Faculteit der Sociale Wetenschappen

Can dimensionality reduction through feature extraction improve classification accuracy compared to whole-brain analysis?

Using high-dimensional neuroimaging data as input for a Support Vector Machine to distinguish Alzheimer patients from healthy controls

Thomas Broers

Master Thesis Psychology

Methodology and Statistics Unit, Institute of Psychology,

Faculty of Social and Behavioral Sciences, Leiden University

Date: May 2017

Student number: 1264486

Supervisor: Dr. Tom F. Wilderjans, Frank de Vos (co-supervisor)

Abstract

When machine learning techniques are applied to neuroimaging brain data, some type of dimension reduction is usually conducted before training a classifier, in order to avoid overfitting and therefore improving the generalization ability of the model. Herewith, it is often assumed that dimension reduction can increase classification accuracy compared to when whole-brain data are being used. Yet, previous studies have shown that when a Support Vector Machine (SVM) is used as a classifier, feature selection does not necessarily improve classification accuracy compared to whole-brain analysis. However, feature selection methods are univariate techniques, which do not take the relationships between the original variables into account. In contrast, feature extraction methods are multivariate techniques that take interactions between the input variables into account when constructing new features. Since strong relationships between voxels are known to exist within neuroimaging data, it is hypothesized and evaluated in the current study that feature extraction might be able to increase classification accuracy compared to whole-brain analysis.

In this study, four common feature extraction methods are compared with whole-brain analysis in terms of classification performance: (1) Principal Component Analysis (PCA), (2) Independent Component Analysis (ICA), (3) Partial Least Squares Regression (PLS-R) and (4) Principal Covariates Regression (PcovR). To demonstrate the effect, data regarding seven neuroimaging properties that are believe to be related to Alzheimer were used to distinguish between people with Alzheimer's disease (AD) and healthy controls (HC's).

The results demonstrated that feature selection, and then especially PLS-R, is able to outperform whole-brain analysis in terms of classification performance. This pattern of results, however, was only observed for some but not all neuroimaging properties used. Among the feature extraction methods, PLS-R and PCA were the most stable and best performing techniques. However, PLS-R needed far less extracted components to reach its maximum classification accuracy, compared to PCA, and was the best performing technique for two of the seven datasets. In general, whole-brain analysis performs stable in terms of classification accuracy across a range of neuroimaging modalities. However, feature extraction can (modestly) increase classification accuracy compared to whole-brain analysis, but it depends on the neuroimaging modality that is adopted.

Table of contents

Section 1. Introduction		1
1.1	Feature selection	2
1.2	Feature extraction	3
1.2.1	Principal Component Analysis (PCA)	4
1.2.2	Independent Component Analysis (ICA)	5
1.2.3	Partial Least Squares Regression (PLS-R)	7
1.2.4	Principal Covariates Regression (PcovR)	9
1.3	Research questions & hypotheses	11
Section 2. Methods		13
2.1	Data	13
2.2	Procedure	15
2.2.1	General procedure	16
2.2.2	Validation approach	17
2.2.3	Classification accuracy (Step 6)	17
2.2.4	Number of components *S*	18
2.2.5	Computation time	19
2.3	Support Vector Machine (SVM)	19
2.3.1	Training the SVM (Step 3)	20
2.3.2	Applying the SVM model to the test data (Step 5)	21
2.4	Analysis specification for the feature extraction approaches (Step 2 and step 4)	21
2.4.1	PCA	21
2.4.2	ICA	23
2.4.3	PLS-R	24
2.4.4	PcovR	25

Section 3. Results 27

3.1 Classification accuracy 27

3.1.1 Voxel size 28

3.1.2 Whole-brain analysis vs. feature extraction 29

3.1.3 PCA vs ICA 32

3.1.4 With vs. without t-tests 33

3.1.5 Supervised vs. unsupervised 34

3.1.6 PLS-R vs. PcovR 34

3.2 Classification accuracy for the Partial Correlations (PC) data 35

3.3 Computation time 36

Section 4. Discussion 40

4.1 Can feature extraction increase classification accuracy? 40

4.2 Comparison among the feature extraction methods 41

4.3 Limitations of the current study 43

4.4 Recommendations for future research 44

References 48

Appendix A. Alternative R-code to perform PcovR-analysis 54

Appendix B. Full plot figures regarding classification performance 57

Appendix C. Computation times for the ALFF2, EX4 and MR4 data 60

Appendix D. Influence of pre-processing on classification results for PcovR 61

Appendix E. Number of AUC estimates for each PcovR analysis 62

Section 1. Introduction

Machine learning techniques, also called Multivariate Pattern Analysis (MVPA) techniques, have found their way into neuroimaging research for some time now. The goal of machine learning studies using neuroimaging data is often to train a classifier that can differentiate between two groups, like, for example, patients and healthy controls (HC). To achieve this goal, the assumption is made that relevant information that can discriminate between these groups is hidden in (the relationships between) various variables, such as activation levels in different areas of the brain (Linden, 2012). An illness which has received a lot of attention within this type of neuroimaging research is Alzheimer's disease (AD). For example, using a support vector machine (SVM) as classifier, Yang et al. (2011) showed that structural MRI-data can be used to distinguish between people with AD, mild cognitive impairment (MCI) and HC's. Like other studies within this field, these authors, however, stressed that methodological improvement is still necessary, since the high-dimensional nature of neuroimaging data poses serious methodological challenges that have to be addressed properly. In most classification studies that use neuroimaging data, the number of variables greatly outnumbers the number of cases, which is often referred to as the *small-n-large-p problem* or the *curse of dimensionality* (James, Witten, Hastie, & Tibshirani, 2015). As a result, without some type of selection of the most relevant features, a machine learning model has the risk of 'overfitting' the data. In that case, the predictive model becomes too much tailored towards the oddities and random noise in the training sample at hand rather than reflecting characteristics from the overall population, which may result in a model with a poor predictive - and therefore generalization - ability; this implies that the learning model is not suited to obtain accurate predictions for novel subjects.

In view of the above, some type of dimension reduction is often applied before fitting a predictive model to neuroimaging data (Mwangi, Tian, & Souares, 2014). However, whether dimension reduction truly improves classification accuracy (by preventing overfitting of the classifier) may depend on the machine learning classifier used. For example, a SVM, which is a kernel method, is known to be able to deal with the high-dimensionality of neuroimaging data. This is because a SVM classifier searches for a solution in a kernel space, which implies a reduction of the solution space when data are high-dimensional. Indeed, in that case, the number of parameters that has to be estimated is equal to the number of (non-zero) inner products between observations, instead of being equal to the number of predictors (James et

al., 2015). In other words, the number of parameters of a SVM is related to the number of observations, instead of to the number of variables, which is an advantageous feature in the *small n-large p* case. It could therefore be stated that, when there are less observations than variables, which is often true for neuroimaging data, a SVM implicitly reduces the dimensionality of the data (Chu et al., 2012) in an effective way such that a good classification performance can be obtained. Because of this, some promising results have been obtained using whole-brain neuroimaging data as input for a SVM classifier. For example, up to 96% of AD patients were correctly classified (AD versus HC), using whole-brain images as input for a SVM (Kloppel et al., 2008). Magnin et al. (2008) provided similar efforts, by also using a SVM classifier to distinguish people with AD from elderly controls. Using whole-brain structural gray matter MRI data, the authors reached classification accuracy's as high as 94.5%.

Although, when using a SVM as a classifier, dimension reduction may not be necessary to prevent overfitting, doing so can still be beneficial. Aside from practical advantages, like speeding up the testing process and enhancing the interpretation of the results (i.e., only having to look at a limited number of parameters/variables), dimension reduction may also improve classification accuracy (Mwangi et al., 2014). In the literature, two forms of dimension reduction are encountered often: feature selection and feature extraction.

1.1 Feature selection

A first way to reduce the number of input features is feature selection, in which a (small) subset of the features is selected and used as input for a classifier algorithm. It is often assumed that applying feature selection enhances classification accuracy in neuroimaging classification studies, because doing so can reduce noise and may increase the contrast/differences between the groups. But contrary to popular belief, Chu et al. (2012) showed that when a SVM is used as a classifier, feature selection methods do not necessarily improve classification accuracy when compared to using whole-brain structural MRI-data. In their study, the classification performance only improved when a priori information in the form of regions of interest (ROI's) related to the problem under study (i.e., AD and MCI in this case) was used to select the features. The authors stressed that when the sample size of the training set is large enough, feature selection without the use of a priori information does

not lead to higher classification accuracies compared to when no form of feature selection is used. In other words, due to the insensitivity of SVM's classification accuracy to high-dimensionality, it appeared not to matter whether whole-brain data was used as input features or whether just a small subset of those input features were adopted. Furthermore, Nilsson, Pena, Björkegren, & Tegnér (2006) evaluated several feature selection methods regarding their ability to improve the classification performance of a SVM using high-dimensional data, and found that none of the feature selection methods improved SVM accuracy. These results may indicate that the regularization step within the SVM itself is sufficient to obtain a good classification performance and that dimension reduction therefore is not essential.

However, feature selection has some drawbacks when it comes to extracting information from high-dimensional data for classification purposes. First of all, feature selection techniques are said to be univariate. That is, they do not take the relations between features into account when selecting the features. Especially in neuroimaging data, in which strong relationships between (neighboring) voxels are known to exist, not taking these relations between the input features into account could lead to ignoring aspects of the data crucial for classification. Another pitfall of feature selection techniques is that a lot of information from the original data is discarded. When a subset of 100 voxels is taken from an original set of 200.000 voxels, a lot of (possibly useful) information is lost in the process. Because of these drawbacks of feature selection strategies, researchers instead often adopt feature extraction techniques to perform dimension reduction.

1.2 Feature extraction

A second, and maybe better, way of reducing dimensionality is to extract new features from the original features, which will be referred to as feature extraction from now on. Feature extraction techniques use all the input features in order to construct - a smaller, limited, number - of new features, often called components. In this way, less information from the original features is lost when a subset of those newly extracted components is used for classification. Also, feature extraction techniques, in contrast to feature selection techniques, are multivariate, which means that they take relationships between the input features into account when constructing new features. Furthermore, when constructing new features, some feature extraction methods are able to use information about the classification task at hand. As

such, new features are extracted that, besides accounting for the multivariate relations between the voxels, supposedly have a good (better) predictive ability.

In light of the above, the goal of this thesis is to examine whether feature extraction can improve classification accuracy when compared to using whole-brain data. The aim of the classification is to distinguish between two experimental groups (AD vs HC) as accurate as possible. For the classification task, an SVM classifier will be adopted and features from (f)MRI data will be used. In order to find out whether or not using feature extraction before fitting a SVM classifier improves classification, in this study, a number of feature extraction techniques will be applied to various types of high-dimensional neuroimaging data. The following feature extraction techniques will be used: Principal Component Analysis (PCA), Independent Component Analysis (ICA), Partial Least Squares Regression (PLS-R) and Principal Covariates Regression (PcovR).

1.2.1 Principal Component Analysis (PCA)

A commonly adopted method for feature extraction is Principal Component Analysis (PCA), in which components are constructed that maximally explain the (variance in the) original input features. PCA decomposes a given data matrix $\boldsymbol{X}$, a n *(*cases*)* by p *(*variables*)* matrix, as:

$$\boldsymbol{X} = \boldsymbol{A}\boldsymbol{B}^{\boldsymbol{T}}, \qquad (1.1)$$

where $\boldsymbol{A}$ is a n by z matrix containing the component scores, $\boldsymbol{B}$ is a p by z matrix containing the loadings of the original variables on the components, $\boldsymbol{B}^{\boldsymbol{T}}$ (z by p) denotes the transpose of matrix $\boldsymbol{B}$ and z indicates the number of components. $\boldsymbol{A}$ and $\boldsymbol{B}$ can be obtained by means of a Singular Value Decomposition (SVD; ten Berge, 1993).

PCA has already been extensively used as a tool for dimension reduction in neuroimaging studies. For example, Koutsoleris et al. (2009) applied PCA to whole-brain grey matter data, in order to discriminate patients who were at risk for psychosis from HC's. Using features describing neuroanatomical brain-structures, these authors reached classification accuracies around 90%. PCA was also used by Zhu et al. (2008), who proposed

and applied a resting-state fMRI based classifier in order to distinguish ADHD children from normal controls. By using PCA in combination with a machine learning classifier, they obtained classification accuracies of 85%.

While PCA is a popular tool for dimension reduction in neuroimaging studies, it comes with a couple of drawbacks. First of all, PCA constructs components that follow a Gaussian-like unimodal distribution. In classification studies, however, predictor variables that follow a Gaussian/unimodal distribution might not be optimally suited for distinguishing between two different groups. Ideally, a predictor variable that discriminates well between groups should follow a multi-modal distribution with distinct peaks, one for each group. Also, there is no guarantee that the components that maximally explain the original variables (in terms of variance) will also be (maximally) related to the response variable (i.e., in our case, the variable indicating the groups). This may happen as PCA is an unsupervised learning method, which means that it does not take the response variable into account when constructing the components. Even when the PCA components to use for classification are selected by means of group information (e.g., taking the PCA components with the largest absolute univariate t-value when predicting the grouping variable), something that will be done in this study, the components itself are still constructed without the use of the response variable and are thus unsupervised.

1.2.2 Independent Component Analysis (ICA)

A dimension reduction technique that does not suffer from PCA's disadvantage of only finding approximately normally distributed unimodal components is Independent Component Analysis (ICA). The aim of ICA is to retrieve independent and non-Gaussian signals $\boldsymbol{S}$ that underlie a set of observed mixture signals $\boldsymbol{X}$ (i.e., the signals in $\boldsymbol{X}$ are linear mixtures of the signals in $\boldsymbol{S}$). In particular, in ICA, the (columns of the) data matrix $\boldsymbol{X}$ (n by p) is considered to be a linear combination of non-Gaussian (independent) components $\boldsymbol{S}$ (n by z):

$$\boldsymbol{X} = \boldsymbol{SM}, \tag{1.2}$$

where the columns of $\boldsymbol{S}$ contain the independent components, $\boldsymbol{M}$ (z by p) is a linear mixing matrix and z denotes the number of independent components. The idea of ICA is to un-mix the data by estimating an un-mixing matrix $\boldsymbol{W}$ (p by z) such that:

$$\boldsymbol{S} = \boldsymbol{XW}. \tag{1.3}$$

Various algorithms exists to find the underlying signals in $\boldsymbol{S}$ through the estimation of $\boldsymbol{W}$. These algorithms differ in the way they measure/approximate and maximize the non-Gaussianity of the source signals $\boldsymbol{S}$ (Hyvärinen, Karhunen, & Oja, 2001). A commonly adopted algorithm is the FastICA algorithm, which aims at maximizing the non-Gaussianity by means of maximizing negentropy (i.e., a normalized version of entropy), which is always positive and only equals zero when a random variable is Gaussian (Hyvärinen, 1999). Note that negentropy increases when variables become less Gaussian.

Within the FastICA algorithm, the data are first centered by subtracting the mean of each variable (i.e., column) of $\boldsymbol{X}$. The centered data $\boldsymbol{X_{cent}}$ are then 'whitened' by projecting the data onto its principal component directions: $\boldsymbol{X_{whitened}} = \boldsymbol{X_{cent}K^T}$, where $\boldsymbol{K}$ (z by p) is the whitening matrix. The FastICA algorithm then estimates an orthogonal rotation matrix $\boldsymbol{R}$ (z by z) in such a way that $\boldsymbol{S} = \boldsymbol{X_{whitened}R} = \boldsymbol{X_{cent}K^TR} = \boldsymbol{X_{cent}W}$ has columns that are as independent and as non-Gaussian as possible. To achieve this, the matrix $\boldsymbol{R}$ is identified that maximizes the negentropy approximation of the independence and non-Gaussianity of the columns of $\boldsymbol{S}$. Finally, $\boldsymbol{W}$ can be obtained as $\boldsymbol{K^TR}$.

In contrast to PCA, ICA is able to retrieve multi-modal components, which are possibly better suited for classification than the PCA components (scores). Just as PCA, ICA has already been widely used in neuroimaging classification studies. To effectively distinguish between normal and abnormal brain tissues, Chai et al. (2010), for example, successfully coupled the use of ICA with a Support Vector Machine. Douglas et al. (2011) successfully combined ICA with several machine learning classifiers, with the aim of distinguishing between the brain states ''belief'' and ''disbelief'' using neuroimaging data. Their efforts produced classification accuracies as large as 92%. Another very promising study was conducted by Yang et al. (2011), whom applied ICA to structural MRI data and used the extracted ICA component scores to train a SVM to discriminate between people with Alzheimer's disease (AD), mild cognitive impairment (MCI) and HC's. They demonstrated

that a fully automatic method based on ICA coupled with SVM for MRI data analysis can be very useful in discriminating among these three groups of subjects.

However, as is true for PCA, ICA is an unsupervised feature extraction method that does not take the response variable into account when deriving the components. Like for PCA, in this study, t-tests will be used in order to select the best ICA components for use in classification. Although such a procedure makes ICA somewhat supervised, the construction of the components is done in an unsupervised fashion nonetheless.

1.2.3 Partial Least Squares Regression (PLS-R)

In contrast to the unsupervised techniques mentioned thus far, supervised feature extraction techniques use both the predictor variables as well as the response (i.e., grouping) variable(s) to derive components (Mwangi et al., 2014). In doing so, one hopes that the obtained components are more relevant for classification than the components constructed with unsupervised methods. Partial Least Squares (PLS) is an often used "supervised" feature extraction method. Combining PLS with regression (or classification, which, in this case, boils down to performing regression with a categorical grouping variable as the response variable) is referred to as PLS-Regression (PLS-R). The main purpose of PLS-R is to build a linear model of the form:

$$\boldsymbol{Y} = \boldsymbol{XB} + \boldsymbol{E}, \tag{1.4}$$

where $\boldsymbol{Y}$ is a n by m (response variables) response matrix (note that in our case $m = 1$), $\boldsymbol{X}$ is an n by p predictor matrix, $\boldsymbol{B}$ is a p by m regression coefficient matrix, and $\boldsymbol{E}$ (n by m) is a noise matrix for the model which has the same dimensions as $\boldsymbol{Y}$. To establish the model, PLS-R estimates a weight matrix $\boldsymbol{W}$ (p by z) for $\boldsymbol{X}$

$$\boldsymbol{T} = \boldsymbol{XW}, \tag{1.5}$$

with the columns of $\boldsymbol{W}$ being weight vectors for the columns (i.e., predictor variables) of $\boldsymbol{X}$; this produces the corresponding score matrix $\boldsymbol{T}$ (n by z), which contains the scores of the n cases on z underlying "components". Using Ordinary Least Squares (OLS) regression for

predicting $\boldsymbol{Y}$ based on $\boldsymbol{T}$ results in the "regression coefficients matrix" $\boldsymbol{Q}$ (z by m), which are the loadings in the following decomposition of $\boldsymbol{Y}$ (with $\boldsymbol{T}$ being the scores):

$$\boldsymbol{Y} = \boldsymbol{TQ} + \boldsymbol{E}. \tag{1.6}$$

Defining $\boldsymbol{B} = \boldsymbol{WQ}$ yields:

$$\boldsymbol{Y} = \boldsymbol{TQ} + \boldsymbol{E} = \boldsymbol{XWQ} + \boldsymbol{E} = \boldsymbol{XB} + \boldsymbol{E}. \tag{1.7}$$

As such, the goal of the PLS-R algorithm is to find the $\boldsymbol{W}$ matrix that yields features $\boldsymbol{T}$ that explain $\boldsymbol{X}$ (i.e., components $\boldsymbol{T} = \boldsymbol{XW}$) and are related to $\boldsymbol{Y}$ (i.e., $\boldsymbol{Y} = \boldsymbol{TQ}$). The PLS-R parameters can be estimated with SVD (Beaton, Dunlop, & Abdi, 2016).

PLS-R is able to model the correlational structure between a large number of strongly correlated predictors while simultaneously taking also the relationships between the predictors and the response variable(s) into account when deriving the components (Wold, Sjöström, & Eriksson, 2001). PLS-R has been used successfully in various domains. For example, Menzies et al. (2007) used PLS-R to derive latent MRI markers from structural MRI-data that were associated to the performance on an inhibitory outcome task that is often used to diagnose obsessive compulsive disorder. Nestor et al. (2002) successfully used PLS-R to link structural MRI brain autonomy measures to neuropsychological test scores from people with schizophrenia. Although PLS-R has been designed for continuous response variables, evidence exists that PLS-R also yields a good classification accuracy when using a binary response variable and this especially in the case of high-dimensional data (Nguyen & Rocke, 2002).

The predictive aspect of PLS-R makes it a very useful tool to summarize the information contained in a large number of variables describing brain activity - as, for example, derived from neuroimaging scans - into a limited number of components that are optimally related to behavioral or diagnostic variables; as such, illnesses or brain states could be predicted from brain activity (Krishnan et al, 2010). Even though the above mentioned studies used continuous outcome variables, and thus adopted regression for their predictive models, the PLS-R approach can easily be extended to a classification situation. To this end, the grouping variables (with K categories) should be converted in a set of $(K - 1)$ dummy variables and these dummy variables should be used as criterion variables. For example, Lehmann et al. (2006) compared PCA with PLS-R in a classification study, in which the goal

was to separate people with Alzheimer from HC's using EEG data. A comparison between the two different methods of dimensionality reduction resulted in a marginal advantage of PLS-R over PCA.

A possible minor drawback of PLS-R is that, while it can reduce bias due to its supervised nature, it has the potential to increase variance (due to overfitting). This is because PLS-R puts a lot of emphasis (perhaps too many) on constructing components that explain the response variable well, and less on explaining the original predictor variables. Some even claim that the overall benefit of PLS-R relative to principal component regression (i.e., performing regression on the PCA component scores), which is an unsupervised method, might turn out to be negligible when used for classification (James et al., 2015).

1.2.4 Principal Covariates Regression (PcovR)

Principal Covariates Regression (PcovR), proposed by De Jong & Kiers (1992), is a dimension reduction technique that, similar to PLS-R, transforms a large set of predictor variables into a smaller set of components, while taking the relationships between these predictors and the response (i.e., grouping) variable(s) into account. When data matrix $\boldsymbol{X}$ contains information for n cases on p predictors and matrix $\boldsymbol{Y}$ contains information for the same n cases on m criteria (response variables; $m = 1$ in our case), PcovR transforms the p predictors into z new variables, named components, such that:

$$\boldsymbol{X} = \boldsymbol{T}\boldsymbol{P}_{\boldsymbol{X}} + \boldsymbol{E}_{\boldsymbol{X}} = \boldsymbol{X}\boldsymbol{W}\boldsymbol{P}_{\boldsymbol{X}} + \boldsymbol{E}_{\boldsymbol{X}}, \tag{1.8}$$

where $\boldsymbol{T}$ is a n x z component score matrix containing scores of the n subjects on the z components, z indicates the number of components, $\boldsymbol{P}_{\boldsymbol{X}}$ is the z x p loading matrix which contains the loadings of the original p predictor variables on the z components, $\boldsymbol{E}_{\boldsymbol{X}}$ (n x p) are the residuals of $\boldsymbol{X}$ and $\boldsymbol{W}$ is a p x z weight matrix. The response matrix $\boldsymbol{Y}$ is then regressed on the component scores $\boldsymbol{T}$ (instead of on the predictors $\boldsymbol{X}$):

$$\boldsymbol{Y} = \boldsymbol{T}\boldsymbol{P}_{\boldsymbol{Y}} + \boldsymbol{E}_{\boldsymbol{Y}} = \boldsymbol{X}\boldsymbol{W}\boldsymbol{P}_{\boldsymbol{Y}} + \boldsymbol{E}_{\boldsymbol{Y}}, \tag{1.9}$$

in which the columns of matrix $\boldsymbol{P}_{\boldsymbol{Y}}$ (z x m) contain the resulting regression weights for each of the m response variables and matrix $\boldsymbol{E}_{\boldsymbol{Y}}$ (n x m) contains the residuals of $\boldsymbol{Y}$. The goal of

PcovR is to find the matrices $\boldsymbol{W}$, $\boldsymbol{P_X}$ and $\boldsymbol{P_Y}$ such that the following loss function is minimized:

$$\begin{aligned} \boldsymbol{L} &= \boldsymbol{a}\frac{\|\boldsymbol{X}-\boldsymbol{T}\boldsymbol{P_X}\|_2^2}{\|\boldsymbol{X}\|_2^2} + (\boldsymbol{1}-\boldsymbol{a})\frac{\|\boldsymbol{Y}-\boldsymbol{T}\boldsymbol{P_Y}\|_2^2}{\|\boldsymbol{Y}\|_2^2} \\ &= \boldsymbol{a}\frac{\|\boldsymbol{X}-\boldsymbol{X}\boldsymbol{W}\boldsymbol{P_X}\|_2^2}{\|\boldsymbol{X}\|_2^2} + (\boldsymbol{1}-\boldsymbol{a})\frac{\|\boldsymbol{Y}-\boldsymbol{X}\boldsymbol{W}\boldsymbol{P_Y}\|_2^2}{\|\boldsymbol{Y}\|_2^2} \end{aligned} \quad (1.10)$$

with $\|\boldsymbol{Z}\|_2$ denoting the Frobenius norm of matrix $\boldsymbol{Z}$ (i.e., the square root of the sum of the squared entries of $\boldsymbol{Z}$). The PcovR algorithm first estimates $\boldsymbol{W}$ by means of SVD. Once $\boldsymbol{W}$ is determined, $\boldsymbol{P_X}$ and $\boldsymbol{P_Y}$ can be calculated by means of multivariate multiple linear regression (ten Berge, 1993; Smilde, Bro, & Geladi, 2004).

The components that PcovR extracts from high-dimensional data are linear combinations of the predictor variables that are constructed in such a fashion that they explain the predictor variables as good as possible (in terms of explained variance), but simultaneously allow for an optimal prediction of the response variable (i.e., R squared), a concept that is similar to that of PLS-R. In contrast to PLS-R, however, PcovR allows the user to choose to what extent both aspects (i.e., good summary of predictors versus optimal prediction of response variable) play a role when constructing the components, by specifying a weighting parameter α (Vervloet et al., 2015). This parameter, which must be a value between zero and one, determines the balance between yielding a good summary of the predictors versus an optimal prediction of the response variable. An α-value of zero indicates that the focus is solely on prediction, while an α-value of one results in an optimal summary of the predictors (which is basically the same as principal component regression). An optimal α-value can be determined through maximum likelihood principles (Vervloet et al., 2015), by means of the following formula:

$$\alpha_{ML} = \frac{\|\boldsymbol{X}\|_2^2}{\|\boldsymbol{X}\|_2^2 + \|\boldsymbol{X}\|_2^2\left(\frac{\sigma_{E_X}^2}{\sigma_{E_Y}^2}\right)}. \quad (1.11)$$

To obtain an optimal α_{ML}-value, the variances $\sigma_{E_X}^2$ and $\sigma_{E_Y}^2$ should be replaced by an appropriate estimate. An estimate for $\sigma_{E_X}^2$ can be calculated by applying PCA to $\boldsymbol{X}$ (the predictor variables) and determining the optimal number of components by, for example,

inspecting a scree plot, with the estimate for $\sigma^2_{\boldsymbol{E_X}}$ taken equal to the associated percentage of unexplained variance. The estimate for $\sigma^2_{\boldsymbol{E_Y}}$ is obtained by taking the percentage of unexplained variance when $\boldsymbol{Y}$ (the response variable) is regressed onto $\boldsymbol{X}$ (for more information, see Vervloet et al., 2015).

Being able to specify α makes PcovR a very flexible and therefore interesting approach. Moreover, PcovR's flexibility forms an advantage over PLS-R, according to Kiers & Smilde (2007). They showed that, for five different simulation settings, there always was at least one so called weighting-scheme of PcovR that outperformed, or at least performed as good as, PLS-R. While the flexibility of PcovR appears to be an advantage when compared to PLS-R, it does also come with the downside of having to optimize more parameters (i.e., optimal value of α), in order to get the best results possible for a particular data set. To the best of our knowledge, PcovR has not yet been used as a tool for feature extraction in a classification context, nor has the method been used in the context of neuroimaging data.

1.3 Research questions and hypotheses

The aim of the current study is to examine whether feature extraction in combination with a SVM classifier can improve classification accuracy in a neuroimaging classification study when compared to using whole-brain data. Although SVM is expected to yield good classification results in the context of high-dimensional data, it is hypothesized that, at least for some neuroimaging properties, feature extraction can improve classification accuracy since it may reduce the noise in the data without discarding potentially useful information, while also taking the multivariate nature of the variables into account. In addition, interest also goes to how the feature extraction techniques perform compared to one another. In this regard, it is hypothesized that PCA will be outperformed by ICA, because, unlike ICA, PCA is not able to extract multi-modal components, which are expected to be more predictive of the groups than unimodal (i.e., Gaussian) components. Further, it is hypothesized that PCA and ICA will perform better when the components are selected based on their relation with the grouping variable (i.e., by using t-tests) compared to when they are selected based on the amount of variance they explained in the original variables. Also, it is hypothesized that due to their supervised nature of constructing the components, PLS-R and PcovR will outperform

the unsupervised techniques PCA and ICA, as well as their semi-supervised t-test counterparts. Finally, it is hypothesized that due to its flexibility in weighting the importance of explaining the predictors and predicting the response, PcovR is able to outperform PLS-R.

In the remainder of this thesis, first, the data and procedure that was is described in Section 2. The results from the analyses as described in the methods section are presented in Section 3. Section 4 summarizes and discusses the results, points to limitations of the study and sketches avenues for further research.

Section 2. Methods

This section describes the data that were analyzed to address the research questions (2.1), the procedure adopted (2.2), the SVM classifier used (2.3) and details about the application of the various feature extraction approaches (2.4).

2.1 Data

In order to examine whether feature extraction improves classification performance when compared to using whole-brain data, data on structural and functional neuroimaging features were collected from 250 participants, of which 77 (30.8%) suffered from AD and 173 (69.2%) were HC's. The AD patients were scanned at the Medical University of Graz as part of a prospective registry on dementia (PRODEM; see also Seiler et al., 2012). Only the patients that were diagnosed with AD according to the NINCDS-ARDRA criteria (McKhann et al., 1984), and for which MRI and fMRI scans were available, were used in this study. Regarding the HC's, image data from the Austrian Stroke Prevention Family Study was used, which is a prospective single-centre community-based follow-up study with the aim of examining the frequency of vascular risk factors and their effects on cerebral morphology and function in the healthy elderly (Schouten et al., 2016).

From the collected neuroimaging data, three properties were selected for the current study: two functional and one structural neuroimaging property. Regarding the structural neuroimaging property, gray matter values were used to distinguish between the two groups. These values indicate the percentage of each voxel that consists of grey matter. Since the loss of grey matter is strongly associated with AD, structural grey matter images are often used in this type of classification studies (Yang et al., 2011; Magnin et al., 2008). The second neuroimaging property is the correlation of each voxels' functional resting state time course with the time course of the so called executive center network within the brain, which is expected to play a role in brain abnormalities in people with AD, although this has not been tested thus far. The third fMRI marker that was used is the amplitude of low-frequency fluctuation (ALFF), which is a resting state fMRI neuroimaging property indicating regional spontaneous low frequency fluctuations in brain activity measured during rest. This

neuroimaging property has proven its use in distinguishing between people with and without mild cognitive impairment (MCI), which often precedes AD (Zhao et al., 2014).

For each of these three neuroimaging properties, data with two voxel sizes were extracted: voxels of size 2mm by 2mm by 2mm and voxels of size 4mm by 4mm by 4mm. Data with 2mm by 2mm by 2mm voxels contain more specific spatial information regarding brain activity, at the cost of also consisting of a lot more variables/voxels, compared to data with 4mm by 4mm by 4mm voxels. Both data versions of each property were used for classification, in order to explore whether voxel size has an influence on classification performance for these three neuroimaging properties. It is expected that the overall classification performance increases when smaller voxels are used because of the extra spatial information present in the data. In total, as can be seen in Table 1 in which the different data sets used are listed, six high-dimensional sets of variables (i.e., 3 properties × 2 voxel sizes) were used as predictor variables to distinguish between people with AD and HC's.

Additionally, since the focus in this study is on very high-dimensional data and some feature extraction methods (i.e., ICA, PLS-R and PcovR) cannot easily deal with such data, a preliminary dimension reduction step (by means of PCA) will be conducted before applying ICA, PLS-R and PcovR (see further). However, the possibility exists that this (negatively) influences the performance of ICA, PLS-R and/or PcovR, since there is no guarantee that PCA is the best preliminary dimension reduction technique preceding the use of these techniques. Therefore, in order to be able to apply ICA, PLS-R and PcovR to the original data, another neuroimaging property with a much smaller number of variables will be analyzed using the same techniques as for the other data sets. In particular, data containing the partial correlations (PC) between the time series of 70 functional brain areas (Schouten et al., 2016) will also be analyzed (see Table 1). When extracting features from this additional data set by means of ICA, PLS-R and PcovR, no preliminary PCA dimension reduction step will be performed; this implies that the feature extraction techniques will be applied to the (unstandardized) original predictor variables instead of to the PCA component scores (see further).

Table 1

Overview of the seven data sets used in this study

Keyword	Description	Voxel size (in mm)	Number of variables
ALFF2	Amplitude of low-frequency fluctuation (ALFF) of each voxel	2 by 2 by 2	190.891
ALFF4	Amplitude of low-frequency fluctuation (ALFF) of each voxel	4 by 4 by 4	25.750
EX2	Correlation of each voxel with executive center	2 by 2 by 2	191.066
EX4	Correlation of each voxel with executive center	4 by 4 by 4	25.759
MR2	Percentage of gray matter for each voxel	2 by 2 by 2	432.031
MR4	Percentage of gray matter of each voxel	4 by 4 by 4	59.049
PC	Partial correlations between time series of 70 functional brain areas	-	2415

2.2 Procedure

There are seven sets of predictor variables, all obtained from the same 250 participants, and one binary outcome variable (i.e., AD vs HC). Using each of these seven sets of predictor variables separately, the goal is to distinguish people with AD from HC's as accurate as possible by means of an SVM classifier, herewith comparing whole-brain analysis to different types of feature extraction. More specifically, on each of the seven sets of predictor variables, six types of feature extraction (with whole-brain being a seventh type) were applied before training the SVM. The first four feature extraction methods are Principal Components Analysis (PCA), Independent Component Analysis (ICA), Partial Least Squares Regression (PLS-R) and Principal Covariates Regression (PcovR). As PCA and ICA, as opposed to PLS-R and PcovR, derive components without taking the relationships between the predictors and the criterion variable into account, an improved PCA and ICA feature extraction method could be obtained by ranking the obtained PCA/ICA components based on their ability to

explain the criterion/grouping variable (i.e., based on the absolute t-value obtained when regressing each component on the grouping variable). As such, T-PCA and T-ICA constitute a fifth and sixth feature extraction approach. For each feature extraction method, the classification performance was evaluated for different numbers of extracted features (i.e., components). In the case of whole-brain analysis, no feature extraction step took place whatsoever, meaning that the original predictor variables were directly fed to the SVM. This results in the following seven (feature extraction) approaches preceding the training of the SVM:

1. PCA
2. T-PCA (PCA components selected using t-tests)
3. ICA
4. T-ICA (ICA components selected using t-tests)
5. PLS-R
6. PcovR
7. Whole-brain analysis (no feature extraction: taking all original predictor variables)

2.2.1 General procedure

The following six-step procedure was executed for each of the feature extraction approaches, on each of the seven datasets. Moreover, this procedure was conducted for a range of values (see further) of the number of components (*S*) that was extracted and used for classification.

- Step 1: Split data into training set (150 subjects) and test set (100 subjects), using the same split for each feature extraction method, number of components *S* and set of predictor variables
- Step 2: Apply feature extraction on the training set and select the first *S* components
- Step 3: Use *S* new variables (scores on the *S* components) from the training set to train the SVM
- Step 4: Derive the component scores of the test data, using exclusively parameters from the training set (obtained in step 2)
- Step 5: Predict class labels for the test set using the component scores of the test set (computed in step 4) and the SVM model from the training set (obtained in step 3)

- Step 6: Compare the predicted class labels for the test set (step 5) with the actual/observed class labels of the test set to compute a measure of classification accuracy.

The procedure for estimating the predicted class label for the test set in the case of whole-brain analysis resembles the six-step procedure described above, with the main difference being that no feature extraction step takes place (i.e., Step 2 and 4 are omitted): the predictor variables are directly used to train the SVM (Step 3), and the SVM model from the training set is applied to the original predictor variables of the test set in order to predict the class labels of the test set (Step 5).

2.2.2 Validation approach

The procedure described in the previous subsection is called the validation approach (James et al., 2015). A disadvantage of the validation approach is that the obtained estimate of classification accuracy can be highly variable as this estimate strongly depends on how the observations are split randomly into a training and a test set. Therefore, in order to get a (more) reliable estimate of classification accuracy, the validation approach is repeated a large number of times and the mean of the estimates across these repetitions is taken as the final estimate of classification accuracy. In this study, the validation process will be repeated 100 times (i.e., 100 different random splits of the data in training and test set). This means that for each of the seven (feature extraction) approaches (for each S-value) on each of the seven sets of predictor variables, 100 estimates of classification accuracy were obtained. To this end, high-performance parallel computing was utilized, in order to deal with the computational intensiveness of this task. A final estimate of classification accuracy per data set and feature extraction approach is obtained by calculating the mean across the 100 obtained estimates of classification accuracy.

2.2.3 Classification accuracy (Step 6)

To compute a measure of classification accuracy (Step 6, see above), Receiving Operating Characteristic (ROC) curves were constructed. A ROC curve illustrates the performance of

a binary classifier when its discrimination threshold is varied. Such a curve is created by plotting the true positive rate (sensitivity) against the false positive rate (specificity) for various threshold values (Hanley & McNeil, 1982). Sensitivity measures the proportion of positives (e.g., people with AD) that are correctly identified as such, whereas specificity measures the proportion of negatives (e.g., HC's) that are correctly identified as such. The area under this ROC curve, referred to as the AUC-value, is equal to the probability that a classifier ranks - in terms of the success probability (e.g., having AD) - a randomly chosen positive instance (e.g., someone with AD) higher than a randomly chosen negative one (e.g., a HC). In other words, the area under the ROC curve represents the probability that a randomly chosen diseased subject is correctly marked with greater suspicion - in terms of the probability of being ill - than a randomly chosen non-diseased subject.

In this study, the AUC-value was used as a measure for classification performance, because it automatically controls for differences in class/group sizes (i.e., number of diseased people and healthy controls). When class sizes are unequal, which is the case in the current study, a model that assigns all cases to the majority class will have a percentage agreement larger than 50% (i.e., about 70% in the current study). By using AUC as a measure of classification accuracy, this unbalanced distribution of the cases across groups is implicitly controlled for. AUC-values were obtained by first using the "roc"-function to construct a ROC plot, followed by the "auc"-function in order to get an estimate of AUC. Both R functions belong to the "AUC"-package (Ballings & van den Poel, 2013).

2.2.4 Number of components *S*

Mean AUC-values were derived for different values of the number of components *S* used for classification, with this value, of course, being irrelevant for the whole-brain analysis. Determining the optimal number of components *S* to be used for classification for each feature extraction approach (and data set) separately is a very computationally intensive effort. Since cross-validation is used (see further) to determine the optimal SVM model, also performing cross-validation to determine the optimal value for *S* would result in a nested cross-validation (i.e., cross-validation within cross-validation), which dramatically increases the computational intensity and duration time of the analyses. Also, by varying *S*, insights about the influence of the number of components that are used for classification on the

classification performance of each approach are obtained. Therefore, the mean AUC-value for each feature extraction approach, on each data set, will be calculated for a range of values of *S*: 1, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120, 130, 140 and S_{max}. Note that as the training set always contains 150 observations, the maximal number of retained components S_{max} cannot be larger than 150. For some feature extraction approaches, however, S_{max} is even a little bit lower (i.e., 148, 149; see further).

2.2.5 Computation time

Even though the focus of this study is on classification performance, the computation times of all analyses were measured. To this end, the elapsed time as indicated by the "system.time"-function from the "base"-package (R Core Team, 2015) was used. The average time it took to perform feature extraction, the time it took to fit the SVM, as well as their combined total time will be presented and compared across methods (see Section 3.3).

2.3 Support Vector Machine (SVM)

The SVM algorithm, proposed by Vapnik (1995), has proven its use in classification studies in which neuroimaging data is used as input (Mourao-Miranda et al., 2005; Kloppel et al., 2008). When confronted with training data on *P* predictor variables of two groups with respective diagnostic labels (AD and HC, for example), the SVM learning process determines a so called (*P*-1)-dimensional hyperplane that optimally separates the training cases into the two labelled groups. The training observations with the smallest distance to the separating hyperplane determine the width of the so called margin; these training observations are called the "support vectors" (James et al., 2015). The aim of the SVM algorithm is to find a maximal margin hyperplane, which is the hyperplane that has the largest distance to the nearest training observation (i.e., the support vectors), and thus the widest margin.

As perfect separation of the cases into the given groups is a rather rare occurrence, some observations may violate the margin. An observation violates the margin when it is on the wrong side of the margin, or even on the wrong side of the hyperplane. The amount of

violations to the margin that is tolerated when fitting a SVM can be varied by means of the cost parameter (*C)*. The value of the cost parameter *C* determines the number and severity of the violations to the margin and hyperplane that will be tolerated by the model. The value for *C* determines how much cost is attached to such a violation (i.e., a penalty indicating how undesirable a violation is). Choosing a small value for *C* allows for a so called soft margin SVM. Embracing a soft margin allows for misclassification errors to be made when fitting the model to the training data. In contrast, utilizing a hard margin (i.e., a high *C*-value) will result in fitting a model that allows no classification errors whatsoever.

2.3.1 Training the SVM (Step 3)

In the case of whole-brain analysis, the input data for the SVM is very high-dimensional, and the number of variables (dimensions) exceeds the number of cases by a mile. In such a high-dimensional space, the margin is not easily violated, and the influence of *C* on the classification performance of the SVM is therefore negligible. Therefore, when using whole-brain data as input, a SVM was fitted with a fixed value of 1 for the cost parameter *C* (and thus no cross-validation was performed to determine an optimal *C*). When using feature extraction, a SVM is fitted to the component scores derived with each of the feature extraction approaches instead of to the original predictor variables. In that case, the cost parameter was tuned by means of 5-fold cross-validation, using the "tune.svm"-function in R. Hsu et al. (2016) found that adopting an exponentially growing sequence of *C*-values yields a good range of parameter values. Often, exponentials of 2 are used (see, for example, Chu et al., 2012). Therefore, in this study, the optimal value for *C* was chosen, by means of cross-validation, out of the following sequence of *C*-values: 2^{-17}, 2^{-15}, 2^{-13}, 2^{-11}, 2^{-9}, 2^{-7}, 2^{-5}, 2^{-3}, 2^{-1}, 2^{1}, 2^{3}.

As can be seen, these selected values for *C* are quite small (i.e., most of them are smaller than 1). These values were used because choosing a range of small values for *C* allows for a soft-margin SVM. As such, a certain amount of misclassification errors is allowed in the training set. This may be useful as allowing misclassifications in the training set may result in a model that generalizes better to novel data (James et al., 2015). In other words, it can prevent the SVM from overfitting the data and therefore may possibly yield more accurate predictions for the test set. In order to make a fair comparison among the feature extraction approaches, the process of fitting a SVM was kept identical for all feature

extraction approaches across all datasets and values of *S*. In particular, a linear kernel SVM was fitted to the training set. To this end, the "svm"-function in the R package "e1071" (Meyer et al., 2015) was used.

2.3.2 Applying the SVM model to the test data (Step 5)

To predict the group labels of the test cases, the SVM model that was trained on the training set (Step 3) was adopted. To this end, the "predict.svm"-function from the "e1071"-package was used. Regarding the feature extraction methods, the component scores of the test data were derived (see further) and the "predict.svm"-function was used in order to predict the class labels for the test set based on these component scores of the test data. This procedure was identical for all feature extraction methods, data sets and *S*-values.

2.4 Analysis specification for the feature extraction approaches (Step 2 and Step 4)

2.4.1 PCA

The key assumption when using PCA for classification purposes is that often a small number of principal components suffices to explain most of the variability in the predictor data, as well as the relationships between the predictors and the response variable. While the second part of this assumption is not guaranteed to be true, it often turns out to be a reasonable assumption that gives good results in classification studies (Mwangi et al., 2014). In this study, PCA was applied to each set of predictor variables using the "prcomp" function belonging to the "stats" package (R Core Team, 2015). Variables that had a variance of zero in the training set were removed from both the training and the test set, as they cannot help in discriminating between groups. For a given neuroimaging property, PCA was performed on the predictor variables of the training set ($\boldsymbol{X_{train}}$), which were first centered (i.e., a mean of zero) and normalized (i.e., a variance of one) based on the variable means and variances in the training set (resulting in $\boldsymbol{X_{train}^{stan}}$). Regarding the PC data, PCA was performed to $\boldsymbol{X_{train}}$ (not standardized).

Performing PCA on the pre-processed training data ($\boldsymbol{X}_{\boldsymbol{train}}^{\boldsymbol{stan}}$) resulted in scores on 150 components; the scores and loadings of this PCA analysis are indicated by $\boldsymbol{A}_{\boldsymbol{train}}^{\boldsymbol{PCA}}$ and $\boldsymbol{B}_{\boldsymbol{train}}^{\boldsymbol{PCA}}$, respectively (i.e., $\boldsymbol{X}_{\boldsymbol{train}}^{\boldsymbol{stan}} = \boldsymbol{A}_{\boldsymbol{train}}^{\boldsymbol{PCA}}\left(\boldsymbol{B}_{\boldsymbol{train}}^{\boldsymbol{PCA}}\right)^{\boldsymbol{T}}$). Note that the maximum number of components that can be extracted with PCA cannot exceed the number of training cases, which is always 150 here, nor the number of variables (>25,000 here). This implies that for PCA, S_{max}= 150. Selecting the most useful *S* components from these 150 components for training the SVM classifier was done in two ways: with and without the use of t-tests. For PCA without the use of t-tests, the natural order of the components that PCA provides, which is based on the amount of explained variance of each component, was used. In particular, the scores corresponding to these *S* components, which can be found in the first *S* columns of $\boldsymbol{A}_{\boldsymbol{train}}^{\boldsymbol{PCA}}$, were fed to the SVM. When the PCA components were selected using t-tests (T-PCA), an independent samples t-test was conducted for each component in the training set separately with group membership as the independent variable. Next, the components were ordered, from largest to smallest, based on their absolute t-values, after which the first *S* components from the training set were selected. The scores associated with the *S* selected components (i.e., the first *S* columns of the ordered $\boldsymbol{A}_{\boldsymbol{train}}^{\boldsymbol{PCA}}$) were used to train the SVM.

After the SVM was fitted to the training data, the component scores for the test data ($\boldsymbol{A}_{\boldsymbol{test}}^{\boldsymbol{PCA}}$) were calculated, herewith following the same steps (and parameters) used to compute the component scores in the training set. To this end, the original variables in the test set ($\boldsymbol{X}_{\boldsymbol{test}}$) were first centered and normalized, herewith using the means and variances of the variables in the training set (resulting in $\boldsymbol{X}_{\boldsymbol{test}}^{\boldsymbol{stan}}$). Component scores of the test data ($\boldsymbol{A}_{\boldsymbol{test}}^{\boldsymbol{PCA}}$) were then derived as:

$$\boldsymbol{A}_{\boldsymbol{test}}^{\boldsymbol{PCA}} = \boldsymbol{X}_{\boldsymbol{test}}^{\boldsymbol{stan}}\boldsymbol{B}_{\boldsymbol{train}}^{\boldsymbol{PCA}}. \quad (2.1)$$

Finally, the first *S* component scores from $\boldsymbol{A}_{\boldsymbol{test}}^{\boldsymbol{PCA}}$ were used to make predictions for the test set. Regarding T-PCA, the component scores of the test set were first ordered, herewith using the t-values from the training set; again, the test set labels were predicted based on the first *S* (ranked) component scores.

2.4.2 ICA

In this study, the FastICA algorithm (Hyvärinen, 1999) was adopted to reduce the number of features by means of ICA. To this end, the "icafast" function from the "ica" package (Helwig, 2015) was used. Since ICA is not able to deal with very high-dimensional data, often a preliminary dimension reduction step is applied before using ICA. For example, Castro et al. (2011), who applied ICA to fMRI data in order to classify schizophrenia patients, used PCA to reduce the dimensionality of their data before applying ICA. This approach was also embraced in the current study (except for the PC data, in which ICA was directly applied to $\boldsymbol{X_{train}}$: $\boldsymbol{X_{train}} = \boldsymbol{S_{train}^{ICA}}\boldsymbol{M_{train}^{ICA}}$). This means that ICA was performed on the 150 PCA components from the training set ($\boldsymbol{A_{train}^{PCA}}$); the source signals and mixing matrix obtained by ICA are indicated as $\boldsymbol{S_{train}^{ICA}}$ and $\boldsymbol{M_{train}^{ICA}}$, respectively (i.e., $\boldsymbol{A_{train}^{PCA}} = \boldsymbol{S_{train}^{ICA}}\boldsymbol{M_{train}^{ICA}}$). The importance of each PCA component (i.e., explained variance) is reflected by its variance. Therefore, in order to retain information about the importance of each PCA component when classifying the training cases, the PCA component scores $\boldsymbol{A_{train}^{PCA}}$ were not standardized before applying ICA to them. Standardizing the data before ICA, which is a common practice, would imply that only the eigenvectors, and not the eigenvalues, would be taken into account when performing ICA, which may result in the loss of information that may be relevant for the classification.

Similar as to with PCA, the selection of *S* ICA component scores ($\boldsymbol{S_{train}^{ICA}}$) was done with (T-ICA) and without (ICA) the use of t-tests. When the first *S* ICA component scores needed to be derived without the use of t-tests, only *S* ICA components were extracted from the PCA component scores (i.e., $\boldsymbol{A_{train}^{PCA}} = \boldsymbol{S_{train}^{ICA}}\boldsymbol{M_{train}^{ICA}} + \boldsymbol{E}$, with the noise $\boldsymbol{E}$ pertaining to the non-extracted components and $\boldsymbol{S_{train}^{ICA}}$ only containing *S* columns), which were then used for classification. In the case of using t-tests, the maximum number of ICA components (S_{max}) was extracted from the PCA component scores of the training set (i.e., $\boldsymbol{A_{train}^{PCA}} = \boldsymbol{S_{train}^{ICA}}\boldsymbol{M_{train}^{ICA}}$, with $\boldsymbol{S_{train}^{ICA}}$ now containing S_{max} columns).[1] Next, the ICA components were sorted based on their absolute t-value obtained with an independent samples t-test with group membership as the dependent variable. Finally, the component scores associated with the first *S* (ranked) ICA components (i.e., the first *S* ranked columns of $\boldsymbol{S_{train}^{ICA}}$) were used to train the SVM classifier.

[1] The maximum amount of useful components that ICA could extract from the 150 PCA component scores was 148, as extracting 149 or 150 components resulted in all ICA components being very similar to each other.

The ICA component scores for the test set were derived as:

$$\boldsymbol{S}_{\boldsymbol{test}}^{\boldsymbol{ICA}} = \boldsymbol{A}_{\boldsymbol{test}}^{\boldsymbol{PCA}}\boldsymbol{W}_{\boldsymbol{train}}^{\boldsymbol{ICA}}, \tag{2.2}$$

where $\boldsymbol{A}_{\boldsymbol{test}}^{\boldsymbol{PCA}}$ are the PCA component scores for the test set (see Section 2.4.1), and $\boldsymbol{W}_{\boldsymbol{train}}^{\boldsymbol{ICA}}$ is the estimated ICA un-mixing matrix from the ICA model fitted to the training set.[2] Note that in the case of the PC data, the ICA component scores for the test set $\boldsymbol{S}_{\boldsymbol{test}}^{\boldsymbol{ICA}}$ were obtained by multiplying the original test data with $\boldsymbol{W}_{\boldsymbol{train}}^{\boldsymbol{ICA}}$ (i.e., $\boldsymbol{S}_{\boldsymbol{test}}^{\boldsymbol{ICA}} = \boldsymbol{X}_{\boldsymbol{test}}\boldsymbol{W}_{\boldsymbol{train}}^{\boldsymbol{ICA}}$). The first *S* ICA components scores for the test data (i.e., the first *S* columns of $\boldsymbol{S}_{\boldsymbol{test}}^{\boldsymbol{ICA}}$) were used to predict the class labels for the test cases, herewith using the SVM model of the training set. In the case of T-ICA, the S_{max} ICA component scores of the test set were ordered using the t-values derived from the training set, after which the first *S* components were selected (i.e., the first *S* columns of the ordered $\boldsymbol{S}_{\boldsymbol{test}}^{\boldsymbol{ICA}}$) for predicting the test labels.

2.4.3 PLS-R

PLS-R aims to find latent variables that capture the information in the predictor variables and simultaneously predict the response variable (Krishnan et al., 2010). In this study, PLS-R was performed using the "plsr" function from the "pls" package (Mevik, Wehrens, & Liland, 2013). In order to reduce the computational effort and to keep the results comparable across the used feature extraction methods, PLS-R was applied to the PCA components scores from the training set ($\boldsymbol{A}_{\boldsymbol{train}}^{\boldsymbol{PCA}}$), except for the PC data where the original unstandardized variables were used ($\boldsymbol{X}_{\boldsymbol{train}}$). In the PLS-R analysis, the grouping variable was used as the response variable ($\boldsymbol{y}_{\boldsymbol{train}}$).

As was the case with ICA, the PCA component scores were not standardized before extracting the PLS-R components. As PLS-R already constructs the components based on their power to predict the class labels in the training set, it is of no use to develop a t-tests-based PLS-R approach to select the *S* most useful PLS-R components ($\boldsymbol{T}_{\boldsymbol{train}}^{\boldsymbol{PLS}}$). Therefore, only *S* PLS-R components were extracted from $\boldsymbol{A}_{\boldsymbol{train}}^{\boldsymbol{PCA}}$ (i.e., $\boldsymbol{y}_{\boldsymbol{train}} = \boldsymbol{A}_{\boldsymbol{train}}^{\boldsymbol{PCA}}\boldsymbol{W}_{\boldsymbol{train}}^{\boldsymbol{PLS}}\boldsymbol{Q}_{\boldsymbol{train}}^{\boldsymbol{PLS}} + \boldsymbol{E}$), or from the original test data $\boldsymbol{X}_{\boldsymbol{train}}$ (for PC data: $\boldsymbol{y}_{\boldsymbol{train}} = \boldsymbol{X}_{\boldsymbol{train}}\boldsymbol{W}_{\boldsymbol{train}}^{\boldsymbol{PLS}}\boldsymbol{Q}_{\boldsymbol{train}}^{\boldsymbol{PLS}} + \boldsymbol{E}$), which were then used as new predictor variables (i.e., $\boldsymbol{T}_{\boldsymbol{train}}^{\boldsymbol{PLS}} = \boldsymbol{A}_{\boldsymbol{train}}^{\boldsymbol{PCA}}\boldsymbol{W}_{\boldsymbol{train}}^{\boldsymbol{PLS}}$ or $\boldsymbol{T}_{\boldsymbol{train}}^{\boldsymbol{PLS}} =$

[2] $\boldsymbol{W}_{\boldsymbol{train}}^{\boldsymbol{ICA}}$ can be computed based on $\boldsymbol{M}_{\boldsymbol{train}}^{\boldsymbol{ICA}}$ and is given in the output of the "icafast" function.

$\boldsymbol{X_{train}W_{train}^{PLS}}$) to train the SVM. Note that PLS-R could only extract 149 components from the PCA component scores.

The PLS-R component scores for the test set ($\boldsymbol{T_{test}^{PLS}}$) were obtained by multiplying $\boldsymbol{A_{test}^{PCA}}$ (or the original test data $\boldsymbol{X_{test}}$ in case of the PC data) with the PLS-R coefficients ($\boldsymbol{W_{train}^{PLS}}$) from the model fitted to the training set. To this end, the "predict"-function was used. Next, the PLS-R components scores for the test set (i.e., the columns of $\boldsymbol{T_{test}^{PLS}}$) were used to predict the test labels, herewith using the SVM model fitted to the training data.

2.4.4 PcovR

Using the "PcovR" package (Vervloet et al., 2015), PcovR was performed on all seven sets of predictor variables with the aim of constructing new features for classification. As with ICA, PcovR encountered difficulties when having to handle very high-dimensional data. As a way out, similar to the approach taken for (T-)ICA and PLS-R, the (not standardized) PCA component scores from the training set ($\boldsymbol{A_{train}^{PCA}}$) were used as variables for the PcovR analysis (i.e., $\boldsymbol{A_{train}^{PCA} = A_{train}^{PCA}W_{train}^{PcovR}P_{X_{train}}^{PcovR} + E_{X_{train}}}$ and $\boldsymbol{y_{train} = A_{train}^{PCA}W_{train}^{PcovR}P_{Y_{train}}^{PcovR} + E_{Y_{train}}}$). Note that for the PC data, PcovR (although with some modifications[3]) was applied to the original variables from the training set $\boldsymbol{X_{train}}$ (i.e., $\boldsymbol{X_{train} = X_{train}W_{train}^{PcovR}P_{X_{train}}^{PcovR} + E_{X_{train}}}$ and $\boldsymbol{y_{train} = X_{train}W_{train}^{PcovR}P_{Y_{train}}^{PcovR} + E_{Y_{train}}}$). The class labels for the training cases ($\boldsymbol{y_{train}}$) were adopted as the dependent variable in the PcovR analysis and the maximum likelihood principle (Vervloet et al., 2015) was used to determine the optimal weight parameter α (see Section 1.2.4).[4] Since PcovR takes the class labels of the training set into account when constructing its components, there is no need for a t-test based approach to select the best *S*

[3] The PcovR algorithm as implemented in the "PcovR"-function of the "PcovR"-package (Vervloet et al., 2015) was not able to analyze the data regarding the partial correlations (i.e., PC data), because the number of variables was too large (note that for the other six neuroimaging properties, PcovR was applied to the PCA component scores, which already implies a serious dimensionality reduction compared to the original data). Therefore, a second (but equivalent) implementation of the PcovR-algorithm was used (see Appendix A for R-code). As this second implementation does not contain the maximum likelihood principle to determine the optimal weight parameter α, for the PC data set, α was fixed to .25, which is a reasonable, but to some extent arbitrary, value.

[4] Note that only the scores of the first 148 PCA components could be used in the PcovR analysis and that the analysis was only able to extract 147 PcovR components (S_{max} = 147 for PcovR).

components regarding PcovR. Thus, *S* components were extracted directly from $\boldsymbol{A}_{\boldsymbol{train}}^{\boldsymbol{PCA}}$ (or $\boldsymbol{X}_{\boldsymbol{train}}$). The resulting component scores were calculated as $\boldsymbol{T}_{\boldsymbol{train}}^{\boldsymbol{PcovR}} = \boldsymbol{A}_{\boldsymbol{train}}^{\boldsymbol{PCA}}\boldsymbol{W}_{\boldsymbol{train}}^{\boldsymbol{PcovR}}$ or $\boldsymbol{T}_{\boldsymbol{train}}^{\boldsymbol{PcovR}} = \boldsymbol{X}_{\boldsymbol{train}}\boldsymbol{W}_{\boldsymbol{train}}^{\boldsymbol{PcovR}}$ (for PC data) and were next utilized to train the SVM classifier.

The PcovR component scores for the test data ($\boldsymbol{T}_{\boldsymbol{test}}^{\boldsymbol{PcovR}}$) were derived as:

$$\boldsymbol{T}_{\boldsymbol{test}}^{\boldsymbol{PcovR}} = \boldsymbol{A}_{\boldsymbol{test}}^{\boldsymbol{PCA}}\boldsymbol{W}_{\boldsymbol{train}}^{\boldsymbol{PcovR}}, \tag{2.3}$$

The PcovR component scores of the test set from the PC data were derived as $\boldsymbol{T}_{\boldsymbol{test}}^{\boldsymbol{PcovR}} = \boldsymbol{X}_{\boldsymbol{test}}\boldsymbol{W}_{\boldsymbol{train}}^{\boldsymbol{PcovR}}$. Finally, scores on the *S* PcovR components of the test data (i.e., the columns of $\boldsymbol{T}_{\boldsymbol{test}}^{\boldsymbol{PcovR}}$) were used to predict the class labels for the test cases.

Section 3. Results

In this section, the results of the classification analyses are presented. First, the classification accuracies for the six neuroimaging data sets and all feature extraction methods are presented (Section 3.1). Next, the results from the analysis of the additional neuroimaging property that has much less features than the other properties are discussed (Section 3.2). Finally, the computation times for the SVM and the feature extraction step are compared (Section 3.3).

3.1 Classification accuracy

In Figure 1, for whole-brain analysis (flat line) and each feature extraction method (i.e., the various curves) separately, the mean AUC-value is plotted against the number of components S for the ALFF data with voxels of size 2mm by 2mm by 2mm (denoted by ALLF2). Note that the mean AUC-value for whole-brain analysis takes the form of a straight line as it does not depend on S. In general, the classification performance of all feature extraction methods increases as S increases, except for PLS-R, which remains stable after retaining S=5 components. It also becomes apparent that for low values of S, the classification performance of most feature extraction methods is very disappointing (i.e., not much larger than at chance level). As this pattern of low mean AUC-values for low values of S, that is encountered for most feature extraction methods across all neuroimaging properties considered, is not very relevant for the purpose of this study, in the remainder of this section only the relevant upper part of the plots will be shown (full plots are presented in Appendix B). As such, the relevant information (i.e., which method is performing best) is emphasized more.

A similar plot as in Figure 1 is presented for each neuroimaging property and each voxel size (2mm by 2mm by 2mm in left panels and 4mm by 4mm by 4mm in right panels) separately in Figure 2 (ALLF property), Figure 3 (EX property) and Figure 4 (MR property). In Table 2, for each data set separately, the largest mean AUC-value, encountered across all considered values of S, is presented for the various feature extraction methods. The value of S that belongs to the largest mean AUC-value varies across feature extraction methods as well as data sets. From this table, it can be seen that the difference in maximum classification performance (across all S -values) between whole-brain analysis and each feature extraction approach is very small. For example, the difference between the best performing technique (PLS-R) and whole-brain analysis for the ALFF4 dataset is .018 (i.e., .839 – .821). However,

this difference in classification performance between PLS-R (M=.839, SD=.036) and whole-brain analysis (M=.821, SD=.035) is significant ($t(99) = 10.186$, $p < .0001$).

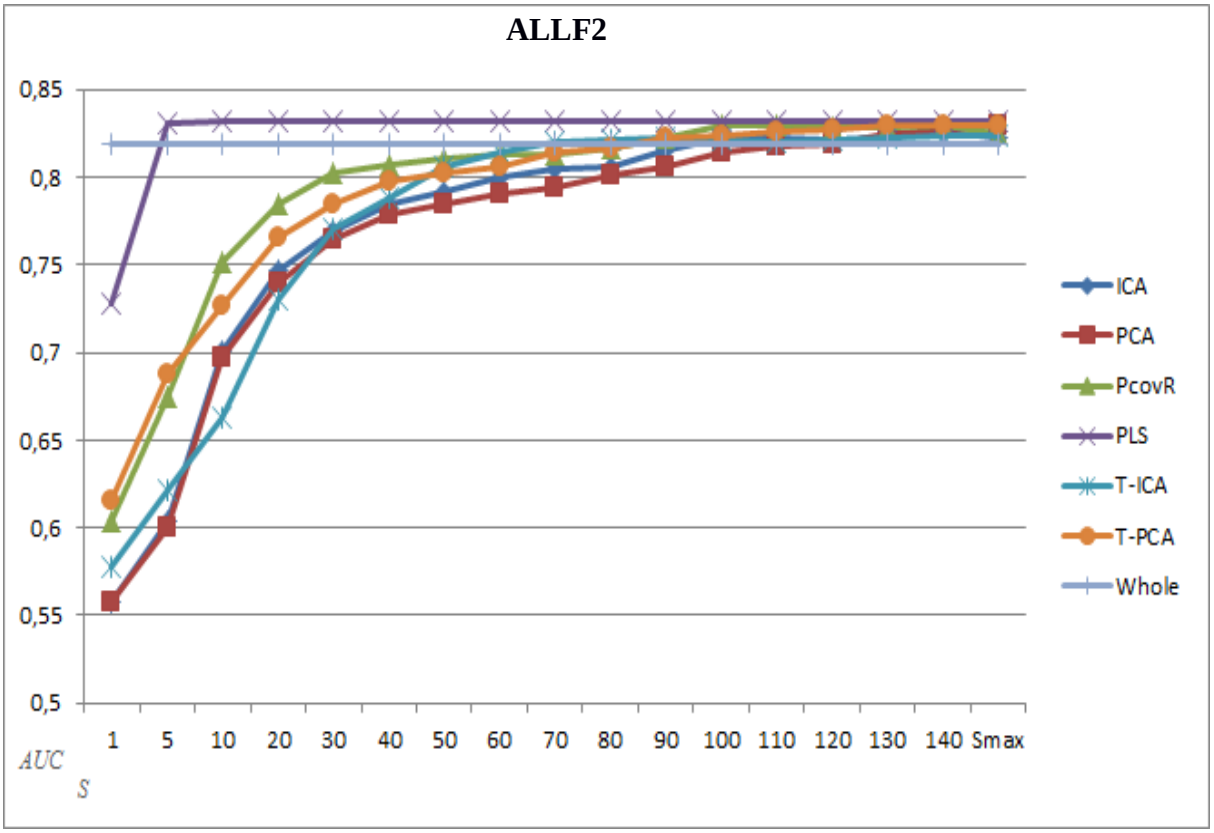

Figure 1. Mean AUC-values plotted against the number of components *S* for the ALFF2 data with voxels of size 2mm by 2mm by 2mm. The various curves represent the results for whole-brain analysis (flat grey line) and the six feature extraction methods (PCA, T-PCA, ICA, T-ICA, PLS-R and PcovR).

3.1.1 Voxel size

When comparing, as illustrated in Table 2, the data sets with smaller (2mm by 2mm by 2mm) voxels to the data sets with larger (4mm by 4mm by 4mm) voxels, only small differences in classification performance are encountered. Regarding the ALFF property, the maximal mean AUC-values for data with larger voxels (ALFF4) are somewhat larger than those for data with smaller voxels (ALFF2). The opposite is true for the EX data sets, in which the maximal mean AUC-values for EX2 are slightly larger than those for EX4 data. For the structural MR property, the data for both voxel sizes provide similar maximum mean AUC-values.

Table 2

Largest mean AUC-value for each feature selection approach (rows), encountered across all values of S, for each data set (columns). For the solution with the optimal S, the standard deviations of the AUC-values (across random splits in the validation approach) are presented between parentheses

	ALFF2	**ALFF4**	**EX2**	**EX4**	**MR2**	**MR4**	**PC**
PCA	.830	.838	.765	.758	.892	.899	.775
	(.039)	(.037)	(.034)	(.036)	(.027)	(.028)	(.052)
T-PCA	.830	.837	.767	.765	.892	.898	.750
	(.039)	(.036)	(.036)	(.036)	(.029)	(.026)	(.055)
ICA	.826	.827	.757	.752	.888	.889	.814
	(.042)	(.040)	(.039)	(.037)	(.033)	(.030)	(.037)
T-ICA	.824	.831	.756	.750	.881	.880	.787
	(.043)	(.043)	(.040)	(.036)	(.032)	(.033)	(.038)
PLS-R	**.831**	**.839**	.764	.760	.890	.898	**.816**
	(.038)	(.036)	(.035)	(.036)	(.027)	(.029)	(.035)
PcovR	.829	.833	.757	.758	.890	.890	.802
	(.035)	(.043)	(.038)	(.039)	(.033)	(.028)	(.035)
WB	.819	.821	**.771**	**.766**	**.913**	**.911**	**.816**
	(.035)	(.035)	(.034)	(.034)	(.022)	(.022)	(.034)

Note. Per data set, the mean AUC-value of the best performing approach is indicated in bold. WB = Whole-brain.

3.1.2 Whole-brain analysis vs. feature extraction

Regarding the ALFF property (Figure 2), all feature extraction approaches performed better in terms of mean AUC than whole-brain analysis, under the condition that S is large enough (i.e., $S > 100$). For lower values of S (i.e., $S < 50$), only PLS-R outperforms whole-brain

analysis in terms of classification performance, except for when *S=1*. Although all these differences are rather modest in size, the maximum mean AUC-value of each feature extraction approach is larger than the mean AUC-value of whole-brain analysis. As was shown to be the case for the ALFF4 data, the difference in AUC-value between PLS-R (M=.831, SD=.038) and whole-brain analysis (M=.819, SD=.035) regarding the ALFF2 data is significant ($t(99) = 7.025$, $p < .00001$) as well. As a consequence, feature extraction apparently can increase classification performance compared to whole-brain analysis in the case of the ALFF data.

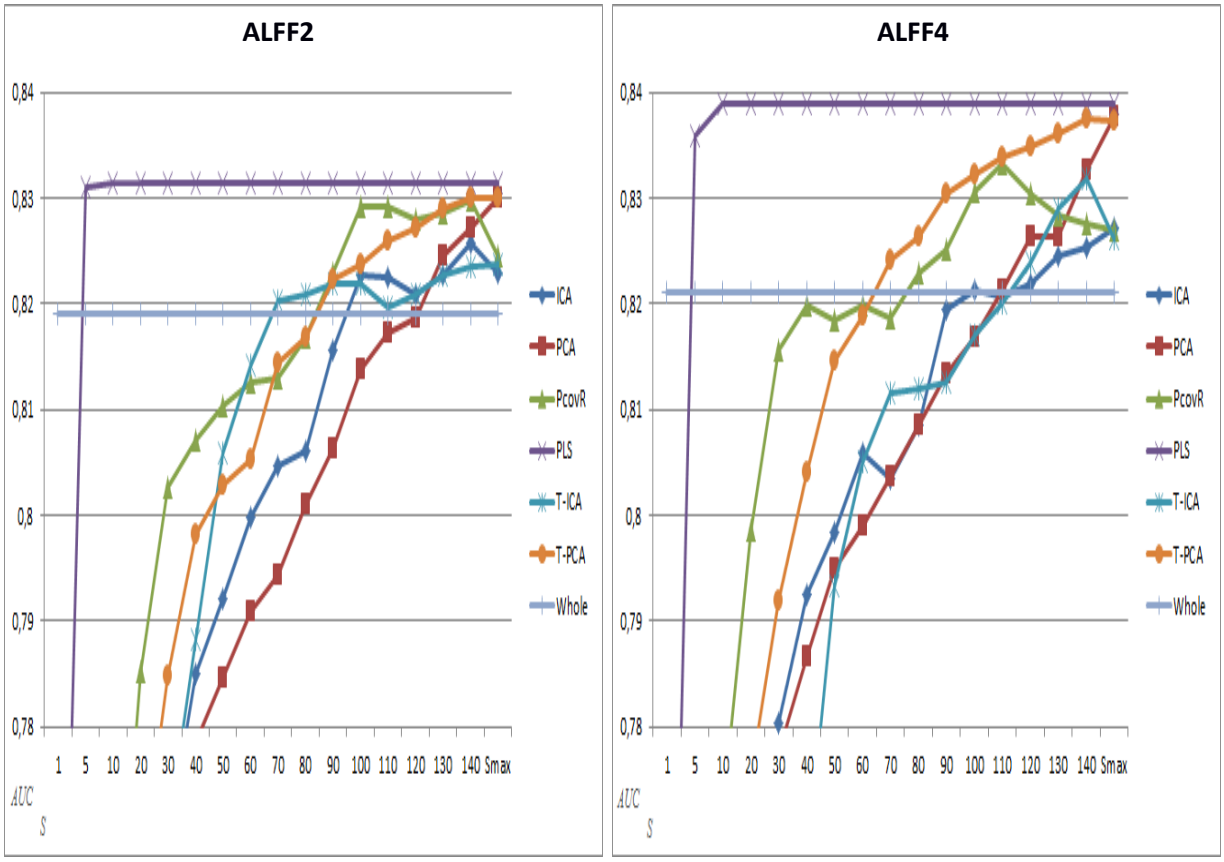

Figure 2. Mean AUC-values plotted against the number of components *S* for the ALFF data with voxels of size 2mm by 2mm by 2mm (left panel) and 4mm by 4mm by 4mm (right panel). The various curves represent the results for whole-brain analysis (flat grey line) and six feature extraction methods (PCA, T-PCA, ICA, T-ICA, PLS-R and PcovR).

In contrast, for both EX data sets (see Figure 3), whole-brain analysis slightly performs better than each of the feature extraction approaches, of which PLS-R (when *S* is low) and T-PCA (when *S* is large) come closest to the classification performance of whole-brain analysis. Regarding the MR data (see Figure 4), whole-brain analysis outperforms each of the feature

extraction approaches in terms of classification performance, with excellent mean AUC-values of .913 (MR2) and .911 (MR4). Again, PLS-R (for low *S*) and (T-)PCA (for large *S*) approach the classification performance of whole-brain analysis the closest. Yet, also here, the difference in classification performance between whole-brain analysis and the best performing feature extraction technique(s) appears to be rather small. For example, in the case of the MR2 dataset, whole-brain analysis only slightly performs better than T-PCA, which is the best performing feature extraction approach (i.e., a mean AUC-value of .913 versus .892). However, this difference in classification performance between T-PCA (M=.892, SD=.029) and whole-brain analysis (M=.913, SD=.022) is significant ($t(99) = 8.452$, $p < .0001$). The same can be said about the MR4 data, for which whole-brain analysis (M=.911, SD=.022) also significantly outperforms the best performing feature extraction technique, which is PCA (M=.899, SD=.028), in terms of classification accuracy ($t(99) = 4.144$, $p < .0001$). Overall, for the structural MR data, whole-brain analysis significantly outperforms the feature extraction methods.

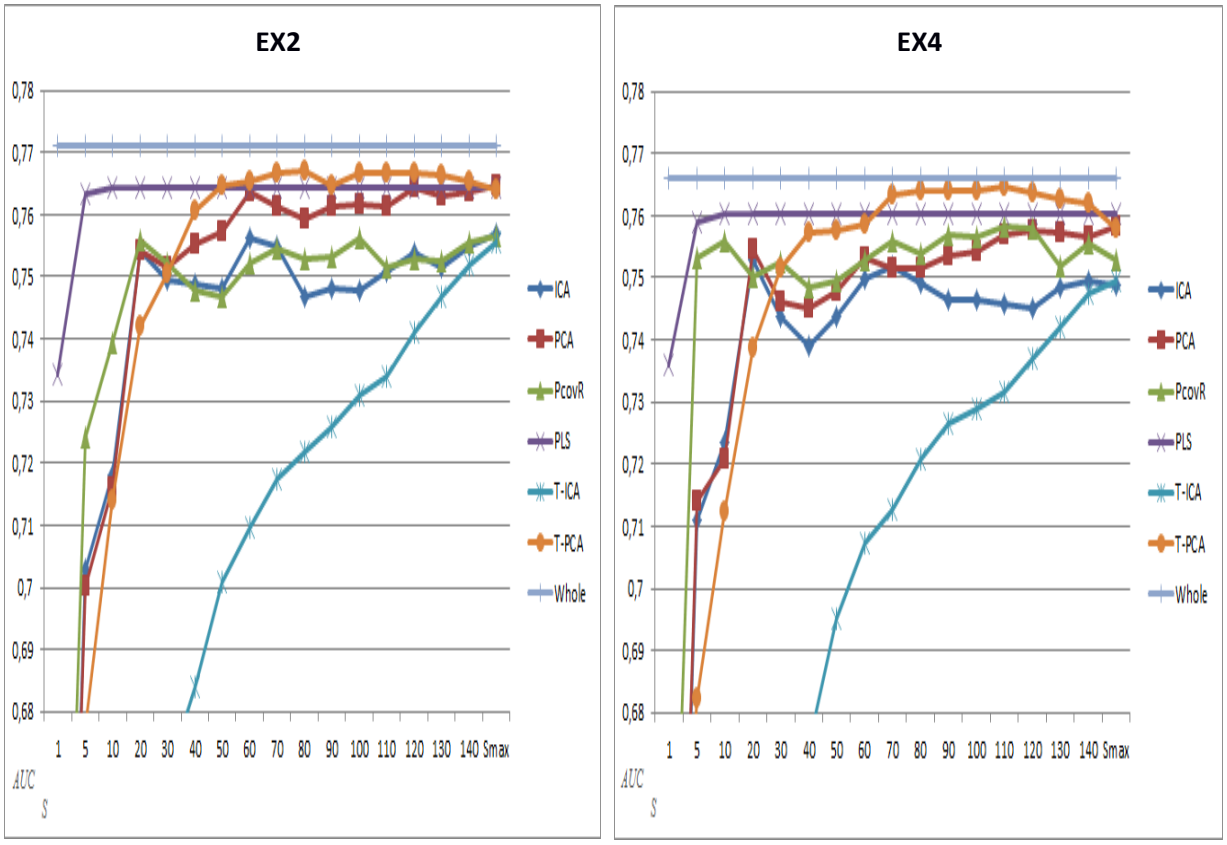

Figure 3. Mean AUC-values plotted against the number of components *S* for the EX data with voxels of size 2mm by 2mm by 2mm (left panel) and 4mm by 4mm by 4mm (right panel). The various curves represent the results for whole-brain analysis (flat grey line) and six feature extraction methods (PCA, T-PCA, ICA, T-ICA, PLS-R and PcovR).

In a nutshell, overall, feature extraction significantly improves classification performance for both ALFF data sets compared to whole-brain analysis, but fails to do so for the EX and MR properties.

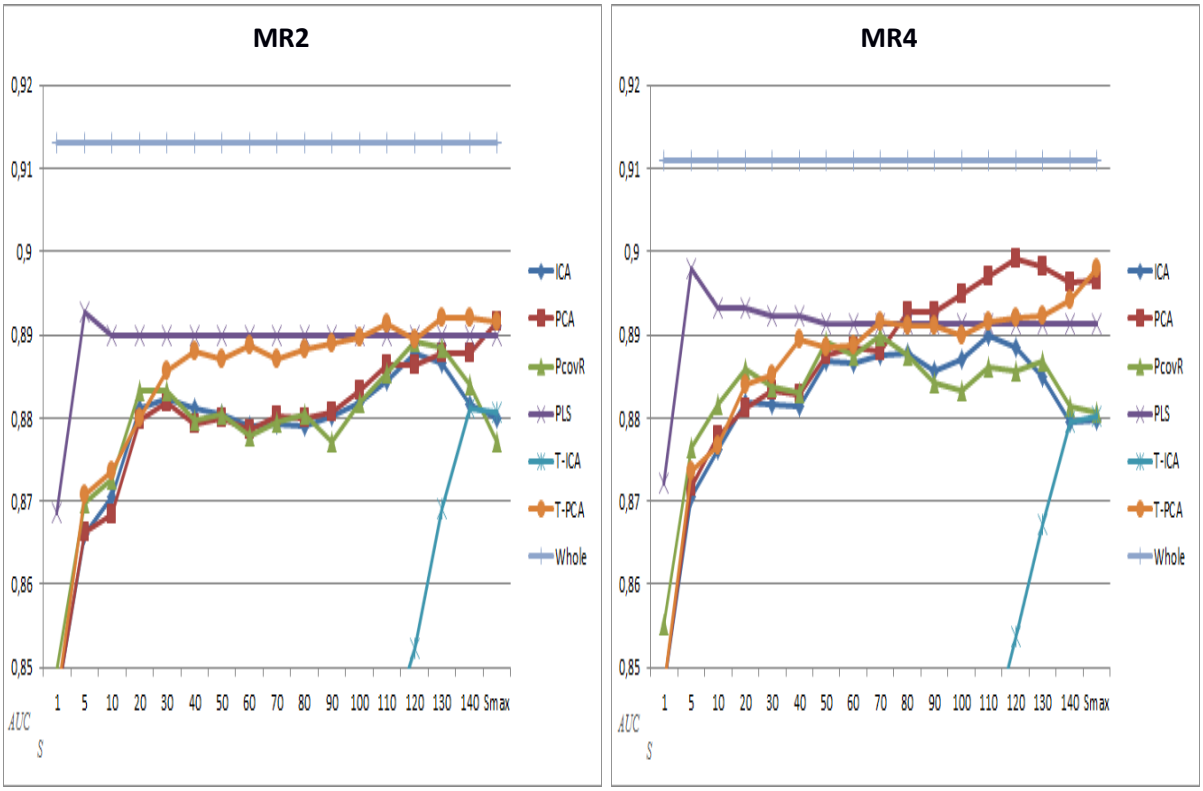

Figure 4. Mean AUC-values plotted against the number of components S for the MR data with voxels of size 2mm by 2mm by 2mm (left panel) and 4mm by 4mm by 4mm (right panel). The various curves represent the results for whole-brain analysis (flat grey line) and six feature extraction methods (PCA, T-PCA, ICA, T-ICA, PLS-R and PcovR).

3.1.3 PCA vs. ICA

As can be seen in Figures 3 and 4, for both the EX and MR data sets, (T-)ICA does not outperform both PCA and T-PCA in terms of classification accuracy. In particular, for low values of S, their classification performances are approximately equal, but the mean AUC-value of (T-)PCA exceeds that of (T-)ICA as S increases all the way up to S_{max}. Regarding the ALFF data sets (Figure 2), ICA does occasionally slightly outperforms PCA (but not T-PCA) when S is between 50 and 100. However, for both ALFF data sets, the maximum AUC-value of (T-)PCA is larger than the corresponding value for (T-)ICA. Overall, the maximum

classification performance of (T-)PCA is higher than that of (T-)ICA for all six data sets (see also Table 2). In general, the performance of (T-)PCA keeps increasing as S increases, sometimes being optimal at S_{max} only, while the performance of (T-)ICA usually flattens out, or even degrades, after a certain value of *S* (see, for example, the right panel of Figures 3 and 4).

3.1.4 With vs. without t-tests

With the exception of MR4 (right panel of Figure 4), the T-PCA approach outperforms ordinary PCA for most values of S for each dataset (Figures 2, 3 and left panel of Figure 4). Apparently, using the PCA components most related to the grouping (in the training set) leads to better classification performances than using the PCA components that explain the most variance of the predictors (in the training set). For example, in Figure 2, one can see that the performance of T-PCA (orange line) exceeds the performance of PCA (red line) across the whole range of S-values. At S_{max}, however, the difference in performance between PCA and T-PCA disappears, because in that case both approaches make use of the same 150 components.

In contrast to PCA, using independent t-tests in a similar fashion to select the most useful ICA components does not seem to improve the classification performance of ICA. Figures 3 (EX) and 4 (MR) illustrate that, for lower values of S, the mean AUC-value of T-ICA is inferior to the mean AUC-value of the other techniques, including ordinary ICA; at best, T-ICA catches up with ordinary ICA in terms of classification performance only for the highest level(s) of S. Regarding the ALFF data sets (Figure 2), T-ICA performs somewhat similarly to ordinary ICA, although its maximum mean AUC-value is still slightly lower than that of ICA. In general, compared to ICA, the usage of t-tests to select the most useful ICA components does not seem to yield an increase in classification performance. However, as is the case for (T-)PCA, at S_{max} , ICA and T-ICA perform equally well, since in that situation both approaches use the same components as features for the classification.

3.1.5 Supervised vs. Unsupervised

As can be seen in Table 2, for both ALFF data sets, the supervised PLS-R technique has the best classification performance. Moreover, for the other data sets (Figures 3 and 4), PLS-R has the highest mean AUC-values amongst all feature extraction techniques when S is small. Remarkably, PLS-R is the only technique that reaches its maximum mean AUC-value already for small S values (i.e., S=5 or S=10). Moreover, for each data set, PLS-R reaches its peak in terms of mean AUC-value for a low S value and flattens out, or even performs somewhat worse (Figure 4), after that.

For each of the six data sets, PLS-R clearly outperforms (T-)ICA. While PcovR does reach higher maximum mean AUC-values than (T-)ICA, its maximum mean AUC-value is lower than that of PLS-R for each data set (see Table 2). In contrast to (T-)ICA and PcovR, the maximum mean AUC-value of (T-)PCA almost equals the maximum mean AUC-value of PLS-R when S is large enough. In general, it appears that PLS-R and (T-)PCA, although peaking at different S values, perform somewhat better than (T-)ICA and PcovR. As a consequence, it cannot be stated that, in general, supervised techniques (e.g., PLS-R and PcovR) necessarily outperform unsupervised techniques (e.g., PCA and ICA). Combining the unsupervised techniques with t-test, which makes these techniques "semi-supervised", however, sometimes leads to improved classification performances (e.g., T-PCA often outperforming PCA).

3.1.6 PLS-R vs. PcovR

For each data set, PLS-R slightly outperforms PcovR. These differences, however, are relatively small, and sometimes almost absent. For example, in the case of ALFF2, the maximum mean AUC-value of PLS-R (M=.831, SD=.038) is nearly equal to that of PcovR (M=.829, SD=.035), and the difference between them is not significant ($t(93) = 1.289$, $p = .201$). However, the fact remains that PLS-R needs far less components to reach its maximum classification performance than PcovR (and the unsupervised feature extraction techniques). Moreover, regarding the MR4 data for example, the difference in AUC between PLS-R (M=.898, SD=.029) and PcovR (M=.890, SD=.028) is significant ($t(94) = 2.918$, $p = .004$). As

such, PLS-R leads to simpler models with less predictor variables, something that speaks in advantage of PLS-R, and is able to significantly outperform PcovR in terms of classification performance, although not for each data set. Also, there is no value of S for any of the data sets at which PcovR outperforms PLS-R in terms of the mean AUC-value.

3.2 Classification accuracy for the Partial Correlations (PC) data

In Figure 5, the mean AUC-values for the various feature extraction techniques and S values are presented for the partial correlations (PC) data, a data set that contains a much smaller number of features that the other data sets in this study. From this figure, it can be seen that none of the feature extraction approaches performs better than whole-data analysis in terms of classification accuracy. However, as can be seen in Table 2, the maximum mean AUC-value of PLS-R equals that of whole-data analysis (i.e., both being .816), closely followed by the mean AUC-value of ICA (.800) and PcovR (.802). The smallest maximum mean AUC-values are obtained for PCA (.775) and T-PCA (.750).

In contrast to the other (more high-dimensional) data sets, T-PCA does not improve the classification performance when compared to PCA. Moreover, when compared to ICA, T-ICA is also not beneficial here, as was the case for the other data sets. As opposed to the larger data sets in this study, for the PC data, ICA (whole S -range), T-ICA (for large S), PLS-R (for small S) and PcovR (whole S -range) all outperform (T-)PCA. A possible reason for this may be that for the PC data, (T-)ICA, PLS-R and PcovR were applied directly to the data instead of to the PCA component scores derived from the data. However, even though PLS-R and ICA have a similar level of classification performance as whole-data analysis, none of these methods is able to exceed the whole-data classification performance. It is remarkable that the performance of PLS-R, which is almost at the level of whole-brain analysis for small S, becomes dramatic for $S > 50$, whereas the performance of PcovR, which is close to the performance of PLS-R (for small S) and whole-data analysis, is almost constant across S. Finally, as opposed to observed for the other data sets, for most feature extraction techniques (except for T-ICA) the classification performance does not increase with S (and even seems to decrease a little).

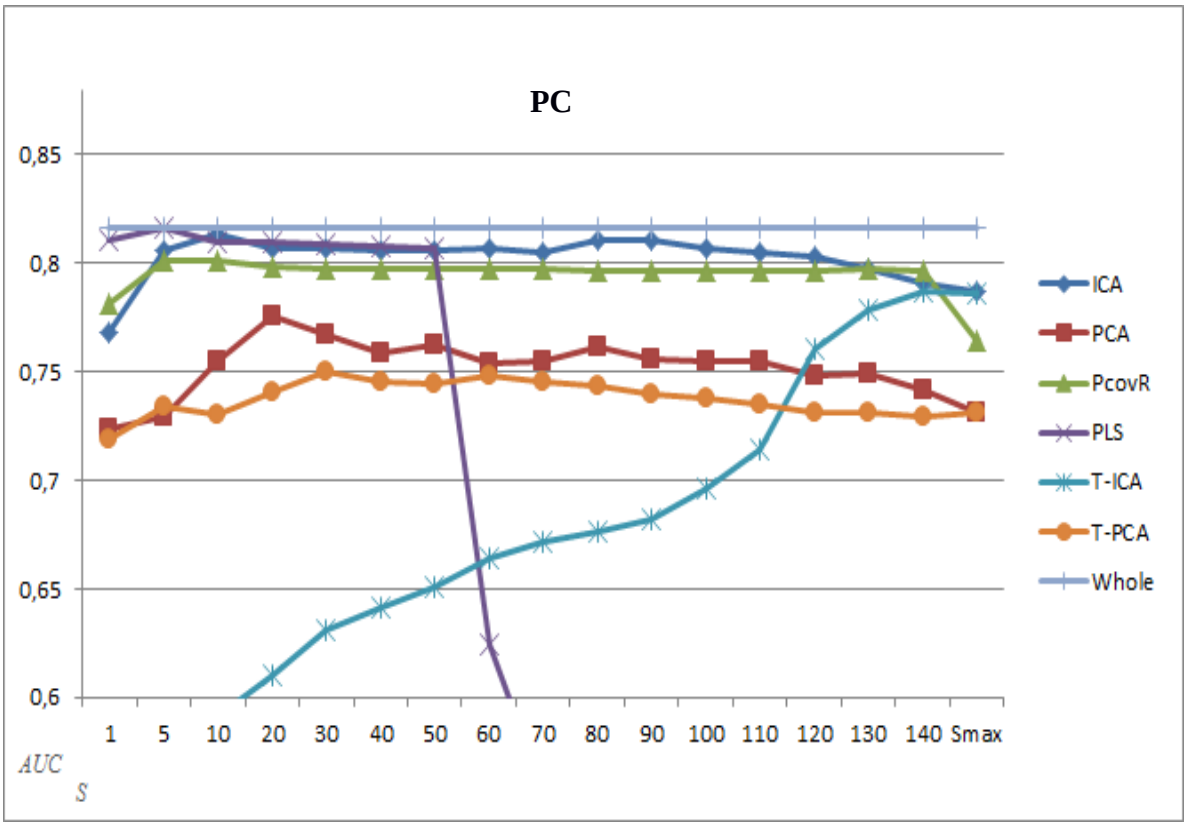

Figure 5. Mean AUC-values plotted against the number of components S for the Partial Correlation (PC) data. The various curves represent the results for whole-brain analysis (flat grey line) and six feature extraction methods (PCA, T-PCA, ICA, T-ICA, PLS-R and PcovR).

3.3 Computation time

In Table 3, the computation time (in seconds) of each feature extraction approach and whole-brain analysis is presented for the ALFF4, EX2 and MR2 data sets, which contain about 25.000, 190.000 and 490.000 features, respectively (see Appendix C for the computation times for the other data sets). Since the optimal number S for each feature extraction approach is not known beforehand, the computation times for S_{max} are displayed. The computation time is presented for the feature extraction and the SVM fitting step separately, as well as for both steps together. Note that for the feature extraction approaches, the optimal cost parameter C for the SVM was selected out of a set of eleven C-values by means of five-fold cross-validation (i.e., performing the SVM $5 \times 11 = 55$ times on $^4/_5$-th of the data); for whole-brain analysis, C was fixed (i.e., performing the SVM only a single time) as the classification performance of SVM is insensitive to the choice of C when the data are very high-dimensional (See Section 2.3.1). As a consequence, for the feature extraction approaches, the

computation times for running the SVM a single time on the reduced features are about a factor of 50 smaller than the computation times presented in Table 3.

From Table 3 it becomes clear that whole-brain analysis has the shortest total computation time for each of the three data sets. The first reason for this is that whole-brain analysis does not imply a potentially time-demanding feature extraction step, which may especially become time-consuming when the data contain many features (e.g., PCA lasts about twenty times longer when comparing ALFF4, with 25.000 features, to MR2 with 490.000 features). Note that for most data sets, especially for the larger ones, the PCA reduction step alone takes longer than the SVM step for the whole-brain analysis. The second reason is that no cross-validation is used when training the SVM for whole-brain data, whereas for the feature extraction methods a computational intensive five-fold cross-validation approach is performed to determine the optimal value for the cost parameter C (see Section 2.3.1). As can be seen in Table 3, the computation time of the SVM fitting step for whole-brain analysis does increase when the number of voxels increases, and clearly exceeds the computation of the SVM step for the feature extraction methods on both the (larger) EX2 and MR2 data sets. However, since whole-brain analysis does not contain a possibly time-demanding feature extraction step, the total computation time for whole-brain analysis is still lower than the computation time for the feature extraction approaches.

Table 3 gives a somewhat distorted image when one wants to study the differences in computation time between the various feature extraction approaches. The reason for this is that all feature extraction techniques, except PCA, were applied to the PCA component scores instead of to the (much larger) original data. In particular, PCA extracted components from thousands of voxels, while the other techniques only extracted S components from 150 PCA component scores. In Table 3, the computation time of the preliminary PCA-step was added to the computation time of each individual feature extraction method. As a consequence, comparing computation times between feature extraction methods can give a distorted image. Table 3, therefore, mainly illustrates the difference in analysis time between whole-brain analysis and the various feature extraction methods as a whole.

Table 3

Computation time (in seconds) of performing feature extraction (first column), fitting the SVM (second column) and total time (third column) for the various feature extraction methods using S_{max} (rows) and data sets (blocks of three columns)

	ALFF4 (25.000 features)			**EX2 (190.000 features)**			**MR2 (490.000 features)**		
Technique	**FE**	**SVM***	**Total**	**FE**	**SVM***	**Total**	**FE**	**SVM***	**Total**
PCA	7.1	14.16	21.26	50.94	13.76	64.7	140.26	14.13	154.39
T-PCA	7.26	14.22	21.48	51.08	13.65	64.73	140.42	13.23	153.65
ICA	9.24	14.19	23.43	53.16	13.71	66.87	142.41	13.51	155.92
T-ICA	9.29	14.12	23.41	53.38	13.92	67.3	142.57	13.68	156.25
PLS-R	7.49	15.04	22.53	51.13	13.95	65.08	140.46	14.29	154.75
PcovR	12.59	14.04	26.63	54.56	13.46	69.02	144.49	13.42	157.91
WB	-	9.41	**9.41**	-	45.49	**45.49**	-	112.52	**112.52**

Note. WB = whole brain, FE = feature extraction method, SVM = support vector machine. *SVM is performed with five-fold cross-validation and eleven C-values for the feature extraction methods and without cross-validation/different C-values for whole-brain analysis.

In Table 4, the computation times for the PC data are presented, where all feature extraction techniques were applied directly to the original data (i.e., without preliminary PCA-feature reduction), enabling a fair comparison between the feature extraction approaches. In this table, the computation time for the feature extraction and SVM fitting step are presented separately as well as combined and this for three selected values of S: 1, 10 and 100.

Regarding the different feature extraction approaches, it can be seen that, across all values of S, PCA (and therefore also T-PCA) and PLS-R are by far faster techniques than (T-)ICA and PcovR. Moreover, as opposed to (T-)PCA and PLS-R, the computation time for (T-)ICA and PcovR did not seem to increase that much (if at all) as S increased, meaning that extracting more ICA or PcovR components from the original variables did not increase (much) the computation time. As can be expected, for all techniques, the computation time for fitting the SVM model increased as S increased. Also, since the PC data are relatively small (i.e., only 2415 variables) compared to the other data in this study (i.e., > 25.000), the total

computation time of whole-brain analysis is very small (.71 seconds), which makes whole-brain analysis one of the fastest approaches for the PC data.

Table 4

Computation time (in seconds) for the PC data of performing feature extraction (first column), fitting the SVM (second column) and total time (third column) for the various feature extraction methods (rows) and for three levels of S (blocks of three columns)

	$S = 1$			$S = 10$			$S = 100$		
Technique	**FE**	**SVM***	**Total**	**FE**	**SVM***	**Total**	**FE**	**SVM***	**Total**
PCA	.42	.69	1.08	.42	2.17	2.61	.42	9.06	9.48
T-PCA	.56	.64	1.1	.56	1.86	2.42	.56	9.24	9.8
ICA	31.15	.58	31.73	31.42	1.5s	32.92	31.56	9.41	41.07
T-ICA	35.02	.59	35.61	35.02	1.42	36.44	35.02	9.23	44.25
PLS-R	.08	.58	.66	.09	1.37	1.46	1.43	9.89	11.32
PcovR	37.69	.56	38.24	37.46	1.38	38.84	37.55	9.36	46.91
WB	-	.71	.71	-	.71	.71	-	.71	.71

Note. WB = whole brain, FE = feature extraction, SVM = support vector machine. *SVM is performed with five-fold cross-validation and eleven C-values for the feature extraction methods and without cross-validation/different C-values for whole-brain analysis.

Section 4. Discussion

In this section the results of the study are summarized and discussed. First, the findings regarding the main research question, whether feature extraction can increase classification accuracy compared to whole-brain analysis, are summarized (Section 4.1). In Section 4.2, the most important conclusions about the comparison of the various feature extraction approaches are discussed. Next, some limitations of the study are provided (Section 4.3). Finally, recommendations for future research within this field are sketched (Section 4.4).

4.1 Can feature extraction increase classification accuracy?

Chu et al. (2012) argued that feature selection (i.e., selecting a subset of the original variables) combined with a SVM classifier does not improve classification accuracy compared to using whole-brain analysis, unless these features are selected based on a priori information regarding the importance of the features. However, as opposed to feature extraction (i.e., using linear combinations –weighted sums– of the variables/voxels), feature selection does not take the relationships between the features/voxels into account. Therefore, it was hypothesized that using feature extraction, as opposed to feature selection, before fitting a SVM would improve classification accuracy when the aim is to, for example, distinguish between people with AD and HC's.

The results showed that feature extraction can increase classification accuracy compared to using whole-brain data as input for a SVM classifier. In particular, for the ALFF data, feature extraction resulted in better classification performances than whole-brain analysis and this especially when adopting PLS-R and (T-)PCA. However, a similar increase in classification performance was not observed for the other neuroimaging properties. In particular, for the EX and structural MR data, whole-brain analysis performed slightly better than all feature extraction approaches; also here, the differences in performance were rather small. The overall superior performance of whole-brain analysis in this study may be explained by the fact that SVM easily yields excellent performance results when applied to (very) high-dimensional data, leaving less room for feature extraction approaches to beat whole-brain analysis (see Kloppel et al., 2008; Magnin et al., 2008).

Regarding computation time, fitting the SVM a single time takes much longer when whole-brain data are used as input data compared to when only a subset of variables or

extracted components are used. However, in the latter case, the optimal cost parameter for the SVM needs to be determined, which is often done by means of a time-consuming procedure, like cross-validation. As a consequence, which is also the case in the current study, fitting the SVM to whole-brain data is faster than the SVM fitting of the extracted features. Moreover, for very high-dimensional data, the feature extraction step itself can also get time-consuming and this for two reasons. First, dimension reduction of such data can be computationally intensive. Second, the optimal number of features the data is reduced to needs to be determined by means of a time-consuming procedure (e.g., cross-validation). In sum, the total computation times for the feature extraction methods ended up exceeding those for whole-brain analysis.

Besides having the highest classification performance on most data sets and having the shortest computation times, whole-brain analysis is also easier to apply than any of the feature extraction approaches, for which a lot of non-trivial choices have to be made. Example are: whether or not to standardize the input variables before feature extraction, determining the optimal number of components to extract, whether or not to use t-tests to rank the retrieved components in terms of importance and selecting the right value for the weight parameter in the case of PcovR. In other words, with a lot of choices that needs to be made, a lot can go wrong when first using feature extraction before training a SVM, compared to when directly using whole-brain data as input data. In that regard, feature extraction can be considered a learning step, along with comes the risk that the training data is overfit.

4.2 Comparison among the feature extraction methods

When the number of extracted components was chosen large enough, performing ICA on the PCA component scores (with or without the use of t-tests) did not improve the classification performance when compared to using the PCA components itself as extracted features for the SVM. Apparently, ICA is not able to extract components that contain more classification-related information than the original PCA components. However, ICA outperformed PCA when it was applied directly to the less high-dimensional PC data set. This suggests that applying ICA to the original variables may result in components with better predictive qualities than the PCA components. Regarding computation time, however, PCA is way faster than ICA.

Using independent t-tests to select the best components in terms of predictive ability is shown to be beneficial for PCA (as was hypothesized), but not for ICA (as opposed to what was hypothesized). For PCA this implies that selecting a subset of components based on their relationship with the response variable is a better method in terms of classification performances than selecting the PCA components that explain the most amount of variance in the original predictor variables. For most neuroimaging properties, however, PCA reached its maximum classification performance when all of its components were used for classification, making the selection of components using t-tests superfluous. For ICA, in contrast to for PCA, the classification performance did not increase when the best ICA components were selected with t-tests from a larger set of extracted ICA components compared to directly extracting the required number of ICA components. It appears that ICA spreads out the classification information across all of its extracted components, implying that the predictive quality of each ICA component becomes better when less components are extracted.

Regarding the supervised feature extraction techniques, for all neuroimaging properties, PLS-R was amongst the best performing techniques, both in terms of classification performance as well as in speed. Further, although PLS-R performed at the level of PCA and/or whole-brain analysis, PLS-R is the only method that performs well when only a limited number of components are extracted. This clearly contrasts with PCA, which often needs all of its components to obtain optimal performance, and with whole-brain analysis, which uses all original variables by definition. As using fewer components to train a SVM speeds up the testing process, and the principle of Occam's razor states that, under equal conditions, predictive models with less variables should be preferred over models using more variables, PLS-R can be considered as the most effective and efficient feature extraction method in this study.

Although PLS-R is a supervised method, the hypothesis that supervised techniques (i.e., PLS-R and PcovR) would outperform unsupervised techniques (i.e., PCA and ICA) does not seem to hold entirely, since (T-)PCA was the second best performing technique, herewith outperforming the supervised PcovR technique. However, both supervised techniques did perform better than their unsupervised counterparts when only a small amount of components was used for classification. As opposed to our expectations, the more flexible PcovrR did not outperform PLS-R both in terms of classification accuracy as in computation speed. One reason for this may be that the classification performance of PcovR is not optimal in this study due to suboptimal decisions made regarding the implementation of PcovR (e.g.,

applying it to PCA component scores instead of original variables, selecting a non-optimal α-value, not standardizing the variables prior to analysis).

4.3 Limitations of the current study

Several choices made regarding the analysis of the data may have had some unknown effect on the obtained results, and may, therefore, point at some limitations of the current study.

A first limitation pertains to the fact that (T-)ICA, PLS-R and PcovR were all applied on the PCA component scores instead of on the original variables. This may have affected the results as there is no guarantee that PCA is an optimal pre-processing approach for the use of ICA, PLS-R and PcovR. Although PCA has been proven to be an effective method for dimension reduction before applying ICA (Sui, Adali, Yu, Chen, & Calhoun, 2011; Castro et al., 2011), little is known about the value of using PCA as a preliminary dimension reduction step before applying PLS-R and PcovR. Better classification performances may be obtained when PLS-R and PcovR would have been performed to the original variables directly. It is, however, not yet clear whether and how PLS-R and PcovR are able to analyze very high-dimensional data and whether this is possible within a reasonable amount of time. Using sparse versions of PLS-R and PcovR, in which it is assumed that only a small number of variables contribute to each linear combination, may be an option here.

Regarding a second limitation, it should be noted that the full potential of PcovR in terms of classification performance might not have been revealed by this study, for which two main reasons may exist. First of all, whereas choosing the optimal α-value by means of maximum likelihood principles is a fast approach (and therefore embraced in this study), determining this parameter through cross-validation, which is computationally more intensive, may result in better classification accuracies for PcovR. Secondly, in order to fairly compare the results across feature extraction techniques, the input variables (i.e., the PCA component scores) were not standardized before applying PcovR. An additional analysis (see results in Appendix D), however, indicated that for the EX and MR data sets, the classification performance - in terms of percentage agreement - of PcovR is better when its input variables are standardized prior to analysis. Another possible limitation regarding PcovR is that occasionally, the PcovR model failed to converge. As a result, for some combinations of data set and the number of extracted components, PcovR has less than 100 estimates of the AUC-

value (see Appendix E). Whether there is a relationship between whether or not the PcovR model convergences and the predictive abilities of the extracted components is not known.

Further limitations of this study are related to the classifier used, the type of feature selection approaches compared and the fact that only information from a single neuroimaging property at a time is used for classification. With respect to the classifier adopted, feature extraction may enhance classification performance to a stronger degree when a classifier is used for which its classification performance is less insensitive to the high-dimensionality of the data, like, for example, lasso logistic regression. Using the same subjects and neuroimaging properties, de Vos et al. (submitted) found somewhat lower classification accuracies when adopting lasso logistic regression as a classifier. Also the choice of feature extraction approaches to include in the study might have been somewhat one-sided. Indeed, all the included approaches are linear feature extraction techniques, in which extracted features are restricted to linear combinations (i.e., weighted sums) of the original variables/voxels. Non-linear feature extraction approaches (e.g., kernel versions of PCA, ICA, PLS-R and PcovR) may be more flexible in retrieving essential classification-related information from the original variables/voxels, herewith increasing the classification performance of the feature extraction approaches. Finally, only looking at a single neuroimaging property at a time may obscure information important for the classification that is hidden in the relationships between the properties. Evidence exists that combining information from multiple neuroimaging properties, like, for example, information on functional and structural brain functioning, may enhance classification performance (Schouten et al., 2016).

4.4 Recommendations for further research

Although whole-brain analysis was not outperformed by the feature extraction approaches for the majority of neuroimaging data sets, the analysis of the ALFF data showed that feature extraction can increase classification performance compared to using whole-brain data as input for a SVM. Therefore, the predictive abilities of various feature extraction methods on other neuroimaging properties as well as for other classification tasks should be investigated in future studies.

In this study, the classification performance of ICA did not exceed that of PCA when ICA was applied to the PCA component scores using the FastICA algorithm. This finding,

however, cannot be generalized to all ICA algorithms and possible dimension reduction steps preceding ICA. Therefore, the design of the current study could be extended by also considering other ICA algorithms, as well as pre-processing steps for ICA. A useful point of departure could be the work of Calhoun & Adali (2006), who, in order to un-mix fMRI data preceding classification, successfully utilized the Infomax algorithm; Infomax computes ICA by means of an Information-Maximization (Infomax) algorithm (Bell & Sejnowski, 1995). Their study revealed that the Infomax algorithm provided better results in terms of distinguishing between schizophrenic patients and HC's compared to the FastICA algorithm used in the current study. Moreover, the dimensionality problem that ICA encounters was, aside from using PCA, also dealt with by applying ICA on several clusters of brain data (by grouping neighboring voxels into clusters) instead of on all data features (Calhoun & Adali, 2006). A future study could evaluate the predictive abilities of the components extracted through each combination of an ICA algorithm (i.e., Infomax vs. FastICA) and a pre-processing step (i.e., PCA vs. a cluster-approach), for example.

To our knowledge, this is the first study in which PcovR was utilized as a way of dimension reduction preceding the training of a SVM on neuroimaging data. Although promising classification accuracies were obtained using the components derived with PcovR as input features for a SVM classifier, the effectiveness of PcovR might be enhanced even further (see also Section 4.3). Therefore, the use of PcovR in neuroimaging classification studies should be investigated to a deeper extent, taking the reasons underlying the possible underestimation of its classification performance into account. To this end, the alternative code for the PcovR algorithm (see Appendix A) could be utilized to apply PcovR on the original data (instead of on PCA components), since the original code written by Vervloet et al. (2015) cannot handle high-dimensional data. A future study could also focus on optimizing the procedure to determine an optimal α-value (e.g., by using cross-validation). Moreover, the influence of several pre-processing steps, like whether or not to standardize the input variables, on the classification performance of PcovR could be investigated. Results from a pilot study (see Appendix D) suggest that a PcovR analysis of standardized data may yield better classification accuracies than applying PcovR to non-standardized data. A reason for this observation may be that when the data are non-standardized, the influence of α on the solution also depends on the (differences in the) scales of the predictors and the criterion, which may impede making an optimal choice for α. When the data are standardized, the influence of the α-weight on the obtained PcovR components is more univocal, herewith facilitating the search for an optimal α.

All feature extraction methods that were employed in the current study are linear methods, which aim to determine a low-dimensional linear subspace to which the data at hand are confined to. However, if the data in reality confines to a non-linear subspace, the feature extraction methods in this study might not be able to extract all classification-related information from the original variables. In contrast, kernel feature extraction methods are non-linear methods that are able to project the original data onto a non-linear subspace, and are therefore more flexible in retrieving information hidden in the relationships between voxels. This study could be extended by using the kernel-version of each feature extraction method to extract components for classification. To this end, the kernel PCA framework, which has already proven to be an effective pre-processing step for classification algorithms (Mika et al., 1998) of Schölkopf, Smola, & Müller (1997) could be utilized. Regarding a kernel-based version of PLS-R, the method proposed by Rosipal & Trejo (2001) could be adopted, while as for kernel ICA, the work of Bach & Jordan (2002) could be consulted. No kernel based method of PcovR exists as to date, and the forthcoming of such a method could be a topic for future research as well.

In this study, the classification accuracy that resulted from several feature extraction techniques (as well as whole-brain analysis), using various neuroimaging data sets, for a varying amount of extracted components was evaluated. However, a parameter that remained constant across all conditions pertains to the sample size, and therefore the size of the training set. Differences in classification performance between the feature extraction methods and whole-brain analysis, however, might also be influenced by the size of the sample. In this regard, Chu et al. (2012) demonstrated that the difference in classification performance between whole-brain analysis and feature selection methods was reduced as the sample size increased. As a result, whole-brain analysis was not outperformed by the feature extraction methods, provided that the sample size was large enough. For small sample sizes, however, feature selection occasionally outperformed whole-brain analysis (Chu et al., 2012). A natural question reads whether varying sample sizes would also change the in this study observed pattern of difference in performance between feature extraction methods and whole-brain analysis. To this end, it would be worthwhile to examine whether feature extraction outperforms whole-brain analysis when confronted with small sample sizes, which seems to be the case for feature selection.

Finally, as this study only focused on using each neuroimaging property separately, the question whether combining information from various neuroimaging properties could possibly enhance classification performance is also worth investigating. In this regard,

Schouten et al. (2016) showed that different MRI modalities provide complementary information for classifying people with AD and that combining modalities can enhance the classification performance compared to using each modality in isolation. A difference between their study and the current study is the use of an elastic-net regression (vs. SVM) classifier, which can be seen as a feature selection step combined with logistic regression as a classifier. A future study could examine whether combining information from several neuroimaging properties in combination with some form of feature extraction can improve classification performance in the context of a SVM classifier. A challenging question herewith pertains to which feature extraction method is optimal in terms of selecting the best features from multiple modalities to be used for classification. A possibility here consists of using Simultaneous Component Analysis (Smilde et al, 2005) or (Generalized) Canonical Correlation Analysis (Tenenhaus & Tenenhaus, 2011) to extract important features from multiple modalities.

In a nutshell, it is important to continue to explore the potential of feature extraction techniques in classification studies using (combinations of) various types of high-dimensional neuroimaging data, with the aim of obtaining higher classification accuracies compared to when whole-brain data is used as input for a SVM classifier.

References

Bach, F. R., & Jordan, M. I. (2002). Kernel independent component analysis. *Journal of Machine Learning Research, 3,* 1-48.

Ballings, M., & van den Poel, D. (2013) AUC: Threshold independent performance measures for probalistic classifiers. R package version 0.3.0. http://CRAN.R-project.org/package=AUC

Beaton, D., Dunlop, J., & Abdi, H. (2016). Partial Least Squares Correspondence Analysis: A framework to simultaneously analyze behavioral and genetic data. *Psychological Methods, 21,* 621-651.

Bell, A. J., & Sejnowski, T. J. (1995). An information-maximization approach to blind seperation and blind deconvolution. *Neural Computation, 7,* 1129-1159.

Calhoun, V. D., & Adali, T. (2006). Unmixing fMRI with Independent Component Analysis: using ICA to characterize high-dimensional fMRI data in a concise manner. *Medicine and Biology Magazine, 12,* 79-90.

Chai, J., Chen, C., Chiang, C., Ho, Y., Chen, H., Ouyang, Y., . . . Chang, C. (2010). Quantitative analysis in clinical applications of brain MRI using independent component analysis coupled with Support Vector Machine. *Journal of Magnetic Resonance, 32,* 24-34.

Chu, C., Hsu, A., Chou, K., Bandinetti, P., & Lin, C. (2012). Does feature selection improve classification accuracy? Impact of sample size and feature selection on classification using anatomical magnetic resonance images. *NeuroImage, 60,* 59-70.

Delac, K., Grgic, M., & Grgic, S. (2005). Independent comparative study of PCA, ICA and LDA on the FERET data set. *International Journal of Imaging Systems and Technology, 15(5),* 252-260.

de Jong S., & Kiers, H. A. L. (1992). Principal covariates regression: Part I. Theory. *Chemometrics and Intelligent Laboratory Systems, 14(1-3),* 155-164.

de Vos, F., Koini, M., Schouten, T. M., Seiler, S., van der Grond, J., Lechner, A., . . . Rombouts, S. A. R. B. (2016). *A comprehensive analysis of resting state fMRI measures to classify individual patients with Alzheimer's disease.* Manuscript submitted for publication.

Douglas, P. K., Harris, S., Yuille, A., & Cohen, M. S. (2011). Performance comparison of machine learning algorithms and number of independent components used in fMRI decoding of belief vs. disbelief. *NeuroImage, 56,* 544-553.

Feng, Y. Z., Yong, H., Chao-Zhe, Z., Qing-Jiu, C., Man-Qiu, S., Meng, L., . . . Yu-Feng, W. (2007). Altered baseline brain activity in children with ADHD revealed by resting-state functional MRI. *Brain & Development, 39,* 83-91.

Ford, J., Farid, H., Makedon, F., Flashman, L. A., McAllister, W., Megalooikonomou, V., & Sayking, A. J. (2003, November). Patient classification of fMRI activiation maps. Paper presented at the *International Conference on Medical Image Computing and Computer-Assisted Intervention.* Retrieved from http://www.cs.dartmouth.edu/farid/downloads/publications/miccai03b.pdf

Hanley, J. A., & McNeil, B. J. (1982). The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology, 148(1),* 29-36.

Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The elements of statistical learning*. New York, NY: Springer.

Helwig, N. E. (2015*). ica: Independent Component Analysis. R package version 1.01* .http://CRAN.R-project.org/package=ica.

Hyvärinen, A. (1999). Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks, 10,* 626-634.

Hyvärinen, A., Karhunen, J., & Oja, E. (2001). *Independent Component Analysis*. New York, NY: John Wiley & Sons.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2015). *An introduction in statistical learning* (6th Edition). New York: Springer.

Kiers, H. A. L., & Smilde A. K. (2007). A comparison of various methods for multivariate regression with highly collinear variables. *Statistical Methods and Applications, 16(2),* 193-228.

Kloppel, S., Stonnington, C. M., Chu, C., Draganski, B., Scahill, R. I., Rohrer, J. D., . . . Ashburner, J. (2008). Automatic classification of MR scans in Alzheimer's disease. *Brain, 131(3),* 681-689.

Koutsoleris, N., Meisenzahl, E. M., Davatzikos, C., Bottlender, R., Frodl, T., Scheurecker, J., . . . Gaser, C. (2009). Use of neuroanatomical pattern classification to identify subjects in at-risk mental states of psychosis and predict disease transition. *Psychiatry, 66(7),* 700-712.

Krishnan, A., Williams, L. J., McIntosh, A. R., & Abdi, H. (2010). Partial Least Squares (PLS) methods for neuroimaging: A tutorial and review. *Neuroimage, 4,* 1-21.

Linden, D. E. J. (2012). The challenges and promise of neuroimaging in psychiatry. *Neuron, 73,* 8-22.

Magnin, B., Mesrob, L., Kinkingnéhun, S., Pelegrini-Issac, M., Colliot, O., Sarazin, M., Dubois, B., Lehericy, S., & Benali, H. (2008). Support vector machine-based classification of Alzheimer's disease from whole-brain anatomical MRI. *Neuroradiology, 51(2),* 73-83.

Mckhann, G., Drachman, D., Folstein, M., (1984). Clinical diagnosis of Alzheimer's disease Report of the NINCDS-ADRDA Work Group under the auspices of Department of Health and Human Services Task Force. *Neurology, 34,* 939–944.

Menzies, L., Achard, S., Chamberlain, S.R., Fineberg, N., Chen, C.-H., delCampo, N., Sahakian, B.J., Robbins, T.W., & Bullmore, E. (2007). Neurocognitive endophenotypes of obsessive–compulsive disorder, *Brain 130*, 3223–3236.

Mevik, B. H., Wehrens, R., & Liland, K. H. (2013). *pls: Partial Least Squares and Principal Component Regression. R package version 2.4-3.* http://CRAN.Rproject.org/package=pls.

Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., & Leisch, F. (2015). *e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien. R package version 1.6-7.* http://CRAN.Rproject.org/package=e1071.

Mika, S., Schölkopf, B., Smola, A. J., Müller, K. R., Scholz, M., & Rätsch, G. (1998). Kernel PCA and de-noising in feature spaces. *Neural Information Processing Systems, 11,* 536-542.

Mourao-Miranda, J., Bokde, A. L., Born, C., Hampel, H., & Stetter, M. (2005). Classifying brain states and determining the discriminating activation patterns: Support Vector Machine on functional MRI data. *NeuroImage, 28,* 980–995.

Mwangi, B., Tian, S. T., & Soares, J. C. (2014). A review of feature reduction techniques in neuroimaging. *Neuroinformatics, 12(2),* 229-244.

Nestor, P. G., O'Donnell, B. F., Mccarley, R. W., Niznikeiwicz, M., Barnard, J., Shen, Z. J., . . . Shenton, M. E., (2002). A new statistical method for testing hypotheses of neuropsychological/MRI relationships in schizophrenia: partial least squares analysis. *Schizophenia Research, 53,* 57–66.

Nguyen, D. V., & Rocke, D. M. (2012). Tumor classification by partial least squares using microarray gene expression data. *Bioinformatics, 18,* 39–50.

R Core Team (2015). *R: A language and environment for statistical computing. R Foundation for Statistical Computing*, Vienna, Austria. URL http://www.R-project.org/.

Rosipal, R., & Trejo, L. J. (2001). Kernel partial least squares regression in reproducing kernel Hilbert space. *Journal of Machine Learning, 2*, 97-123.

Schölkopf, B., Smola, A., & Müller, K. R. (1997). Kernel principal component analysis. Paper presented at the *International conference on Artificial Neural Networks*. Retrieved from https://link.springer.com/chapter/10.1007/BFb0020217

Schouten, T. M., Koini, M., de Vos, F., Seiler, S., van der Grond, J., Lechner, A., . . . Rombouts, S. A. R. B. (2016). Combining anatomical, diffusion, and resting state functional magnetic resonance imaging for individual classification of mild and moderate Alzheimer's disease. *NeuroImage, 11,* 46-51.

Seiler, S., Schmidt, H., Lechner, A., Benke, T., Sanin, G., Ransmayr, . . . Schmidt, R. (2012). Driving cessation and dementia: results of the prospective registry on dementia in Austria (PRODEM). *PLoS One 7(12),* 1-5.

Smilde, A. K., Bro, R., & Geladi, P. (2004). *Multi-way analysis with applications in the chemical sciences*. Chichester, UK: Wiley.

Smilde, A. K., Jansen, J. J., Hoefsloot, H. C., Lamers, R. J. A., van der Greef, J., & Timmerman, M. E. (2005). ANOVA-simultaneous component analysis (ASCA): a new tool for analyzing designed metabolomics data. *Bioinformatics, 21(13),* 3043-3048.

Stone, J. (2004). *Independent Component Analysis*. Cambridge, MA: MIT Press.

Sui, J., Adali, T., Yu, Q., Chen, J., & Calhoun, V. D. (2012). A review of multivariate methods for multimodal fusion of brain imaging data. *Journal of Neuroscience Methods, 204,* 68-81.

ten Berge, J. M. F. (1993). *Least squares optimization in multivariate analysis.* Leiden: DSWO Press.

Tenenhaus, A., & Tenenhaus, M. (2011). Regularized generalized canonical correlation analysis. *Psychometrika, 76(2),* 257-284.

Vapnik, V. (1995). *The nature of statistical learning theory*. New York: Springer.

Vervloet, M., Kiers, H. A. L., Noortgate, W., & Ceulemans, E. (2015). PcovR: An R package for Principal Covariates Regression. *Journal of Statistical Software, 65(8),* 1-14.

Wold, S., Sjöström, M., & Eriksson, L. (2001). PLS-regression: a basic tool of chemometrics. *Chemometrics and Intelligent Laboratory Systems, 58,* 109-130.

Yang, W., Lui, R. L. M., Gao, J. H., Chan, T. F., & Yau, S. T. (2011). Independent component analysis-based classification of Alzheimer's disease. *Journal of Alzheimer's Disease, 24,* 775-783.

Yoon, U., Lee, J. M., Im, K., Shin, Y. W., Cho, B. H., Kim, I. Y., Kwon, J. S., & Kim, S. I. (2007). Pattern classification using principal components of cortical thickness and its discriminative pattern in schizophrenia. *NeuroImage, 34,* 1405-1413.

Zhao, Z., Lu, J., Jia, X., Chao, W., Han, Y., Jia, J., & Li, K. (2014). Selective changes of resting-state brain oscillations in aMCI: An fMRI study using ALFF. *BioMed Research International, 2014,* 1-7. http://dx.doi.org/10.1155/2014/920902

Zhu, C. Z., Zang, Y. F., Cao, Q. J., Yan, C. G., He, Y., Jiang, T. Z., Sui, M. Q., & Wang, Y. F. (2008). Fisher's discriminative analysis of resting-state brain function for attention-deficit/hyperactivity disorder. *NeuroImage, 40,* 110-120.

Appendix A. Alternative R-code to perform PcovR-analysis

Code for alternative implementation of the PcovR function (without built-in method to optimize α). This code was used in order to directly apply PcovR to the PC data, at the cost of not having a built in method of optimizing alpha.

The main function *PcovR_Alternative* calls two auxiliary functions: *ssq* and *ed*.

```
ssq <- function( X )
{
  out = sum( X^2 )
  return( out )
}


ed <- function( X , tol=1e-9 )
{
  #Computes sorted eigenvalues and eigenvectors
  #  X: a square data matrix (preferably symmetric)
  #  tol: a tolerance value (default: 1e-9)

  # returns Out
  #   values: sorted eigenvalues (in a vector)
  #   vectors: associated eigenvectors (in the columns of a matrix)

  temp = eigen( X , only.values = FALSE, EISPACK = FALSE )

  #sort( temp$values , decreasing = TRUE ) #sorted eigenvalue
  OptOrder = order( temp$values , decreasing = TRUE ) #order of the eigenvalues
  tempvalues = temp$values[ OptOrder ]
  tempvectors = temp$vectors[ , OptOrder ]

  #select eigenvalues and associated eigenvectors larger than a given tolerance value
  SelectedValues = abs(tempvalues) > tol
  values = tempvalues[ SelectedValues ]
  vectors = tempvectors[ , SelectedValues ]
  vectors = vectors %*% diag( sqrt( colSums( vectors ^ 2 ) ) ^ -1 )

  #standardize the vectors
  Out = list()
  Out$values = values
  Out$vectors = vectors
  return( Out )
}
```

This is the alternative function for PcovR.

```
PcovR_Alternative <- function( X , y , nComp , Alfa )
{
 # X (nRow x xnCol): predictor matrix (with predictors as columns)
 # y (nRow x 1): criterion vector
 # nComp: number of components (1, 2, ..., min(nRow,nCol) )
 # Alfa(0-1): alfa weight (a number between 0 and 1, but not being 0 or 1)


 require( MASS )

 checkinput = 1

 Xdim = dim( X )
 Ydim = dim( y )
 if( Ydim[2] > 1 )
 {
  cat( " " , fill=TRUE )
  cat( "y should be a vector containing the scores on a single criterion variable" , fill=TRUE )
  cat( " " , fill=TRUE )
  checkinput=0
 }

 if( Xdim[1] != Ydim[1] )
 {
  cat( " " , fill=TRUE )
  cat( "the number of rows in X should match the number of elements in y" , fill=TRUE )
  cat( " " , fill=TRUE )
  checkinput=0
 }
 else
 {
  nElem = Xdim[1]
  nPred = Xdim[2]
  rm( Xdim , Ydim )
 }

 if( nComp > min( nElem , nPred ) )
 {
  cat( " " , fill=TRUE )
  cat( "nComp should be an integer between 1 and " , min(nElem,nPred) , fill=TRUE )
  cat( " " , fill=TRUE )
  checkinput=0
 }

 if ( (Alfa < 0) || (Alfa >= 1) )
 {
  cat( " " , fill=TRUE )
  cat( "Alfa should be between 0 and 1 (but not 0 or 1)" , fill=TRUE )
  cat( " " , fill=TRUE )
  checkinput=0
 }

 Out = list()
```

```
  if( checkinput == 1 )
  {
   # compute projector on X
   S = t(X) %*% X

   if( det(S) < 1e-12 ) #near-singular
   {
    tempsol = ed( S )
    selval = tempsol$values > (1e-8 * max(tempsol$values))
    tempval = tempsol$values[selval] ^ -1
    Sh = tempsol$vectors[ , selval ] %*% diag( tempval ) %*% t( tempsol$vectors[ , selval ] )
    Hx = X %*% Sh %*% t(X)
   }
   else
   {
    Hx = X %*% solve(S) %*% t(X)
   }

   # Compute PCovR solution: see de Jong & Kiers (1992)
   G = ( Alfa / ssq( X ) ) * X %*% t(X) + ( ( 1 - Alfa ) / ssq(y) ) * Hx %*% y %*% t(y) %*% Hx
   sol = ed( G )
   #try( if(nComp > length(sol$values)) stop("not possible to extract the specified number of components
(nComp)") )
   T = sol$vectors[ , 1:nComp ]  # Note: T = K[,1:r] = XW
   W = ginv(X) %*% T             # T = XW = K so use Moore Penrose inverse of X
   Px = t(T) %*% X
   Py = t(T) %*% y
   Rx2 = ssq(T %*% Px) / ssq(X)
   Ry2 = ssq(T %*% Py) / ssq(y)
   B = W %*% Py

   Out$W = W
   Out$B = B
   Out$Rx2 = Rx2
   Out$Ry2 = Ry2
  }

  return( Out )
}
```

Appendix B. Full plot figures regarding classification performance

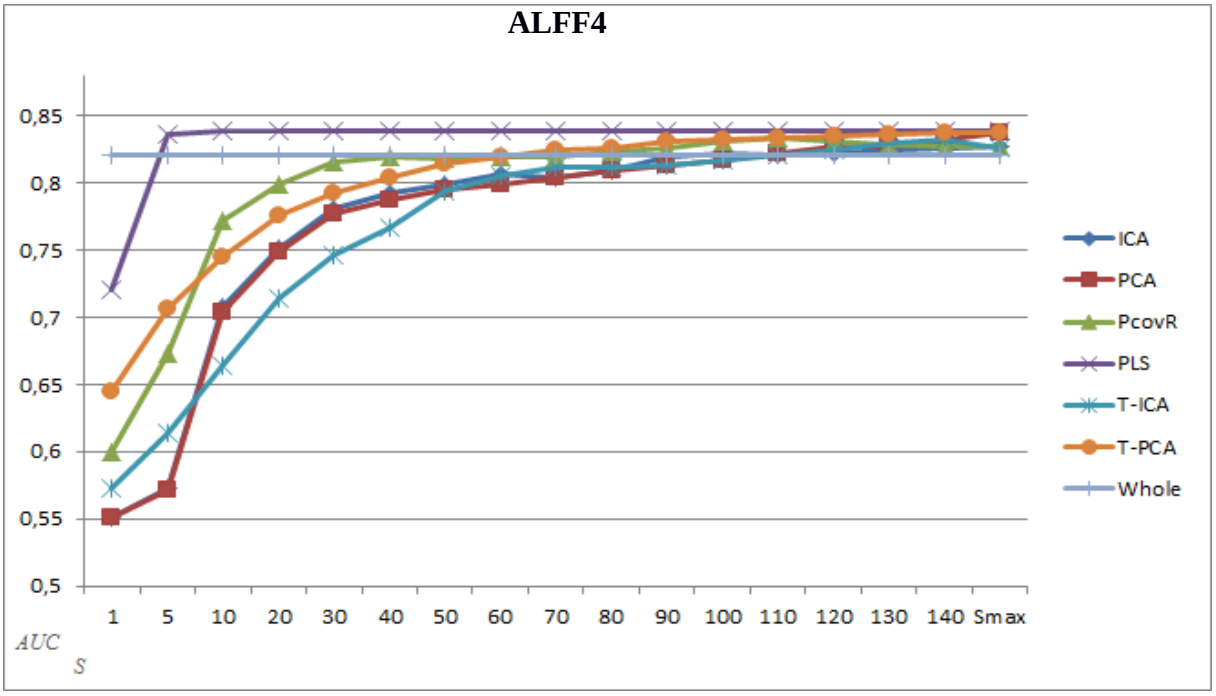

Figure 1. Mean AUC-values plotted against the number of components *S* for the ALFF4 data. The various curves represent the results for whole-brain analysis (flat grey line) and the six feature extraction methods (PCA, T-PCA, ICA, T-ICA, PLS-R and PcovR).

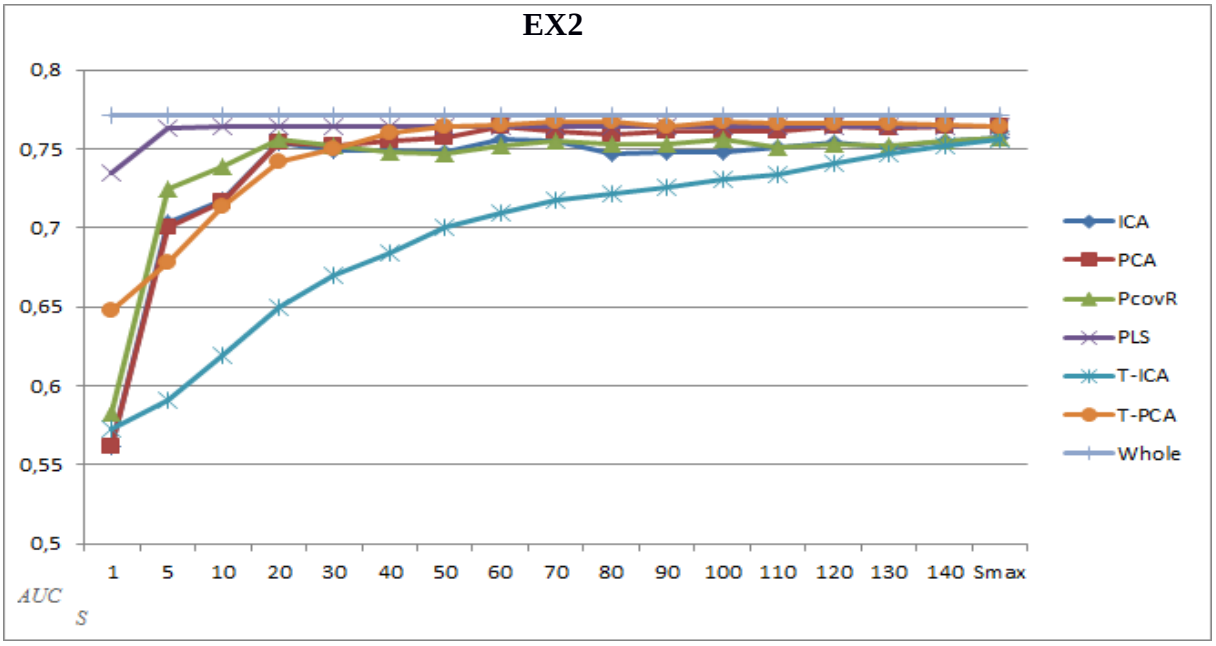

Figure 2. Mean AUC-values plotted against the number of components *S* for the EX2 data. The various curves represent the results for whole-brain analysis (flat grey line) and the six feature extraction methods (PCA, T-PCA, ICA, T-ICA, PLS-R and PcovR).

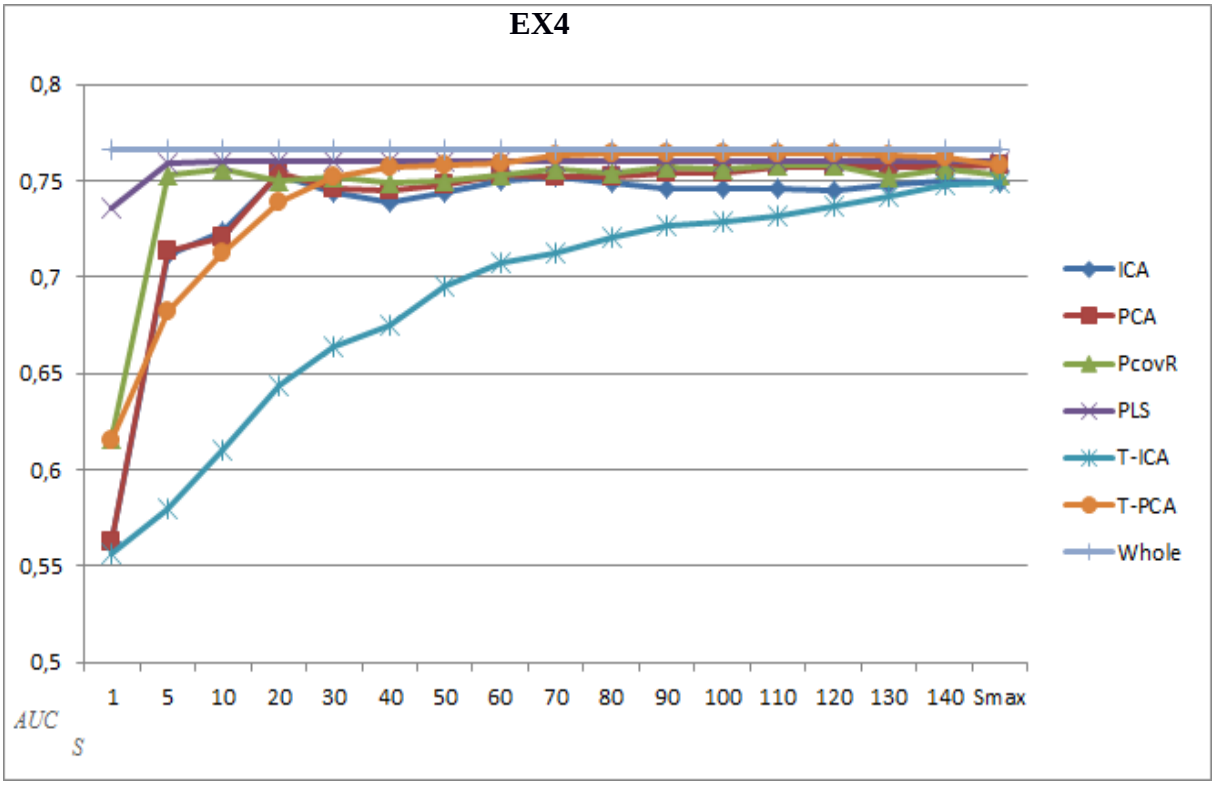

Figure 3. Mean AUC-values plotted against the number of components *S* for the EX4 data. The various curves represent the results for whole-brain analysis (flat grey line) and the six feature extraction methods (PCA, T-PCA, ICA, T-ICA, PLS-R and PcovR).

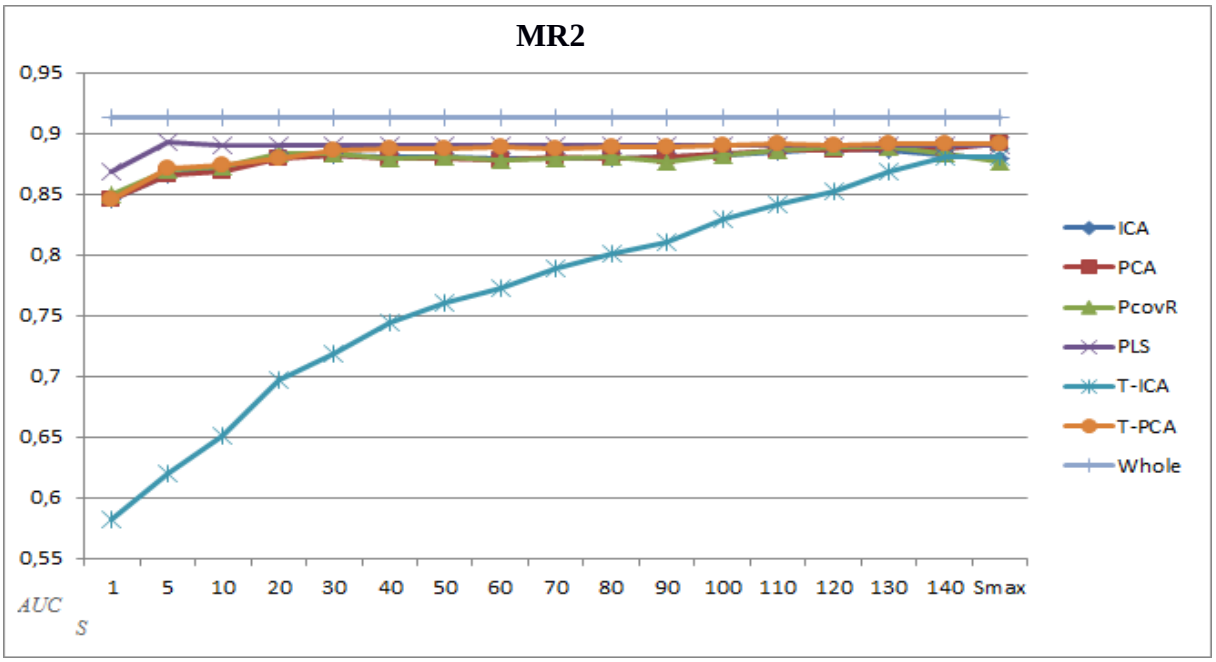

Figure 4. Mean AUC-values plotted against the number of components *S* for the MR2 data. The various curves represent the results for whole-brain analysis (flat grey line) and the six feature extraction methods (PCA, T-PCA, ICA, T-ICA, PLS-R and PcovR).

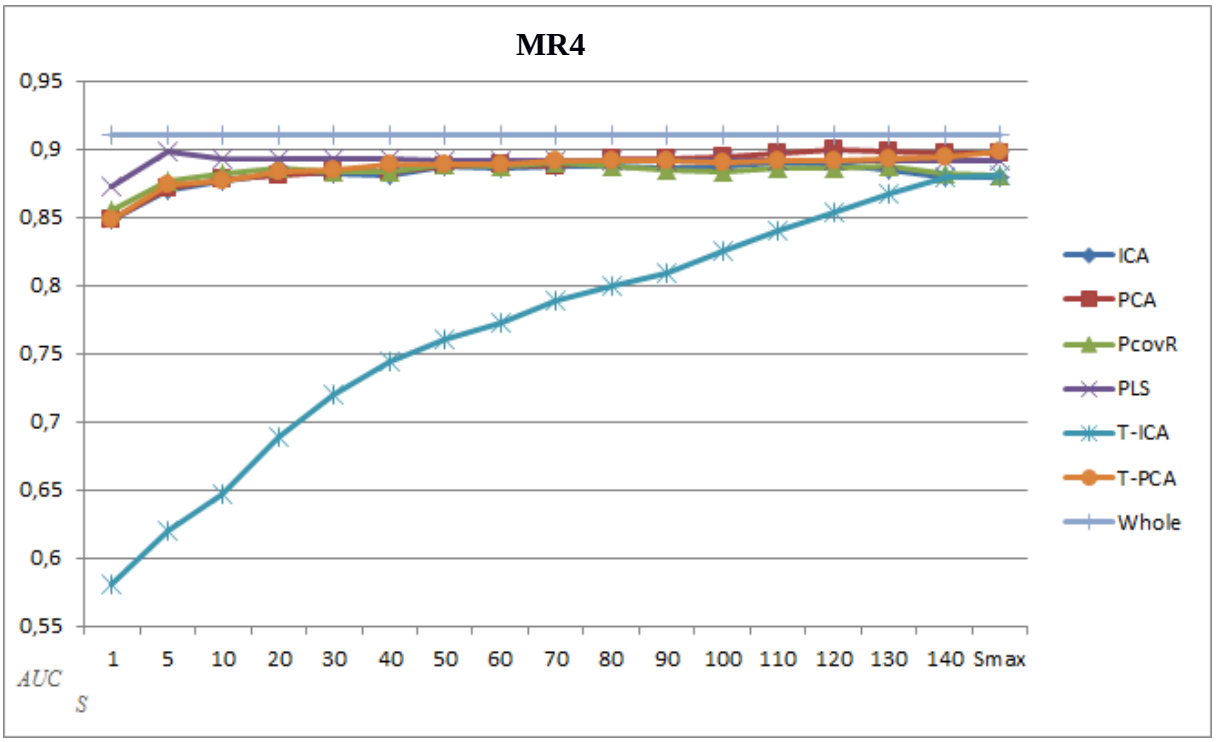

Figure 5. Mean AUC-values plotted against the number of components *S* for the MR4 data. The various curves represent the results for whole-brain analysis (flat grey line) and the six feature extraction methods (PCA, T-PCA, ICA, T-ICA, PLS-R and PcovR).

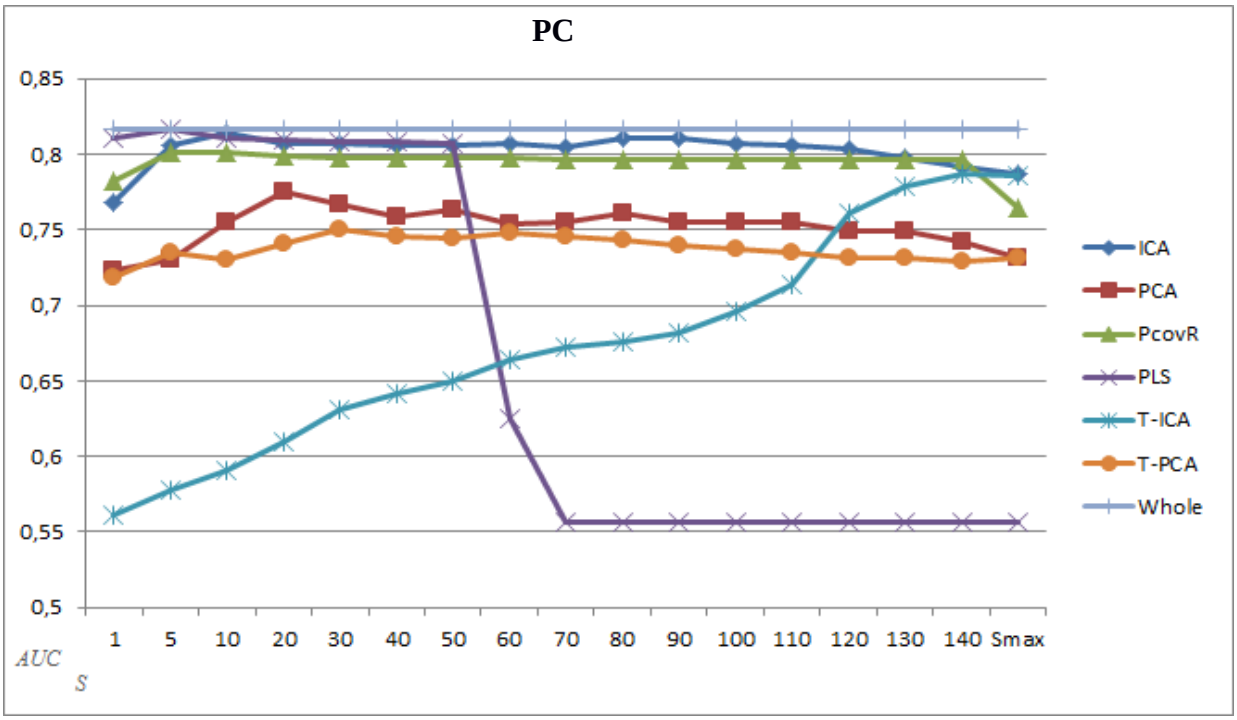

Figure 6. Mean AUC-values plotted against the number of components *S* for the PC data with. The various curves represent the results for whole-brain analysis (flat grey line) and the six feature extraction methods (PCA, T-PCA, ICA, T-ICA, PLS-R and PcovR).

Appendix C. Computation times for the ALFF2, EX4 and MR4 data

Table 1

Computation time (in seconds) of performing feature extraction (first column), fitting the SVM (second column) and total time (third column) for the various feature extraction methods using S_{max} (rows) and data sets (blocks of three columns)

	ALFF2			EX4			MR4		
Technique	**FE**	**SVM***	**Total**	**FE**	**SVM***	**Total**	**FE**	**SVM***	**Total**
PCA	60.64	13.82	74.46	8.35	14.00	22.35	15.70	13.90	29.60
T-PCA	60.78	13.65	74.43	8.49	13.94	22.43	15.83	13.94	29.77
ICA	63.00	13.69	76.69	10.73	13.84	24.57	18.03	13.85	31.88
T-ICA	63.14	13.70	76.84	10.94	14.03	24.97	18.24	13.89	32.13
PLS-R	60.84	13.88	74.72	8.59	13.89	22.48	15.94	13.79	29.73
PcovR	64.89	13.62	78.51	12.12	13.78	25.90	19.64	13.83	33.47
WB	-	55.42	55.42	-	8.41	8.41	-	23.32	23.32

Note. WB = whole brain, FE = feature extraction method, SVM = support vector machine.
*SVM is performed with five-fold cross-validation and eleven C-values for the feature extraction methods and without cross-validation/different C-values for whole-brain analysis.

Appendix D. Influence of pre-processing on classification results for PcovR

Table 1

Classification accuracy in terms of percentage agreement for a set of values of the number of components S (rows) for several data sets (columns), both when the input data was centered (Cent) and standardized (Stand) before the PcovR analysis

	EX2		EX4		MR2		MR4	
NC	**Cent**	**Stand**	**Cent**	**Stand**	**Cent**	**Stand**	**Cent**	**Stand**
1	.599	.690	.595	.699	.783	.815	.794	.824
5	.685	.695	.711	.702	.796	.817	.819	.825
10	.695	.695	.702	.707	.822	.841	.833	.835
20	.704	.716	.723	.719	.838	.847	.834	.849
40	.707	.719	.717	.719	.829	.843	.833	.852
60	.716	.717	.720	.726	.832	.845	.839	.849
80	.715	.719	.721	.720	.829	.847	.842	.853
100	.714	.717	.722	.718	.831	.852	.835	.854
125	.711	.715	.719	.721	.833	.851	.828	.854
145	.713	.717	.717	.718	.843	.845	.850	.852

Note: Cent = centered. Stand = standardized. NC = number of components.

Appendix E. Number of AUC estimates of each PcovR analysis

Table 1

Number of successful PcovR analyses, and therefore estimates of AUC, for every combination of number of components NC (rows) and data set (columns)

NC	ALFF2	ALFF4	EX2	EX4	MR2	MR4	PC
1	100	100	100	100	100	100	100
5	100	100	100	100	100	100	100
10	100	100	100	100	100	100	100
20	100	100	100	100	100	100	100
30	100	97	96	99	99	100	100
40	100	100	97	100	97	100	100
50	100	99	100	100	100	100	100
60	98	95	99	96	92	95	100
70	97	94	97	99	88	95	100
80	99	97	99	99	96	94	100
90	99	99	97	96	96	98	100
100	98	98	99	97	96	99	100
110	95	90	90	86	89	88	100
120	93	91	94	90	87	86	100
130	94	80	90	90	92	91	100
140	94	93	96	94	95	89	100
150	96	91	96	92	92	95	100

Note: NC = number of components.