately comparing the voxels on

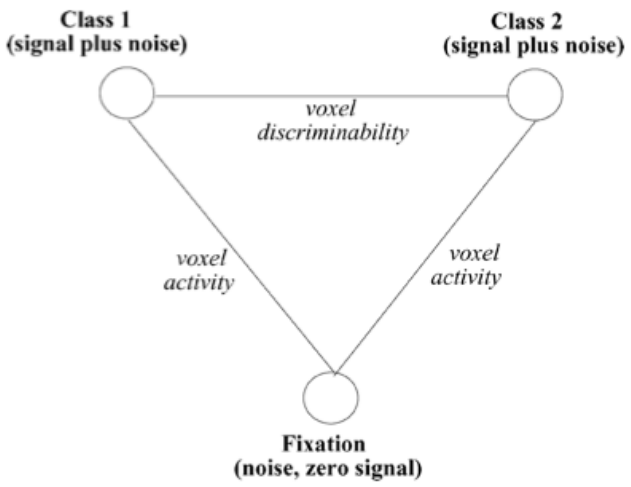

Figure 4: It is customary in fMRI research to obtain a measurement when the subject is not doing a task (is in "fixation" or "resting state"). This measurement is treated as only containing noise, without a signal. This can then be compared to a measurement of a subject that *is* doing a task (with signal). This method of feature selection is referred to as activity based. When directly comparing two measurements of a subject doing different tasks, we *discriminate* between two classes and refer to this method of feature selection as discriminability based. (Image from Mitchell et al.[31])

which the light shines. We include this method because it shows how clever use of information or "supervision" can help in feature selection.

Overview: Kriegeskorte *et al.* reason that in functional brain mapping (as is the goal in *functional* MRI analysis), so called functional regions are expected to more or less be activated as a whole. In fMRI scans, these functional regions span multiple voxels. This promotes the idea of smoothing the data spatially, for example with a Gaussian averaging function[13]. When smoothed, the data contains less, as Kriegeskorte *et al.* call it, "salt-and-pepper" fine structure. This makes the average activity of functional regions statistically more valuable, and thus the detection accuracy is increased. However, this fine structure may contain significant and relevant information that is unused in this way. Especially when the resolution and accuracy of the fMRI scanners increase, it would be a shame to overlook all this information.

So, instead of the activation based approach with spatial smoothing, Kriegeskorte *et al.* propose to use a spatial "searchlight". For every voxel, they calculate the mahalanobis distance[8] between two activity patterns of a spherical volume around it. This information is then used to perform statistical inference to obtain a map of p-values. This p-map is in turn thresholded so that the expected amount of falsely marked voxels is not higher than 5%.

Conclusion: Kriegeskorte *et al.* found that their searchlight method was more sensitive than both smoothed and unsmoothed activity based approach when tested on human subjects, and significantly more accurate when tested on simulated data. Also, the search space is only a linear increase over that of the univariate methods (for every voxel a constant amount of other voxels have to be considered), so the increase in computational complexity is not significant. On the other hand, there might be relations beyond the functional regions or between functional regions that are not considered with this method. So while using significantly more information than the univariate methods described earlier, there might still be information that is overlooked in this information based feature selection method.

4.3 ReliefF

ReliefF is a multivariate filter method. It is the extended version of the original Relief algorithm, which was used for two-class problems. ReliefF, on the other hand, can perform feature selection in datasets with multiple classes, which is necessary when dealing with fMRI datasets containing instances from more than two categories.

Overview: The basic idea of ReliefF is that features are assigned greater or lesser weight based on how useful they are in distinguishing neighboring instances from different classes [38]. During its execution, ReliefF will assign weights to features and at the end, only features passing a certain threshold value will be selected.

An important issue is how to define what "neighboring" means when it comes to calculating which instances are closest to each other. Different similarity functions can be used for this purpose, and their outputs can be combined in different ways. When dealing with numerical features we can, for example, take the sum of the differences between each pair of values (known as the Manhattan distance). However, the common characteristic of the various similarity functions is that all features are included in the search for neighbors, so that this search is being done in a multivariate way and feature dependencies are included in it.

To see the idea of the algorithm, let's first consider Relief and after that discuss in which ways ReliefF differs from its predecessor.
Consider the following case: we take two instances belonging to different classes and consider one feature. If these two instances have the same value for this feature, Relief will "punish" this feature for failing to differentiate the instances. On the other hand, if they have different values for this feature, Relief will "reward" this feature as it serves as an indicator of the class difference.

This is an example run of Relief, with a user defined number of iterations k and threshold value t:

1. Take an instance i at random
2. Find the nearest instance s belonging to the same class as i (called the nearest hit)
3. Find the nearest instance d belonging to the other class (called the nearest miss)
4. For each feature f in i:
    - If f has different values in i and s, decrease weight of f
    - If f has different values in i and d, increase weight of f
5. Repeat above process k times and return all features with weight $> t$.

As can be seen from the algorithm, Relief works when only two classes are present in the problem. In ReliefF, n nearest hits and misses are taken instead of one nearest hit and nearest miss (the value of n can be chosen). If there are m classes, the feature weight adjustment will be based on the contributions of $m \cdot n$ number of instances.

Discussion: The main strength of ReliefF when it comes to analysing fMRI data is its ability to detect dependencies between features. This is inherent in the way neighboring instances are determined, as the whole feature set is taken into account for this. If, for example, belonging to some class requires a high value for a certain feature and a low one for some other feature, then this piece of information will be included in the search for the closest instances. We have already mentioned the interdependent nature of the feature space associated with fMRI data; this aspect of ReliefF is therefore nicely suited to deal with this issue.
The basic difference between Relief and ReliefF, namely the selection of multiple instances from each class instead of one, is another important point when considering the application of ReliefF to

fMRI data. As noted before, fMRI data typically contains a significant amount of noise. This issue is addressed by ReliefF by considering multiple instances and averaging their impact, increasing the algorithm's robustness against noise. A fine-tuning strategy when applying ReliefF to an fMRI dataset known to be noisy, could therefore be to increase the value of n. The downside of this is of course the corresponding increase in computational complexity, since the calculation of the closest instances is the most complex operation in the algorithm.
This last point seems to predominate the assessment of ReliefF when applied to fMRI data, since noise is such an important factor there. In Zhang *et al.*[43], for example, a similar application of ReliefF is investigated (to a gene selection problem). There it is used only as a preprocessing step to filter out a part of the redundant features and reduce the workload for the next step in the feature selection process.

4.4 Sequential Search Methods

Sequential search methods are a branch of wrapper-methods characterized by their efficient running time and iterated hill-climbing algorithmic structure [3], where a proposed solution is gradually improved upon after each iteration. In this section the general concept behind sequential methods will be introduced. Next, some well-known examples of such methods will be mentioned. Then, the attributes of sequential search will be evaluated in the context of fMRI data and lastly an overall conclusion will be drawn regarding these methods' usefulness in the fMRI domain.

Overview: As discussed in section 3.1, the goal of wrappers in general is to find an optimal subset. However, since traversing the entire feature subset-space is impractical, they employ a variety of search-algorithms that allow them to traverse that subset-space more efficiently and thus improving their estimation of the optimal subset after each iteration of the algorithm.
These search-algorithms can be divided into three main groups: exponential, randomized and sequential [10]. Exponential algorithms (as the name suggests) become painfully slow with larger input sizes, and therefore are not suitable for fMRI data analysis. Randomized methods however have no such performance drawbacks but rather tuning difficulties caused by the random nature of the algorithm. Sequential algorithms on the other hand have an efficient running time and are predictable enough to make tuning not a tedious process.

Sequential search algorithms are iteration based, and have the following structure:

1. Start with a predefined subset of features
2. Perform analysis on potential candidates to add or remove from the subset
3. Add the feature-candidates that improve the subset's performance the most
4. Remove the feature-candidates that hinder the subset's performance the most
5. Repeat until addition and/or removal does not trigger any improvement in performance

Some well-known such search algorithms are Forward Sequential Selection (FSS) and Backward Sequential Selection (BSS). FSS starts with an empty subset, and each iteration adds the feature that improves the subset's performance the most. BSS is the mirroring of FSS; the initial subset comprises the entirety of the feature-set and each iteration the most harmful feature is removed. Measuring the impact of a feature's addition/subtraction is done via the classifier, acting as the black-box.

Discussion: The efficient running time of sequential search methods is definitely a plus when it comes to large data clusters such as in the context of fMRI. Another advantage to name is the inclusion of relationships between multiple features, an aspect that most filter methods lack. The advantages compared to other wrappers that use exponential search algorithms are clear cut, however it is hard to tell whether they offer a better alternative to randomized methods, as those

often times are capable of achieving great accuracy as well [3].
Such a randomized method is discussed in section 4.9 - memetic algorithms. It can be argued that its random, evolutionary nature will yield subsets that sequential methods will skip but have just as much (if not more) merit. Also, since memetic algorithms have more user-defined parameters available to test with, they can be fine-tuned to a better degree than sequential methods can. The drawback is of course an increased tuning-phase of those parameters, but this flexibility is nonetheless an advantage relative to sequential methods.
Lastly it must also be taken into account that although sequential search methods are efficient, their polynomial complexity power might still be too costly for such high dimensional datasets as are found in the fMRI field.
Sequential search methods therefore offer an intuitive process that ultimately is capable of finding the most relevant features in a given problem instance. However it is debatable whether it is the best-suited method for extremely large data sets as it certainly has valid competitors in that regard.

4.5 Intermezzo: Support Vector Machines

In the coming sections we will review a few methods that have something to do with Support Vector Machines (SVMs). SVM is a well-known method and we wish to explain its fundamentals before going on to evaluating feature selection methods that incorporate it. Another subject that is closely related to SVMs are norms, the L1- and L2-norm in particular. These will also be briefly explained to illustrate their significance in SVMs and in general.

Overview: Support Vector Machines (like some others) regard the set of data as points as if they are in a multidimensional vector space spanned by the feature-set. The aim of the method is to find an optimal multidimensional plane (hyperplane) that separates the different classes in a given training-set the best. The distance between a hyperplane and the closest training sample (also called the support vector) is dubbed as the margin of that plane. An optimal such hyperplane is defined to be the one that maximizes all of its margins [34]. See figure 4.5 for a demonstration of these principles. SVM-algorithms are designed to fit this optimal plane to the training set, and once that is done, it may be used against new previously unseen data.
SVMs can be used both as a classifier as well as an aid in feature selection [34] [26]. However this paper will only concentrate on its use in feature selection.

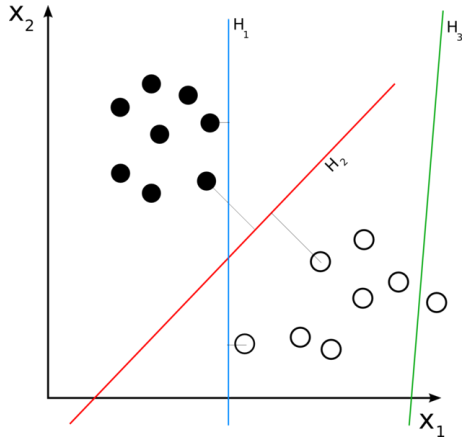

Figure 5: An example of the fitness of different hyperplanes (in this case in two dimensions) to two classes of data. Green does not fit, blue fits and is an example of a solution using the L1-norm. Red fits and is an example of a solution using the L2-norm.

Regarding Norms: In mathematics, norms are functions that assign a length to a vector. Since SVMs use the notion of lengths and distances in its algorithm for computing the optimal hyperplane, the way by which this distance is calculated affects the method's output and should be chosen with care.
The two norms that we wish to discuss are the L1- and L2-norm. The L1-norm measures the dimensional distance, that is to say, a vector $v = [1, -2, 3]$ has a total dimensional length of $|1| + |-2| + |3| = 6$. Comparing vectors using the L1-norm is akin to checking on how many dimensions they differ and to what degree.
The L2-norm is equivalent to the Euclidean length of a vector. Comparing vectors using the L2-norm therefore "mashes" the dimensions together to compute a distance.
Inside the equation of the found hyperplane, both norms assign weights to features. These correspond to their respective importance in the classification of the dataset. The L1-norm often-times nullifies certain features altogether by assigning them a zero-weight. This implicitly selects the rest of the features. The L2-norm on the other hand very seldom does this and almost always assigns some weight to each and every feature. In this case we can select features by applying a threshold on the weights, selecting only the most significant ones.
The way SVM fits the right plane to the training data is comprised of solving a maximization problem (of margins) using regularization combined with penalties (for wrongly separating datapoints). For these two often either the L1- and L2-norms are used, however in the context of fMRI it would be beneficial to use the L1-norm, as it was shown that it is less prone to overfitting when there are a large amount of irrelevant features involved [35].

4.6 Support Vector Machine - Recursive Feature Elimination

In this section we will take a look at the wrapper method Recursive Feature Elimination (RFE). As explained in chapter 3, a wrapper method treats the classifier as a black box, which makes it possible to fit different classifiers to a wrapper. Here, we will consider RFE together with the classifier Support Vector Machine (SVM).

SVM-RFE is, to a certain extent, similar to the sequential search methods discussed earlier and especially to BSS. The main difference is that at each iteration, given a feature set s of size n:

- BSS will train the classifier on all subsets of size $n-1$ and remove the feature which is absent in the best performing subset;
- SVM-RFE will train the classifier on s, calculate the scoring for each feature based on the training result and remove the lowest scoring feature(s).

Overview: RFE is a multivariate feature selection algorithm and uses the training algorithm (in our case, SVM) to recursively eliminate less useful features.
The procedure is iterative and works as follows:

1. An SVM classifier is trained on a subset of the data.
2. The training determines the discriminative value for each feature.
   As discussed in the intermezzo about SVM's, the classifier will have calculated a hyperplane; the weights assigned to the features follow from this plane.
3. The lowest scoring features are discarded.
4. The next iteration starts with the set of remaining features.

Additionally, a stopping criterion needs to be defined. We could make the algorithm stop when the original feature set is empty or when the new feature set has reached a certain size; in both cases the best feature set is given by the SVM with the highest generalization performance. A different approach is to, during the execution of the algorithm, keep a separate validation dataset to assess how well the current feature set helps generalizing. This way, we can stop the algorithm when we notice a drop in generalization performance.

Discussion: The method used to reduce the initial feature set seems to make a difference. For example, according to the findings discussed in de Martino *et al.*[9], a significantly better performance (with respect to sensitivity and generalization) was achieved when RFE was preceded by a univariate activity based reduction of features rather than a discriminability based one. The reason presented for this is that, at the level of individual voxels, the BOLD changes of a condition are greater when compared to the baseline (activity based) than when compared to some other condition (discriminability based). This combination seems especially well suited in cases where no prior knowledge is available about which features, or the features in which region of the brain, might be the best candidates to classify a certain instance.
Another important point is the setting of two of the parameters of RFE: the number of iterations to perform and the number of features to discard at each iteration. A thorough, exhaustive search requires the former to be high and the latter to be low, and this is undesirable for fMRI data because of the large number of features. Still, using the combination of feature selection methods presented in de Martino *et al.*, good results were achieved with a relatively low number of iterations (10).
To summarize, according to de Martino *et al.*, the two-step feature selection phase (univariate activation-based followed by RFE) leads to good classification performances. They assert that because of its iterative and data-driven character, no a priori assumptions are needed about the sought patterns and is therefore apt for datasets where this information is indeed unknown. In our case, this aspect is useful because of the interdependent feature space associated with fMRI data. A downside, however, seems to be the need for a specific sort of preprocessing. If RFE was preceded by a discrimination-based filter, for example, noticeably less successful results were achieved.

4.7 Spatial Support Vector Machine

Spatial Support Vector Machines is a relatively modern embedded feature-selection method [26]. As the name suggests, it incorporates the general SVM method, but adds additional processing steps in order to take advantage of some of the properties of fMRI data, and thus alter the ranking of feature for the better.

Overview: Liang et al. (2006) argued SVM's usefulness as a feature-selection method inside the fMRI niche. They proposed that if the problem of fMRI activation detection would be regarded as one of feature-selection, SVMs could then be used to great effect. Their selection method consists of two stages: Feature ranking and thresholding. In the first stage, SVM is used to rank the features according to their importance. In the second, a threshold is applied and only the features who pass it are included in the output.
They also proposed a twist on standard SVM, namely the incorporation of the spatial correlation properties of fMRI data into the algorithm. This means that while fitting an optimal hyperplane to the data, the spatial correlation of the features (voxels) is taken into account. This is done by altering the weights of different data points in the plane's equation based on their correlation properties.

Discussion: Support Vector Machine methods are in general well suited for high-dimensional data [41], which makes them very appealing for fMRI analysis, of which high-dimensionality is an essential property.
Since the Spatial SVM is a non-probabilistic method, this exempts the user from having to perform repeated runs in order to gauge the method's benefit using certain parameters, which makes the tuning of these parameters easier than some randomized methods, such as Random Forest and Memetic algorithms (sections 4.8 and 4.9).
The incorporation of prior knowledge regarding the spatial correlation of fMRI features (voxels) is welcome, and was found to increase the model's accuracy significantly [26]. This demonstrates the ability of embedded methods to sacrifice flexibility in favor of greater accuracy than their

wrapper-counterparts.
One drawback of SVM is its complex structure, making it hard to implement and even harder to do so efficiently.
Spatial Support Vector Machines is a definite improvement on the general SVM model when it comes to fMRI. It is resistant to the high-dimensionality of the data. It is non-probabilistic and therefore consists of a brief training process. Lastly, its incorporation of prior knowledge regarding the correlation between voxels in fMRI scans gives it an edge above more universal methods, namely wrappers. Its practical drawback is its structural complexity.

4.8 Random Forests

Random forests are a modern and probabilistic take on the decision tree type methods, which are of the embedded method type. The idea behind Random Forests (henceforth RF) will be elaborated firstly, after which a discussion of the method applied (theoretically) to fMRI data and a short conclusion follows. RF is a widely used method, since it has a very generalised structure and its randomness leads to a natural feature selection and classification.

Overview: As said, RF is a new take on decision trees (DT), which means that a set of trees will be generated or "grown" based on a (set of) features, thus the term forest. The data used for generating these trees is called "bagged" (aggregated bootstrap) data: from a search space of n samples by d features, n samples are chosen (with replacement) with a random subset of d features. This means that some sample sets may be in the bag multiple times, but because of the random subset of features for each sample, these are unlikely to be the same. By doing this, not every feature will be included in the set, so the search space is reduced [5].
After producing a training set, the trees are grown and evaluated. This algorithm is described well by Liaw and Wiener [27]: The complete dataset is entered at the root of the tree, here a random set of features is selected (the number of which is described by a parameter k), and for this set the most differentiating (set of) features is determined with a simpler and faster method that has been selected for this RF setup. These features will form the new "node", branching into a 'yes' and a 'no' instance for these particular features, meaning that if in the new sample the features match the features for the node, the algorithm will continue down the 'yes' branch, or the 'no' branch otherwise. This way the tree will continue to grow, until the desired size is achieved, and this way a whole forest is grown. The class the sample is classified as for a specific tree, is determined by the leaf the sample ends up in. The final classification is done with a majority vote of all the trees for the same sample, as explained in Genuer *et al.* [14]: "(...) for a new observation, each tree predicts a class and RF finally returns the most popular class". By bagging and randomising the features to use for generating the forest, the feature importance can be determined quite easily by calculating the prediction error when only that feature is permuted (excluding or including that feature). There actually are four different definitions, which are described by Breiman [5].
RF differs from DT in that the subset from which a tree will be grown, is selected randomly. To maximize the effect of this random selection, the samples from the learning set are bagged. But this also introduces a new type of error, the Out Of Bag error (OOB). This is done by fixing one data sample, and gather all bootstrapped sets that do not contain this sample, thus making this sample out of bag. Afterwards put the new bag through the selection and a prediction error can be obtained for this particular sample, which is aggregated for all trees. This can be done for some borderline-crucial features (features that are or are not included in the set depending on the tuning) to fine-tune the selector and classifier.
An interesting effect is the convergence of the forest: at a certain point the addition of another tree to the forest does not improve the classification. This seems sort of counter-intuitive, as the forest is generated randomly from a randomly bagged set of samples. But that is, in essence, the reason for convergence. With a certain number of trees in the forest, a certain number of classification problems (e.g. according to this (set of) features, does this sample belong to class x or y) is covered. When no new classification problems can be resolved by adding a tree, the classification accuracy will not increase.

Another aspect of this method being probabilistic is that the generation of trees can be adapted to the specific needs of the problem. Should the focus be more on the clustering or on the contrasts between voxels, the generation algorithm can be adjusted. This way the algorithm is easily modifiable and the user can steer the algorithm towards the area of interest.
The true feature selection part of this method can be somewhat difficult to pinpoint, because it is embedded in the whole process. Selecting features is done ad random, after which the natural selection determines (through the voting process) which selection of these random selections would be the best. This is very normal for embedded methods, since they all incorporate the feature selection and classifier construction in one process by definition. This does mean that feature selection with Random Forests can be done by evaluating the trees after training and tuning: the more the feature is found in the trees, the more important the feature is considered important. This can then be used for feature selection.

Discussion and Conclusion: One of the first issues marked by the more random mechanics in RF that there is a very obvious spatial correlation between features, as these are voxels, spatially distributed measuring points. A bare implementation of the RF will not use that correlation and will make overly complex classifiers. One way to resolve this is by integrating local search to address the clustered nature of the voxels when growing a tree. An upside to that randomness is that prediction models of RF converge for large datasets, based on observations made by Breiman [5]. Bagging the data also means that while there is some variance inside, it remains mainly consistent over the whole set. This makes for easier OOB error calculations and gives more consistent results. Another point to be made is the idea behind RF: by using a more "natural" approach to learning algorithms, more complicated (e.g. even higher dimensional with more multivariate dependencies) data is no problem for a properly set-up RF algorithm. While being a very different method, this "natural" selection idea also is used in the memetic algorithms, but more on those in section 4.9. All in all, RF is a fairly complex method, but is also easily modifiable for the user's specific needs. In this case for fMRI data classification, with some minor adjustments for the inclusion of spatial correlation.

4.9 Memetic algorithms

Some algorithms can adjust themselves for the type of work and input they get, they can "evolve" so to say. They adjust by only selecting the strongest features and methods, following rules inspired by natural selection. These types of machine learning algorithms are a subtype of evolutionary algorithms. Memetic algorithms (MA for short) are a subclass modified and tweaked for analysis of clustered voxel data such as fMRI, which makes them of the embedded methods class.

Overview: Memetic Algorithms are a subclass of Genetic algorithms (GA), which are a subclass of Evolutionary Algorithms (EA). These are heuristic algorithms that mimic natural selection methods like inheritance, mutation and selection to improve the inner methods, in essence solving optimisation problems in a natural way [12] [30]. The improvement on "classic" heuristic algorithms is that EA's can be designed with directed mutation and guided inheritance. These two factors result in a method for optimisation that can "evolve" faster but also in a guided way that ensures a solution within the desired spectrum [15].
Just like genetics, within a set of features, there is a chromosome. This means (in computer science) that from that "bag of features" (see explanation on *bagging* in 4.8), only certain features are randomly selected to be in the model. This gives a chromosome encoding of 100101101 for example, with each element representing if that feature is in the model (a 1) or not (a 0). After generating the gene-pool, selection is done with a derivative of sequential search (backwards elimination or forward selection 4.4), looking for the most descriptive chromosomes according to the fitness function.
After selection, the "evolutionary operations" are executed, meaning that a pair of chromosomes is chosen, which will generate "offspring". Firstly the matching features are selected (selective

inheritance), and after that the most descriptive features, again using the fitness function, but this time with small mutations in the genetic code, in places the parents do not match up (guided mutation). This is done until a new pool of appropriate size is grown. To keep the gene-pool healthy, also the less fit ones are kept in the pool [30] [33].
Where MAs improve on GAs is the ability to search locally in the dataset, instead of only searching globally [32]. This is done to tackle the problems "standard" GAs have with volumetric and clustered data (e.g. fMRI) and converge faster and more stable for volumetric data: "*(...) the GA's lack of precision and, consequently, unstable convergence.*"[4]. This local search also makes MAs usable for simultaneous multiple region detection, which is one of the most difficult part of interpreting fMRI data.
The convergence is measured after multiple runs by looking at the prediction error with (again) a modified form sequential search, both on the number in the gene-pool and which generation of the gene-pool is used. There is a set convergence threshold at which the selected chromosomes are considered "trained". Decisions are made with a majority vote, much like Random Forests (section 4.8).
Because of the reproduction of the strongest features, overfitting is a real issue when running the algorithm too many times. This is combated by using a substantial amount of the data for fitness calculation and correction, while still using the majority of it for parameter estimation [2] [1].
The modifiability argument (see section 4.8) is also valid for Memetic Algorithms, but in this case the evolution phase can be adjusted to "evolve" in a certain direction.
Again, since this is an embedded method, the true feature selection part of the process is difficult to find (see Random Forests, section 4.8). The randomized starting pool is already a form of feature selection, although, since it is random, not a very good one. But the natural evolution process "selects" (or better, generates) a feature set that is optimised according to the specified fitness function. This set can be extracted from the resulting gene-pool by selecting the features that occur the most in the chromosomes, the more the feature can be found in the pool, the more important it can be considered.

Discussion and Conclusion: Given the high dimensional nature of fMRI data (spatial as well temporal, and each voxel has his own intensity), MAs look like the best way to easily apply feature selection on fMRI datasets, since the underlying selection mechanics are based on natural evolution and selection. The downside is that these kind of algorithms take a lot of tweaking and training before they generate a really good classifier or feature-set, which is a problem due to the not so abundant fMRI data. However, the first tuning steps are easy because of the evident convergence that occurs in the early stages.
Because of the continuous adaptation of the fitness function, MAs are insensitive for the noisy aspect of the data, and can generate a more generalised model (to counter the fact that there are differences in the neural activity centra between people).
The local search addition enables MAs to address the correlation between and within clusters of voxels, while still using the global scope for differences between clusters.
In short: when a more standardised approach for ME's in regard to fMRI data is designed, this has a lot of potential. Like Random Forests, Memetic Algorithms have a lot of potential due to their solving power and flexibility. But implementing them correctly is still very difficult, so a more standardised or self-regulating system has to be designed before their true potential can be reached.

Discussion and conclusion

In this paper we analyse the application of feature selection to fMRI data. The main challenges in the analysis of fMRI data are the large feature-to-instance ratio, the considerable amount of noise and the interdependency between voxels.
These characteristics of the feature space associated with fMRI data necessitate the usage of feature selection methods, the goal of which is to select the most relevant features to be used for

classification. If this step is omitted, the classifier which will use every feature will be less accurate due to redundant features ('overfitting'), and will simply be slower because of the sheer size of the feature set.
To this end, we evaluate several feature selection methods with respect to their effectiveness in analysing fMRI data. We include several methods from each of the three main categories of feature selection algorithms:

1. *Filters* assess the usefulness of features by only looking at the data itself;
2. *Wrappers* include a classifier in the selection process and determine the score of a feature subset by the classifier's accuracy on that subset;
3. *Embedded methods* are similar to wrappers but are designed for a specific classifier.

Key points in our assessment are speed, accuracy, robustness against noise and the ability to capture dependencies between features. Modifiability and ease of training are additional concerns for wrappers and embedded methods. We summarize our findings in table 1 and elaborate on them afterwards.

Class	**Method**	**Advantages**	**Disadvantages**
Filter	Univariate	Fast; useful as preprocessor	Ignores dependencies
	Information based	Fast; useful as preprocessor	Only captures local dependencies
	ReliefF	Captures dependencies	Sensitive to noise Slow for large datasets
Wrapper	Sequential search	Simple and efficient	Not fast enough
	SVM-RFE	Fast and accurate	Dataset must be pre-analysed
Embedded	Spatial SVM	Fast training Use of spatial correlation	Complex implementation
	Random forest	Fast and accurate Highly modifiable	Complex implementation
	Memetic algorithms	Basic tuning easy Highly modifiable	Might require long training

Table 1: An overview of the advantages and disadvantages of various feature selection methods when applied to fMRI data

The first impression is that, in general, filter methods lack the necessary sophistication to successfully handle data as complex as that of fMRI. We found their main shortcomings to be sensitivity for noise and inability to detect feature dependencies. For ReliefF, for example, robustness against noise could in principle be achieved, but only at the cost of a significant increase in computational complexity. They can, however, be useful as preprocessing steps, in order to remove some of the redundant features and reduce the workload of the next step in the feature selection phase.

We find more positive results for the wrappers that we analyse, both of which are good options for finding relevant features in a given dataset. The common drawback is their algorithmic complexity, caused mainly by the requirement to capture relations between features. In the case of SVM-RFE, for example, this can only be done efficiently if the RFE algorithm is preceded by the initial pass of a filter method (in this case, activity based feature selection).

It seems that the best results when selecting features from fMRI data can be achieved with the usage of embedded methods. They are resistant to the high-dimensionality of the feature space, select features accurately and are robust against noise. The probabilistic methods that we analyze (random forests and memetic algorithms) also have efficient running times.

An interesting aspect of the embedded methods that we review is the trade-off that emerges between training performance and accuracy. The probabilistic ones have a longer training phase and a higher accuracy, while for the deterministic methods the opposite is the case. An explanation for the higher accuracy could be that, because of their random/evolutionary nature, these methods might include subsets that would otherwise be skipped by the deterministic methods. Trying to do this deterministically by evaluating all possible subsets would be infeasible because of the large feature space. The longer training phase is caused by having to execute a number of test runs before the parameters of these probabilistic methods can be set effectively; this is problematic because, relative to the number of features, the number of instances that can be used is small.

To summarize, the probabilistic embedded methods look to be the best candidates for feature selection from fMRI data. One of the main challenges here is implementation complexity and this applies in two ways. The first is that both random forests and memetic algorithms are difficult to implement in themselves. Besides this, a second challenge is 'modifying' them; that is, implementing them in such a way that they can optimally handle the specific challenges inherent to fMRI data and can make use of any prior knowledge that might be available about a dataset. A final challenge is devising optimal training strategies for these methods, so that high training performances can be achieved even with the relatively sparse fMRI data.

We feel that a lot of value can be added by research focusing on how best to implement random forests and memetic algorithms for the specific case of fMRI datasets. Purposeful and efficient implementations of these algorithms hold a lot of promise for the accurate selection of the most relevant parts of neural activity mappings, which will in turn improve the accuracy and efficiency of the methods used to classify these.

Acknowledgements

We would like to thank Jesse Krijthe and Veronika Cheplygina for their constructive comments on the paper and helping to understand the subject matter. This paper has been written in commission of the Delft University of Technology.

References

[1] M Björnsdotter Aberg and Johan Wessberg. An evolutionary approach to the identification of informative voxel clusters for brain state discrimination. *Selected Topics in Signal Processing, IEEE Journal of*, 2(6):919–928, 2008.

[2] Malin Björnsdotter Åberg, Line Löken, and Johan Wessberg. An evolutionary approach to multivariate feature selection for fmri pattern analysis. pages 302–307, 2008.

[3] David W Aha and Richard L Bankert. A comparative evaluation of sequential feature selection algorithms. In *Learning from Data*, pages 199–206. Springer, 1996.

[4] Malin Bjornsdotter and Johan Wessberg. A memetic algorithm for selection of 3d clustered features with applications in neuroscience. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 1076–1079. IEEE, 2010.

[5] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[6] Thomas A Carlson, Paul Schrater, and Sheng He. Patterns of activity in the categorical representations of objects. *Journal of Cognitive Neuroscience*, 15(5):704–717, 2003.

[7] Christos Davatzikos, Kosha Ruparel, Yong Fan, DG Shen, M Acharyya, JW Loughead, RC Gur, and Daniel D Langleben. Classifying spatial patterns of brain activity with machine learning methods: application to lie detection. *Neuroimage*, 28(3):663–668, 2005.

[8] Roy De Maesschalck, Delphine Jouan-Rimbaud, and Désiré L Massart. The mahalanobis distance. *Chemometrics and intelligent laboratory systems*, 50(1):1–18, 2000.

[9] Federico De Martino, Giancarlo Valente, Noël Staeren, John Ashburner, Rainer Goebel, and Elia Formisano. Combining multivariate voxel selection and support vector machines for mapping and classification of fmri spatial patterns. *Neuroimage*, 43(1):44–58, 2008.

[10] Justin Doak. *An evaluation of feature selection methods and their application to computer security*. University of California, Computer Science, 1992.

[11] Christine Ecker, Andre Marquand, Janaina Mourão-Miranda, Patrick Johnston, Eileen M Daly, Michael J Brammer, Stefanos Maltezos, Clodagh M Murphy, Dene Robertson, Steven C Williams, et al. Describing the brain in autism in five dimensionsmagnetic resonance imaging-assisted diagnosis of autism spectrum disorder using a multiparameter classification approach. *The Journal of Neuroscience*, 30(32):10612–10623, 2010.

[12] Candida Ferreira. Gene expression programming: a new adaptive algorithm for solving problems. *arXiv preprint cs/0102027*, 2001.

[13] Karl J Friston, Andrew P Holmes, Keith J Worsley, J-P Poline, Chris D Frith, and Richard SJ Frackowiak. Statistical parametric maps in functional imaging: a general linear approach. *Human brain mapping*, 2(4):189–210, 1994.

[14] Robin Genuer, Vincent Michel, Evelyn Eger, and Bertrand Thirion. Random forests based feature selection for decoding fmri data. In *Proceedings Compstat*, number 267, pages 1–8, 2010.

[15] David E Goldberg and John H Holland. Genetic algorithms and machine learning. *Machine learning*, 3(2):95–99, 1988.

[16] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *The Journal of Machine Learning Research*, 3:1157–1182, 2003.

[17] Isabelle Guyon, Steve Gunn, Masoud Nikravesh, and L Zadeh. Feature extraction. *Foundations and applications*, 2006.

[18] Stephen José Hanson and Yaroslav O Halchenko. Brain reading using full brain support vector machines for object recognition: there is no face identification area. *Neural Computation*, 20(2):486–503, 2008.

[19] J. Haxby, M. Gobbini, M. Furey, A. Ishai, J. Schouten, and P. Pietrini. Faces and objects in ventral temporal cortex (fmri). http://dev.pymvpa.org/datadb/haxby2001.html, 2001.

[20] Ian Jolliffe. *Principal component analysis*. Wiley Online Library, 2005.

[21] Ron Kohavi and George H John. Wrappers for feature subset selection. *Artificial intelligence*, 97(1):273–324, 1997.

[22] F Andrew Kozel, Kevin A Johnson, Qiwen Mu, Emily L Grenesko, Steven J Laken, and Mark S George. Detecting deception using functional magnetic resonance imaging. *Biological psychiatry*, 58(8):605–613, 2005.

[23] Nikolaus Kriegeskorte, Rainer Goebel, and Peter Bandettini. Information-based functional brain mapping. *Proceedings of the National Academy of Sciences of the United States of America*, 103(10):3863–3868, 2006.

[24] Ludmila I Kuncheva, Juan J Rodríguez, Catrin O Plumpton, David EJ Linden, and Stephen J Johnston. Random subspace ensembles for fmri classification. *Medical Imaging, IEEE Transactions on*, 29(2):531–542, 2010.

[25] Thomas Navin Lal, Olivier Chapelle, Jason Weston, and André Elisseeff. Embedded methods. In *Feature extraction*, pages 137–165. Springer, 2006.

[26] Lichen Liang, Vladimir Cherkassky, and David A Rottenberg. Spatial svm for feature selection and fmri activation detection. In *Neural Networks, 2006. IJCNN'06. International Joint Conference on*, pages 1463–1469. IEEE, 2006.

[27] Andy Liaw and Matthew Wiener. Classification and regression by randomforest. *R news*, 2(3):18–22, 2002.

[28] Donald W McRobbie. *MRI from Picture to Proton*. Cambridge University Press, 2007.

[29] Ravi S Menon, Seiji Ogawa, Seong-gi Kim, Jutta M Ellermann, Hellmut Merkle, David W Tank, and Kamil Ugurbil. Functional brain mapping using magnetic resonance imaging: Signal changes accompanying visual stimulation. *Investigative radiology*, 27:S47–S53, 1992.

[30] Melanie Mitchell. *An introduction to genetic algorithms*. MIT press, 1998.

[31] Tom M Mitchell, Rebecca Hutchinson, Radu S Niculescu, Francisco Pereira, Xuerui Wang, Marcel Just, and Sharlene Newman. Learning to decode cognitive states from brain images. *Machine Learning*, 57(1-2):145–175, 2004.

[32] Pablo Moscato and Carlos Cotta. A gentle introduction to memetic algorithms. In *Handbook of metaheuristics*, pages 105–144. Springer, 2003.

[33] Pablo Moscato and Carlos Cotta. A modern introduction to memetic algorithms. In *Handbook of Metaheuristics*, pages 141–183. Springer, 2010.

[34] Janaina Mourão-Miranda, Arun LW Bokde, Christine Born, Harald Hampel, and Martin Stetter. Classifying brain states and determining the discriminating activation patterns: support vector machine on functional mri data. *Neuroimage*, 28(4):980–995, 2005.

[35] Andrew Y Ng. Feature selection, l 1 vs. l 2 regularization, and rotational invariance. In *Proceedings of the twenty-first international conference on Machine learning*, page 78. ACM, 2004.

[36] Kenneth A Norman, Sean M Polyn, Greg J Detre, and James V Haxby. Beyond mind-reading: multi-voxel pattern analysis of fmri data. *Trends in cognitive sciences*, 10(9):424–430, 2006.

[37] Jay Pujara. *Understanding feature selection in functional magnetic resonance imaging*. PhD thesis, Carnegie Mellon University, 2005.

[38] M. Robnik-Sikonja and I. Kononenko. Theoretical and empirical analysis of relieff and rrelieff. *Machine Learning*, 2003.

[39] Serge ARB Rombouts, Frederik Barkhof, Rutger Goekoop, Cornelis J Stam, and Philip Scheltens. Altered resting state networks in mild cognitive impairment and mild alzheimer's disease: an fmri study. *Human brain mapping*, 26(4):231–239, 2005.

[40] Yvan Saeys, Iñaki Inza, and Pedro Larrañaga. A review of feature selection techniques in bioinformatics. *bioinformatics*, 23(19):2507–2517, 2007.

[41] Vladimir Vapnik. *The nature of statistical learning theory*. springer, 2000.

[42] Liang Wang, Yufeng Zang, Yong He, Meng Liang, Xinqing Zhang, Lixia Tian, Tao Wu, Tianzi Jiang, and Kuncheng Li. Changes in hippocampal connectivity in the early stages of alzheimer's disease: evidence from resting state fmri. *Neuroimage*, 31(2):496–504, 2006.

[43] C. Ding Y. Zhang and T. Li. Gene selection algorithm by combining relieff and mrmr. *BMC Genomics*, September 2008.

[44] Okito Yamashita, Masa-aki Sato, Taku Yoshioka, Frank Tong, and Yukiyasu Kamitani. Sparse estimation automatically selects voxels relevant for the decoding of fmri activity patterns. *NeuroImage*, 42(4):1414–1429, 2008.

[45] Lei Zhang, Dimitris Samaras, Dardo Tomasi, Nora Volkow, and Rita Goldstein. Machine learning for clinical diagnosis from functional magnetic resonance imaging. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 1211–1217. IEEE, 2005.