Transactions on Machine Learning
and Data Mining
Vol. 9, No. 1 (2016) 27-45
ISSN 1864-9734
ISBN: 879-3-942952-41-5

ibai Publishing
www.ibai-publishing.org

Combining Multiple Feature Selection Methods and Deep Learning for High-dimensional Data

Mihaela A. Mares[1], Shicai Wang[1], and Yike Guo[1,2,*]

[1]Imperial College Data Science Institute , Imperial College London, UK
[2]School of Computer Science, Shanghai University, Shanghai, China
[*]Corresponding author: y.guo@imperial.ac.uk

Abstract. Feature or variable selection when the number of features is relatively large to the number of samples or $n << p$ is a challenge in many machine learning applications. A large number of statistical methods have been developed to address this challenge. Each method uses different statistical assumptions about the shape of the regression function relating the predicted variable to the predictors. In this paper we propose an alternative: combining results from different feature selection methods relying on disjoint assumptions about the regression function. We show that our method will lead to better sensitivity than using different methods individually, on synthetic datasets and datasets from the UCI machine learning repository. Our empirical studies on data with $n << p$ show that the accuracy obtained when training deep neural networks with variables selected using our method is at least as good as the accuracy obtained when not selecting variables in advance. Our first conclusion is that the feature selection results are improved by enlarging the body of limiting assumptions about the function relating the predicted variable to the predictors. Our second conclusion is that, feature selection can improve accuracy in deep learning at least on data with $n << p$.

Keywords: combining feature selection, high-dimensional data, deep learning, nonlinear regression, variable selection.

1 Introduction

One of the main statistical problems, coming with large dimensionality of data is variable or feature selection. When given a large number of variables or features, training classifiers and regression algorithms using all of them could lead to results no better

than random choice as shown for example in [1]. This is the case because most prediction algorithms, which don't have embedded feature selection mechanisms, attribute a small, noisy contribution of a large number of wrong predictors to the response variable or to the features extracted to fit the response response variable. This large number of small noisy contributions to the response will in turn add up and lead to high prediction errors [1]. Variable selection is therefore needed to select a small subset of variables from a wider set, such that, the selected variables explain as much variability in a given response, as possible.

This task is particularly important and challenging in high-dimensional data when when the number of samples n is relatively small to the number of variables p. This is a typical scenario in many machine learning applications as for example in bioinformatics where, usually only several hundreds of samples are collected from cost and availability reasons, yet each sample may contain thousands or even millions of biological markers. Different methods have been developed for feature selection and they usually rely on simplifying assumptions regarding the shape of the underlying function connecting the response to the predictor variables (an example of such simplifying assumption is linearity of the regression function). That is because searching and scoring through the features subsets space by training a highly non-linear prediction algorithm is computationally infeasable. In this context, we are firstly interested in selecting a few feature selection methods that can cope with highly dimensional datasets particularly when $n << p$. If these methods rely on partially different assumptions about the underlying regression function that generated the response variable data then, combining their results will cover a larger set of possible underlying data generating function shapes (for example the predictors could have small and large effect linear contributions, small effect non-linear contribution and small or large effect interactions contributions to the predicted variable). Therefore, our first hypothesis is that combining the results from different methods as explained above will lead to better sensitivity or a larger number of correctly selected features. The purpose of variable or feature selection is usually to further train a classifier or prediction model using the selected variables. Therefore, we are interested to which extent combining the results from different methods will also lead to better accuracy of the prediction models in use. As prediction models, recently, Deep Neural Network (DNNs) have gained popularity due to their high accuracy rates in areas like speech or image recognition [2]. DNNs architectures have embedded feature extraction mechanisms in which a new layer of features are learned as functions of the previous layer. In this context, we are lastly interested to which extent DNNs can be used as prediction models in a typical $n << p$ set-up and whether feature selection methods are still useful a-priori training DNNs.

Most feature selection research efforts were directed towards improving individual techniques. Some recent state-of-the-art feature selection algorithms where focused for example, on improving the regularization under the assumption of linear relationship between the predictive features and the predicted variable [3], [4], [5]. Other recent state-of-the-art methods [6], [7] such as the ones improving spectral feature selection fall into the class of pair-wise feature selection which discards the possibility that the response variable might be a complex function of multiple predictor variables, including perhaps interactions. Another attempt to formalize pair-wise feature selection and thus

focused on the class of assumptions mentioned above can be found in [8]. Few recent approaches proposed combining several feature selection techniques in order to improve results. In [9] different feature subsets selection methods and pair-wise methods, as well as methods of combining their results were evaluated on para-linguistic speaker trait data using nearest-neighbor classifier with Euclidean distance. It was found that different methods perform best on different tasks but no inside was given about how the underlying assumptions of each method or of the classifier used, contribute to the results. In [10] PCA, Genetic Algorithms and decision trees where combined and tested by training and predicting stock market data using ANNs as classifiers. This study was also purely empirical and no insides were provided as to why to choose these particular three methods apart from the fact that they perform well individually on certain tasks. Other strategies for combining feature selection methods proposed a voting systems of feature selection procedures or a combinatorial approach i.e. combining a set of methods in all possible ways for finding which combination performs best [11].

In this paper we firstly study the effect of certain classes of model assumptions when used to select features from data generated by different response function shapes. For example, we are interested in what happens when the assumed model is linear and the underlying data generating function contains interactions or vice-versa. We further show through our empirical results that one possible approach for improving feature selection sensitivity is by joining the sets of variables identified by different methods, relying on different assumptions. Lastly, we show that feature selection remains an important step at least for the cases in which $n << p$ when training DNNs.

The paper is further organized as follows: in Section 2 we formally define the feature selection problem and review the classes of methods used based on the assumptions they make about the underlying function that generated the data and our hypotheses. We then describe our method, the data and the implementation details we have used to test our hypotheses. In Section 3 we discuss our results and in Section 4 we provide our conclusions.

2 Materials and Methods

To formally describe the feature selection problem when given data for a single response variable, we consider the following sets of variables: $X^G = \{X_1^G, X_2^G, ..X_p^G\}$ and Y are the given random variables in our data-set and $X^A = \{X_1^A, X_2^A, ..\}$ is the set of variables associated with the response Y, which may or may not be present in our data-set. We denote by $y^i, x_1^i, ..x_p^i$ the values of the i^{th} data-set unit and we consider we have n such units. We assume the data-set units to be i.i.d.. The goal of non-causal variable selection is to identify the elements of $X = X^G \cap X^A$. In most variable selection approaches it is also considered that for two variables $\{X_i, X_j\} \in X^G \cap X^A$, X_i is more important than X_j if it explains more variability in the response Y. In order to encompass all assumptions made by a large body of variable selection methods either implicitly or explicitly about the regression function linking the predicted variable to the predictors, we consider the following model in which y^i (or a transformation of it) can be predicted as a combination of linear, non-linear and interacting terms depending on the covariates x^i:

$$y^i = f(x^i) = \beta_0 + \beta x^i + \sum_j \alpha_j g_j(x_1^i, x_2^i..x_p^i) + \sum_k \gamma_k h_k(x_1^i, x_2^i..x_p^i) + \varepsilon_i \qquad (1)$$

where $\beta = \{\beta_1, ..\beta_p\}$ and $h_k(X)$ are multivariate linear interaction terms of subsets of X. We define the interaction terms as follows: two variables X_p and $X_q \in X$ interact if their effect on the response Y is non-additive [12], that is Y cannot be expressed as $Y = f_1(X_i) + f_2(X_j)$. Higher order interactions between more than two variables can be defined in a similar manner. $g_j(X)$ are non-linear terms of some variables $X_i \in X$, other than interactions of the form $h_k(X)$, which in turn may have their own parameters used to define the exact shape of the function. The error term ε_i, at general level may or may not be dependent on the variables $X_i \in X$.

Variable selection methods have this far been classified into filters, wrappers and embedded methods [13] [14]. Filters are considered prediction model independent, such as pair-wise tests, wrappers are methods which select subsets independently and then these are evaluated against the prediction model while embedded methods have the variable selection mechanism included in the prediction algorithm. Our approach is to evaluate variable selection methods based on the simplifying statistical assumptions that they make about a general regression model as the one in Equation 1. First we note that non-parametric methods do not attempt to find information about the sparsity and values of the coefficients β, α or γ or the coefficients inside the g functions when these are explicitly expressed. However, these methods may implicitly make assumptions regarding the relationship between Y and the possible predictors. For example, non-parametric pair-wise tests such as Spearman [15] or Maximal Information Coefficient (MIC) [16] aiming at detecting linear and non-linear dependencies between two variables, implicitly assume that the some coefficients β_i or γ_i are large enough so that $f(X_i)$ has a function-like shape when plotted against Y, regardless of other small contributions from other terms, affecting Y. Using the general response model described above, we classify the basic assumptions of variables selection methods as follows:

1. Linearity Assumption. It is assumed that at least some variables $\beta_i X_i$ have an independent, linear effect on Y i.e. $\beta \neq \{0,,,0\}$. Under these assumptions, there are two different strategies employed in variable selection. If it is considered that some variables X_i exhibit a strong signal i.e. the X_i have a strong influence on variation in Y, it makes sense to test dependence between each X_i and Y separately. There are numerous cases in data mining applications falling into this class of assumptions. For example, in bioinformmatics it is often assumed that a disease is mainly caused by a particular gene variant independently and other potential factors may have small contributions to it. Pairwise correlation tests statistics such as Pearson Correlation Test [17] are used under this assumption. If it is assumed that multiple cumulated small linear effects of the covariate variables add up to influence the value of the response, the effect of each variable, if tested separately, might be weak and undetectable. Methods such as Regularized Linear Regression [18] under either the frequentist approach or the Bayesian approach are widely used under this assumption.

2. Interaction Assumption. It is assumed that some variables X_i may statistically interact in their effect on Y i.e. $\sum_i \gamma_i h_i(X) \neq 0$. Examples here are Bayesian Variable Partition Models [19], [20] which partition the variables in: variables that don't have effect on the response, variables that have a linear independent effect and variables that have an effect by interacting with other variables. Markov Chain Monte Carlo (MCMC) [21] methods are used to search over the space of the possible partitions in these methods. Another example of models here are the Regularized Regression Models that include interaction terms [22] but limit the assumption to pairwise interaction. Perhaps the most commonly used models are Decision Trees [23] and Random Forests [24] which assume interaction terms only. A few methods were proposed to include the additive assumption of the interaction terms using decision trees [12].

3. Non-linearity Assumption. It is assumed that some variables X_i have an independent, non-linear effect on Y i.e. $\sum_j g_j(X) \neq 0$. Similarly to the linearity assumption we can search for variables exhibiting strong signals using pair-wise tests or we can consider multiple small non-linear effects coming from several predictors. For pairwise tests, examples here are the non-parametric correlation tests such as Spearman [15] or the mutual information based ones such as Maximal Information Coefficient (MIC) [16]. For multivariate case, some approaches perform a randomized feature selection [25] and use as evaluation a machine learning algorithm capable of capturing highly non-linear functions, such as Artificial Neural Networks (ANNs). Randomized feature selection methods such as genetic algorithms and simulated annealing, rely on a random search through the feature subsets space, each subset being evaluated on a classification or regression method of choice. However, randomized feature selection algorithms and MCMC-based approaches are either unlikely to converge or are computationally unfeasible for very large variable sets [25].

It is obvious from the classification above that some methods overlap at least partially in their assumptions. For example MIC should identify predictors which can be identified with Pearson Test but this doesn't hold the other way around. However, it is obvious that certain assumptions are disjunct: for example MIC may not identify variables of small linear pair-wise interacting effect which can be identified with a group-interaction regularized regression method. The aim of assumptions is to reduce the search space of possible explanatory models. Therefore, our first hypothesis is that a union of the variables subsets selected by different methods applicable to large datasets, for which $n << p$, will lead to a better sensitivity than using each method individually. That should be the case if we select methods with at least partially disjunct sets of assumptions. Given that the joint set of the variables selected by different methods can be used for a larger set of non-linear explanatory functions, we use as prediction model the neural network model:

$$f_k = act(b_k + W_k f_{k-1}) \tag{2}$$

where $f_0 = X^G$ is the input to the neural net, f_k (for $k > 0$) is the output of the k_{th} hidden layer, which has weight matrix W_k and offset b_k. act is the activation function of each node for example, logistic function $act(x) = 1/(1 + e^{-(b+wx)})$ or hyperbolic tan-

gent $act(x) = tanh(b + wx)$ (note that we consider the same activation function for all nodes) and for $k > 2$ layers the model above is a Deep Neural Network (DNN) model. For $k = 1$ the neural network corresponds to a linear regression or logistic regression depending on whether the activation logistic function is used or not for the output layer and therefore maps to the linear terms in our model in Equation 1. By searching through the space of possible architectures of neural networks (i.e number of layers and dimensions of the weight matrices b_k and W_k for each k), we therefore cover a large number of non-linear explanatory functions. However in the $n << p$ setting, similar to the linear and logistic regression, this model may fail to train for a high accuracy on test data because of noise accumulated from the irrelevant variables [1] even if we had the computational resources to train large architectures with a large number of input variables. That is because the neural network model has no embedded mechanism of selecting a small subset of the input variables. Hence, a variable selection method is necessary. Here, we hypothesize that a union of the variables subsets selected by different methods applicable to large datasets, for which $n << p$, will cover a large body of possible non-linear explanatory functions and will lead to a better prediction accuracy than using each method individually or not using any variable selection method when training DNN models.

2.1 Combining variable selection methods

Our approach outputs a union of the variable selection results from three different methods relying on disjoint assumptions about the regression function. We considered as selection criteria for these methods, the level of generalization in terms of data type i.e. they can be applied to both discrete and continuous response variables, the fact that they do not depend on data-specific priors and the fact that their assumptions about the underlying shape of the response function are only partially overlapping. In a similar manner other methods can be combined. We detail further below the three methods.

1. the first approach implies using a non-parametric test statistic able to detect non-linear associations here we chose MIC [16] .
2. the second approach is using lasso regression by starting with all the variables in the model and selecting the regularization parameter performing best over cross validation data [18]. This method will identify variables of large linear effect or when small linear contributions of multiple variables add up to explain variation in the response variable.
3. the third approach implies using hierarchical network lasso regression with the aim of discovering variables possibly involved in pairwise interactions [22] [26] with small or large effect on the response variable.

We further detail the three methods enumerated above.

Maximal Information Coefficient (MIC) MIC is based on the mutual information of two random variables. For the case of two continuous variables Y and X_i the mutual information is as follows:

$$I(X_i;Y) = \int_{X_i}\int_{Y} log(\frac{p(x,y)}{p(x)p(y)})dxdy$$

For discrete variables the integrals are to be replaced with summations over all the possible discrete values of the two variables. $p(x,y)$ it is either the joint probability distribution (in the case of discrete functions) or the joint probability density function (in the case of continuous variables) and $p(x)$ and $p(y)$ are the marginal probability distribution or density functions. Intuitively, the mutual information measures the information that X_i and Y share i.e. it measures how much knowing one of these variables reduces uncertainty about the other. This further implies that the more information the two variables, the more likely it is that the data points on scatterplot grid of the two variables will fall in particular boxes while others will be left empty. Based on this idea, MIC uses an heuristic search over possible ways to grid the scatterplot and find the maximum mutual information. But when the two variables share information, i.e. knowing one of them enables us to find information about the value of the other, then there is a functional relationship between the two variables. Therfore, MIC will identify variables with large linear or non-linear effect on the response however, it will lose power when there are several small linear or non-linear effects on the response, because it is a pair-wise test. MIC is thus applicable to large datasets with discrete or continuous response functions.

Linear regression with $l_1 - regularization$ or Lasso The linear model reduces the general model in Equation 1 to the following:

$$\mathbb{E}(Y|X) = \sum_{j=1}^{p} \beta_j X_j + \varepsilon_1 \tag{3}$$

subject to $\sum_{j=1}^{p} |\beta_j| \leq K$, for some positive value K. ε_1 is independent of $X_j's$ and $\varepsilon_1 \backsim N(0,\sigma^2)$. For binary response $\mathbb{E}(Y|X)$ is replaced with $logit(\mathrm{P}(Y=1|X))$. A data-dependent equivalent to the $\sum_{j=1}^{p} |\beta_j| \leq K$ constraint is actually implemented in practice. The role of it is to shrink the coefficients β_j forcing some to be 0 and thus embedding variable selection. The actual sparsity level corresponding to the size of K is usually chosen via cross-validation. This model is expected to capture the linear terms from our general model in Equation 1. As shown in Section 4 this method is prone to explaining variation coming from non-linear terms and interactions by wrongly adding irrelevant linear terms.

Group interaction Lasso The group interaction model reduces the general model in Equation 1 to the following:

$$\mathbb{E}(Y|X) = \sum_{i=1}^{p} \theta_i X_i + \sum_{i<j} \theta_{i:j} X_{i:j} + \varepsilon_2 \tag{4}$$

subject to $\sum_{j=1}^{p} |\theta_j| + \sum_{i<j} |\theta_{i:j}| \leq K$, for some positive value K. ε_2 is independent of $X_j's$ and $\varepsilon_2 \backsim N(0, \sigma^2)$. $X_{i:j}$ are interaction terms in the sense described in Section 2. For binary response $\mathbb{E}(Y|X)$ is replaced with $logit(\mathrm{P}(Y = 1|X))$.

The model above is called hierarchical if we impose the restriction of having variables appearing in the interaction terms also contributing linearly. There are two types of hierarchy: weak hierarchy and strong hierarchy. Weak hierarchy model imposes that at least one variable appearing in each interaction term to also contribute independently to the response and strong hierarchy model imposes that all the variables appearing in interaction terms also contribute independently. Similarly, to the regular lasso model, the group interaction model imposes the sparsity constraint $\sum_{j=1}^{p} |\theta_j| + \sum_{i<j} |\theta_{i:j}| \leq K$ which forces most coefficients to be 0. In practice other constraints are used to limit the number the search space of group interactions and also only pair-wise interactions are usually assumed. We note that although the model above covers both linear and interaction terms it only partially overlaps with the assumptions of the linear model in Equation 3 and that is because it might prefer to explain variation through interacting terms there where the model in Equation 3 may consider it noise or explain it through linear terms. As it is shown in the Results Section this leads the algorithms implementing the two models to cover slightly different true positives and false positives in their results, depending on the underlying shape of the response function.

2.2 Experiments

We first studied on synthetic data how the assumptions of the methods we chose affect the variable selection result and then drew conclusions about the general class of methods with similar assumptions. We then tested the first hypothesis on synthetic datasets, for which the number of samples is relatively small to the number of variables. For the second hypothesis, we used data from the UCI Machine Learning Repository and calculated the prediction accuracy of DNNs trained with variables selected by the three different methods. We then compared the results with accuracy of DNNs trained using the joint set of variables selected by all the methods. We finally compare with the accuracy of DNNs trained using the whole set of variables available.

We have implemented our simulation in *R* statistical tool [27] and used the package *glmnet* [28] for lasso. For hierarchical lasso we have used the packages *hiernet* when data had maximum 2000 variables and *glinternet* [26] for data with more than 2000 variables [22]. For training DNNs we used the neural networks package *neuralnet* [29]. For calculating MIC we have used the *MINE* application [16]. Each variable was considered identified for a $p - value$ threshold of 0.05 or for a non-zero coefficient in the case of regression methods. The regression algorithms picked the best regularization parameter using 100-fold cross-validation. For training DNNs we performed a wide parallel search over a grid of possible number of layers in the network and nodes in each layer. Our search methods stopped adding more layers and nodes in the DNN architecture when the accuracy on the cross-validation started decreasing. We then selected the model with the highest accuracy on cross-validation data. This method was applied for each dataset separately. We used the logistic function as activation function.

2.3 Synthetic Data Sets

Our first data set contained $p = 2000$ random variables with randomly generated distributions and parameters (Uniform, Gaussian, Gamma). We calculated the response Y as follows: we selected $r = 16$ relevant variables at random and generated β from a uniform distribution with $1/r$ mean and we used a normally distributed additive noise ε with mean 0 and variance $\sigma^2 = 0.1$. The first response function was linear. The second response function was generated by having half of the relevant variables contributing linearly and in pairwise interaction terms and the other half contributing only in interactions terms, such that, at least one variable from an interaction term is also involved in a linear term (i.e. weak hierarchy). The third function was generated by having half of the relevant variables contributing linearly and the other half contributing non-linearly (as sum of sine functions). Finally the forth function was calculated using non-linear contribution from the predictors, as a sum of sine functions of each of the 16 relevant variables.

2.4 UCI Machine Learning Repository Data Sets

We have selected three datasets from the UCI Machine Learning Repository: 'Dorothea' [30], 'P53 Mutants' [31] [32] [33] and 'Arcene' [30] and selected a number of features and samples in order to simulate the $p >> n$ scenario. Dorothea is a drug discovery dataset. Chemical compounds represented by structural molecular features must be classified as active (binding to thrombin) or inactive. From this dataset, we have used the first 5000 features and 300 training and cross-validation samples for our experiments and the rest of 1650 for testing. For the second dataset, the goal is to model mutant p53 proteins transcriptional activity (active vs inactive) based on data extracted from biophysical simulations. We have used the first 5000 features from this dataset, with 100, 200 and 300 respectively, training and cross-validation samples. The rest up to 3000 samples were used for testing. ARCENE's task is to distinguish cancer versus normal patterns from 5480 mass-spectrometric features. We have used the 100 samples provided for training and validation and the rest of 100 samples provided were used for testing.

3 Results and Discussion

Our results confirm our hypotheses that joining variables selected by methods relying on different classes of assumptions into a single set increases sensitivity (i.e. the number of correctly selected variables) and accuracy of DNNs.

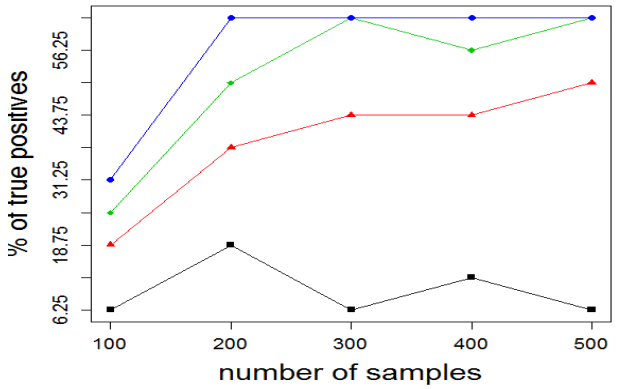

Fig. 1: Percentages of correctly identified variables for the linear ground truth response function. Black squares - MIC, Red triangles- Lasso, Green diamonds- Hierarchical Network Lasso, Blue circles- Results Union.

For the linear function, it is clear from Fig.1 that the three models capture slightly different sets of true positives, which is why the union model performs best. The model which includes interactions returns a large number of false positives due to wrongly explained variations (i.e over-fits the response function using wrong explanatory interactions terms). These false positives are then reflected in the union result as we can see in Fig.2. Although there are several univariate methods able to cope with non-linearity in variable selection, they lose power when there are several covariates of small effect in the data generating function.

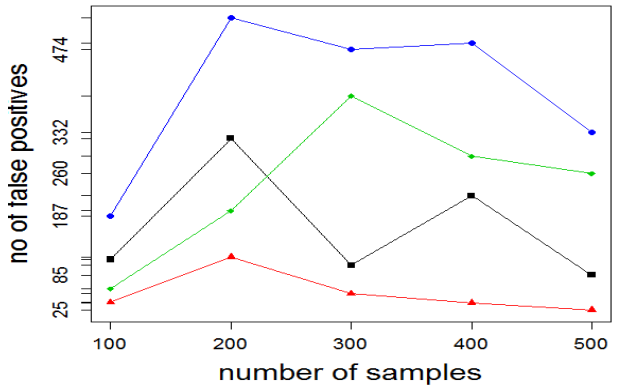

Fig. 2: Number of wrongly selected variables for the linear ground truth response function. Black squares - MIC, Red triangles- Lasso, Green diamonds- Hierarchical Network Lasso, Blue circles- Results Union.

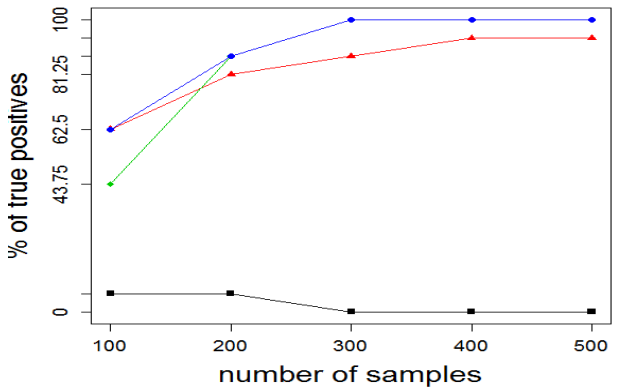

Fig. 3: Percentages of correctly identified variables for the ground truth response function with half of the variables contributing both linearly and in pairwise interaction terms and the other half of the variables appearing only in pairwise interaction terms . Black squares - MIC, Red triangles- Lasso, Green diamonds- Hierarchical Network Lasso, Blue circles- Results Union.

The multivariate linear models, such as regularized regression perform well even on large data sets. Sensitivity of univariate methods relying on non-linearity assumption, such as MIC, is well outperformed by multivariate methods such as lasso regression. In the case of the functions with half of the variables included in interactions terms only, the model with interactions captures all the true positives as seen in Fig.3. In this case the interactions model includes the results of all the other models as the number of samples increases.

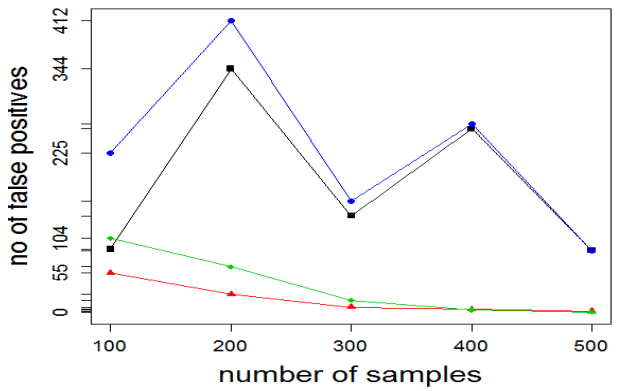

Fig. 4: Number of wrongly selected variables for the ground truth response function with half of the variables contributing both linearly and in pairwise interaction terms and the other half of the variables appearing only in pairwise interaction terms . Black squares - MIC, Red triangles- Lasso, Green diamonds- Hierarchical Network Lasso, Blue circles- Results Union.

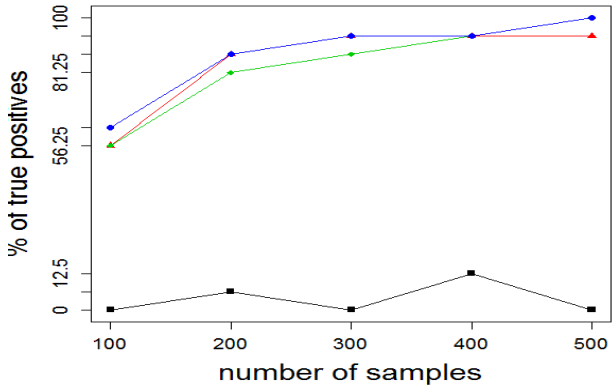

Fig. 5: Percentages of correctly identified variables for the ground truth response function with half of the variables contributing linearly and the other half of the variables appearing only in non-linear terms .Black squares - MIC, Red triangles- Lasso, Green diamonds- Hierarchical Network Lasso, Blue circles- Results Union.

The explanation here is that the variation generated by the interaction terms in the ground-truth function is explained by the wrong variables in the linear model.

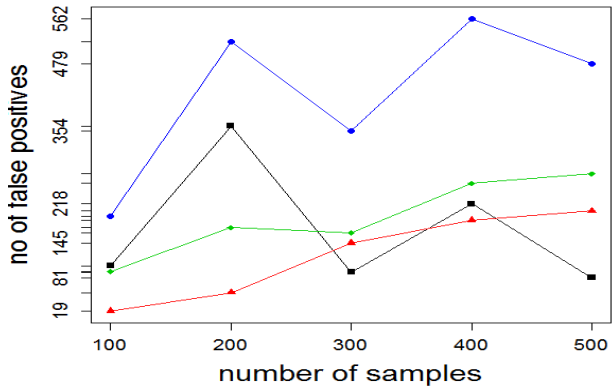

Fig. 6: Number of wrongly selected variables for the ground truth response function with half of the variables contributing linearly and the other half of the variables appearing only in non-linear terms . Black squares - MIC, Red triangles- Lasso, Green diamonds- Hierarchical Network Lasso, Blue circles- Results Union.

As we tested our non-linear functions with different number of samples, we also observed that, methods with embedded variable selection, like lasso and the hierarchical network, tend to increase the number of variables included in the best fitted model, with the increase in the number of samples, as we can see in Fig.5-8. This makes sense, as

when the assumed shape of the function is wrong (e.g. lasso assumed a linear effect on the response while our function is non-linear), with each new sample included in the model comes the tendency of including more variables to explain for the extra sample variation. This tendency is given by the fact that the assumed function shape is wrong and therefore fails to explain variation in multiple samples with the same variables. As a consequence, a larger number of variables in the model out of the total number of variables in the data set, resulted in increased number of both false positives and correctly identified variables.

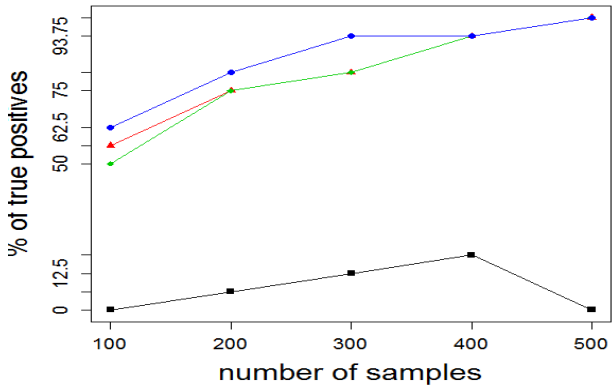

Fig. 7: Percentages of correctly identified variables for the non-linear ground truth response function. Black squares - MIC, Red triangles- Lasso, Green diamonds- Hierarchical Network Lasso, Blue circles- Results Union.

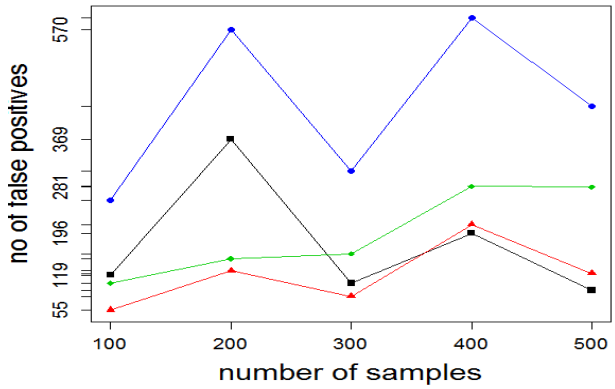

Fig. 8: Number of wrongly selected variables for the non-linear ground truth response function. Black squares - MIC, Red triangles- Lasso, Green diamonds- Hierarchical Network Lasso, Blue circles- Results Union.

Our results on the UCI Machine Learning Repository datasets Tables 1-5 suggest that variable or feature selection is useful for improving accuracy of DNNs. This step is particularly important when few samples are available, relative to the number of features. As seen in Tables 1-3 and 5-7, the results on the P53 mutants dataset and Dorothea dataset suggest that as we increase the number of samples, DNN accuracy becomes closer or the same regardless of use of feature selection prior to training. We note however that these results were obtained on two-class classification rather than regression tasks and DNNs perform best on classification.

Table 1. P53 data results - first 100 samples used for training

Method	**Acc.**	**Neural Network Architecture**
MIC + DNN	97.54%	
Lasso	98.71%	
Hierarchical Net	97.44%	
Union of features selected by the 3 methods above + DNN	99.45%	12 hidden layer with 4 nodes each
Training DNN with all features	99.33%	10 hidden layers with the following number of nodes each: 80 70 60 55 45 35 25 15 10 5

Table 2. P53 data results - first 200 samples used for training

Method	**Acc.**	**Neural Network Architecture**
MIC + DNN	99.44%	
Lasso	97.75%	
Hierarchical Net	97.75%	
Union of features selected by the 3 methods above + DNN	99.45%	12 hidden layer with 4 nodes each
Training DNN with all features	99.45%	hidden layers with the following number of nodes each: 120 115 110 105 100 95 90 85 80 75 70 65 60 55 50 45 40 35 30 25 20 15 10 5

Table 3. P53 data results - first 300 samples used for training

Method	**Acc.**	**Neural Network Architecture**
MIC + DNN	97.75%	
Lasso	99.15%	
Hierarchical Net	99.05%	
Union of features selected by the 3 methods above + DNN	99.55%	10 hidden layers with the following number of nodes each: 80 70 60 55 45 35 25 15 10 5
Training DNN with all features	99.55%	hidden layers with the following number of nodes each: 120 115 110 105 100 95 90 85 80 75 70 65 60 55 50 45 40 35 30 25 20 15 10 5

Table 4. Arcene data results

Method	**Acc.**	**Neural Network Architecture**
MIC + DNN	76%	
Lasso	64%	
Hierarchical Net	73%	
Union of features selected by the 3 methods above + DNN	76%	hidden layers with the following number of nodes each: 25,20,15,10,5
Training DNN with all features	73%	hidden layers with the following number of nodes each: 120 110 90 80 70 60 50 40 30 20 15 10 5

Table 5. Dorothea data results - first 100 samples used for training

Method	**Acc.**	**Neural Network Architecture**
MIC + DNN	78.04%	
Lasso	84.24%	
Hierarchical Net	88.22%	
Union of features selected by the 3 methods above + DNN	93.46%	hidden layers with the following number of nodes each: 110 90 80 70 60 50 40 30 20 15 10 5
Training DNN with all features	91.58%	hidden layers with the following number of nodes each: 160 140 120 110 90 80 70 60 50 40 30 20 15 10 5

Table 6. Dorothea data results - first 200 samples used for training

Method	**Acc.**	**Neural Network Architecture**
MIC + DNN	81.36%	
Lasso	87.55%	
Hierarchical Net	89.08%	
Union of features selected by the 3 methods above + DNN	94.22%	hidden layers with the following number of nodes each: 90 80 70 60 50 40 30 20 15 10 5
Training DNN with all features	93.76%	hidden layers with the following number of nodes each: 120 110 90 80 70 60 50 40 30 20 15 10 5

Table 7. Dorothea data results - first 300 samples used for training

Method	**Acc.**	**Neural Network Architecture**
MIC + DNN	90.03%	
Lasso	97.56%	
Hierarchical Net	97.02%	
Union of features selected by the 3 methods above + DNN	97.86%	hidden layers with the following number of nodes each: 120 110 90 80 70 60 50 40 30 20 15 10 5
Training DNN with all features	97.86%	hidden layers with the following number of nodes each: 160 140 120 110 90 80 70 60 50 40 30 20 15 10 5

4 Conclusions

Our first contribution in this paper was to highlight the effects of assumptions implied by different feature selection methods onto the results when the underlying function connecting the response variable to the predictors takes different shapes. Our results suggest that when the assumed model doesn't include certain types of terms (e.g. non-linear or interactions between the predictors) then wrong predictors will be selected to explain variability coming from these terms. The number of wrong predictors tends to increase as we increase the number of samples in the $n << p$ set-up, as each sample adds extra variability which needs which cannot be explained by the assumed model. One way to overcome this problem could be to keep the number of selected variables low or significantly lower than the number of samples when using regularized regression, to ensure that we reduce the number of false positives. This approach requires further research though. Our second contribution is showing through our empirical studies that one possible approach for improving feature selection sensitivity is by joining the sets of variables identified by different methods, relying on different assumptions. Future work in this direction could be classifying other methods based on the three main simplifying assumptions we presented in this paper and testing other possible combinations of feature selection methods using the same criteria of covering different classes of assumptions. Our last contribution is showing through our empirical studies that de-

spite deep learning capacity of training multivariate complex non-linear functions via feature extraction, feature selection remains an important step at least for the cases in which $n << p$. Future work here could be directed towards embedding feature selection mechanisms in neural networks architectures.

,

References

1. Fan J, Fan Y. High-dimensional classification using features annealed independence rules. Ann Statist. 2008 12;36(6):2605–2637. Available from: http://dx.doi.org/10.1214/07-AOS504.
2. Schmidhuber J. Deep Learning in Neural Networks: An Overview. CoRR. 2014;abs/1404.7828. Available from: http://arxiv.org/abs/1404.7828.
3. Nie F, Huang H, Cai X, Ding CH. Efficient and Robust Feature Selection via Joint L2,1-Norms Minimization. In: Lafferty JD, Williams CKI, Shawe-Taylor J, Zemel RS, Culotta A, editors. Advances in Neural Information Processing Systems 23. Curran Associates, Inc.; 2010. p. 1813–1821. Available from: http://papers.nips.cc/paper/3988_efficient_and_robust_feature_selection_via_joint_l21_norms_minimization.pdf.
4. Cawley GC, Talbot NL, Girolami M. Sparse Multinomial Logistic Regression via Bayesian L1 Regularisation. In: Schölkopf B, Platt JC, Hoffman T, editors. Advances in Neural Information Processing Systems 19. MIT Press; 2007. p. 209–216. Available from: http://papers.nips.cc/paper/3155_sparse_multinomial_logistic_regression_via_bayesian_l1_regularisation.pdf.
5. Ma Z, Nie F, Yang Y, Uijlings JRR, Sebe N. Web Image Annotation Via Subspace-Sparsity Collaborated Feature Selection. Multimedia, IEEE Transactions on. 2012 Aug;14(4):1021–1030.
6. Zhao Z, Liu H. Spectral Feature Selection for Supervised and Unsupervised Learning. In: Proceedings of the 24th International Conference on Machine Learning. ICML '07. New York, NY, USA: ACM; 2007. p. 1151–1157. Available from: http://doi.acm.org/10.1145/1273496.1273641.
7. Zhao Z, Wang L, Liu H. Efficient Spectral Feature Selection with Minimum Redundancy; 2010. Available from: http://www.aaai.org/ocs/index.php/AAAI/AAAI10/paper/view/1737.
8. Szepannek G, Weihs C. Local Modelling in Classification on Different Feature Subspaces. In: Proceedings of the 6th Industrial Conference on Data Mining Conference on Advances in Data Mining: Applications in Medicine, Web Mining, Marketing, Image and Signal Mining. ICDM'06. Berlin, Heidelberg: Springer-Verlag; 2006. p. 226–238. Available from: http://dx.doi.org/10.1007/11790853_18.
9. Pohjalainen J, Rsnen O, Kadioglu S. Feature selection methods and their combinations in high-dimensional classification of speaker likability, intelligibility and personality traits. Computer Speech and Language. 2015;29(1):145 – 171. Available from: http://www.sciencedirect.com/science/article/pii/S0885230813001113.
10. Tsai CF, Hsiao YC. Combining multiple feature selection methods for stock prediction: Union, intersection, and multi-intersection approaches. Decision Support Systems. 2010;50(1):258 – 269. Available from: http://www.sciencedirect.com/science/article/pii/S0167923610001521.

11. Li Yanjun HDF, M CS. Combination of Multiple Feature Selection Methods for Text Categorization by Using Combinatorial Fusion Analysis and Rank-score Characteristic. International Journal on Artificial Intelligence Tools. 2013;22(02):1350001. Available from: http://www.worldscientific.com/doi/abs/10.1142/S0218213013500012.
12. Sorokina D, Caruana R, Riedewald M, Fink D. Detecting Statistical Interactions with Additive Groves of Trees. In: Proceedings of the 25th International Conference on Machine Learning. ICML '08. New York, NY, USA: ACM; 2008. p. 1000–1007. Available from: http://doi.acm.org/10.1145/1390156.1390282.
13. Tang J, Alelyani S, Liu H. Feature Selection for Classification: A Review;.
14. Saeys Y, Inza I, Larraaga P. A review of feature selection techniques in bioinformatics. Bioinformatics. 2007;23(19):2507–2517. Available from: http://bioinformatics.oxfordjournals.org/content/23/19/2507.abstract.
15. Pirie W. Spearman rank correlation coefficient. Encyclopedia of statistical sciences. 1988;.
16. Reshef DN, Reshef YA, Finucane HK, Grossman SR, McVean G, Turnbaugh PJ, et al. Detecting Novel Associations in Large Data Sets. Science. 2011;334(6062):1518–1524.
17. Sedgwick P, et al. Pearsons correlation coefficient. BMJ. 2012;345.
18. Tibshirani R. Regression Shrinkage and Selection Via the Lasso. Journal of the Royal Statistical Society, Series B. 1996;58:267–288.
19. Zhang Y, Jiang B, Zhu J, Liu JS. Bayesian Models for Detecting Epistatic Interactions from Genetic Data. Annals of Human Genetics. 2011;75(1):183–193. Available from: http://dx.doi.org/10.1111/j.1469-1809.2010.00621.x.
20. Zhang Y. A novel bayesian graphical model for genome-wide multi-SNP association mapping. Genetic Epidemiology. 2012;36(1):36–47. Available from: http://dx.doi.org/10.1002/gepi.20661.
21. Gilks WR. Markov chain monte carlo. Wiley Online Library; 2005.
22. Bien J, Taylor J, Tibshirani R. A lasso for hierarchical interactions. Ann Statist. 2013 06;41(3):1111–1141. Available from: http://dx.doi.org/10.1214/13-AOS1096.
23. Lior Rokach OM. Data Mining with Decision Trees. vol. 69; 2007.
24. Breiman L. Random Forests. Machine Learning. 2001;45(1):5–32. Available from: http://dx.doi.org/10.1023/A%3A1010933404324.
25. Stracuzzi DJ.) Randomized Feature Selection in: Computational Methods of Feature Selection; 2007.
26. Michael Lim TH. Learning interactions through hierarchical group-lasso regularization. 2013;Available from: http://arxiv.org/abs/1308.2719.
27. R Core Team. R: A Language and Environment for Statistical Computing. Vienna, Austria; 2014. Available from: http://www.R-project.org/.
28. Friedman J, Hastie T, Tibshirani R. Regularization Paths for Generalized Linear Models via Coordinate Descent. Journal of Statistical Software. 2010;33(1):1–22. Available from: http://www.jstatsoft.org/v33/i01/.
29. Günther F, Fritsch S. neuralnet: Training of neural networks. The R Journal. 2010;2(1):30–38.
30. Guyon I, Gunn SR, Ben-Hur A, Dror G. Result analysis of the NIPS 2003 feature selection challenge. NIPS. 2004;.
31. Danziger SA, Baronio R, Ho L, Hall L, Salmon K, Hatfield GW, et al. Predicting Positive p53 Cancer Rescue Regions Using Most Informative Positive (MIP) Active Learning. PLoS Comput Biol. 2009 09;5(9):e1000498. Available from: http://dx.doi.org/10.1371%2Fjournal.pcbi.1000498.
32. Danziger SA, Zeng J, Wang Y, Brachmann RK, Lathrop RH. Choosing where to look next in a mutation sequence space: Active Learning of informative p53 cancer rescue mutants. Bioinformatics. 2007;23(13):i104–i114.

33. Danziger SA, Swamidass SJ, Zeng J, Dearth LR, Lu Q, Brachmann RK, et al. Functional census of mutation sequence spaces: The example of p53 cancer rescue mutants. IEEE/ACM Transactions on Computational Biology and Bioinformatics. 2006;3:2006.