AN OPTIMIZATION-BASED METHOD FOR FEATURE RANKING IN NONLINEAR REGRESSION PROBLEMS

LUCA BRAVI, VERONICA PICCIALLI AND MARCO SCIANDRONE

Abstract. In this work we consider the feature ranking problem where, given a set of training instances, the task is to associate a score to the features in order to assess their relevance. Feature ranking is a very important tool for decision support systems, and may be used as an auxiliary step of feature selection to reduce the high dimensionality of real-world data. We focus on regression problems by assuming that the process underlying the generated data can be approximated by a continuous function (for instance, a feedforward neural network). We formally state the notion of relevance of a feature by introducing a minimum zero-norm inversion problem of a neural network, which is a nonsmooth, constrained optimization problem. We employ a concave approximation of the zero-norm function and we define a smooth, global optimization problem to be solved in order to assess the relevance of the features. We present the new feature ranking method based on the solution of instances of the global optimization problem depending on the available training data. Computational experiments both on artificial and real data sets are performed. The obtained results and the comparison with existing methods show the effectiveness of the proposed feature ranking method.

Keywords: Feature ranking; inversion of a neural network; concave approximation of the zero-norm function; global optimization.

1. Introduction

In supervised machine learning a set of training instances is available, where each instance is defined by a vector of features and a label. Classification and regression are learning techniques to build predictive models using the available training data. In a classification task the labels belong to a finite set, while in a regression task the labels are continuous.

Reducing the high dimensionality of real-world data has become increasingly important in machine learning and, in this context, feature selection plays a crucial role. The problem of feature selection is that of determining a subset of features in such a way that the accuracy of the predictive model built on the training data, containing only the selected features, is maximal. The main benefits of feature selection are the improvement of the

L.Bravi and M. Sciandrone are with Dipartimento di Ingegneria dell'Informazione, Università di Firenze, Italy, (l.bravi@unifi.it, marco.sciandrone@unifi.it), V. Piccialli is with Dipartimento di Ingegneria Civile e Ingegneria Informatica, Università di Roma "Tor Vergata", Italy (veronica.piccialli@uniroma2.it).

prediction performance, the reduction of memory requirements and of training and testing time, a better understanding of the phenomenon underlying the generated data. This latter issue can be of great interest in important fields such as medicine and biology. A valuable survey on feature selection methods can be found in [10].

In this work we consider regression tasks (for the reasons explained in Section 2) and we focus on the feature ranking problem which consists in computing a score for each feature (a high score indicates that the feature is highly "relevant"). For several applications (see, e.g., [1], [2], [7]), a ranking of the most relevant features can be a very important tool for decision support systems. Indeed, feature ranking assesses the role of each feature in the decision process and may help the decisor to investigate the relevance of each feature. Feature ranking can be used as a preprocessing or an auxiliary step of feature selection and is not necessarily used to define learning models.

The two main classes of feature ranking methods are the following:

- **-:** Wrapper methods, which use a machine learning model as a black box to score the individual features;
- **-:** Filter methods, which use a scoring function of the features based on general characteristics of the training data and independent of a given learning algorithm.

In a wrapper approach, a ranking can be obtained, for instance, according to goodness of fit, obtained by a given machine learning model, of individual features. Correlation criteria are often used in filter-based approaches. In most of existing algorithms the features are processed individually for computing the associated score. However, this can be a drawback since, as pointed out in [10], "a variable that is completely useless by itself can provide a significant improvement when taken with others". Therefore, we are interested in developing a ranking method that assesses the relevance of the features processing them simultaneously. The training data are used, first, to build an analytical model of the process using all the features. Then, the training set and the built mathematical model are employed together to measure the "relevance" of the features. In both the two phases the original features are considered simultaneously. We briefly present the approach that leads to formally introduce a notion of relevance and to adopt sophisticated nonlinear optimization tools.

Let us assume given the training set

$$TS = \{(x^p, y^p) : \ x^p \in R^n, \ y^p \in R, \ p = 1, \ldots, P\},$$

which can be viewed as a set of samples of an unknown function $f : R^n \to R$. We are interested in analyzing the "relevance" of the variables of the unknown, underlying function. To this aim, we define a machine learning model $F(., w) : R^n \to R$, say a neural network, approximating the function f, where $w \in R^m$ is the vector of adjustable parameters determined by the training process. Given a training pattern x^p, a training label y^q, with

$p \neq q$, we can introduce the notion of *relevance* of a variable i w.r.t. the pair (x^p, y^q). Roughly speaking, we say that i is relevant if, starting from the point x^p and modifying the minimum number of components of x^p, we determine an input $x^\star$, which yields the desired output y^q, say $F(x^\star, w) = y^q$, and is such that i is one of the modified components, say $x_i^\star \neq x_i^p$. Whenever a variable i is relevant, its score is increased, so that, by varying the pair (x^p, y^q) among the training data, we obtain the score of all the variables, and hence, the features ranking. Note that the computation of the relevant features w.r.t. a given pair (x^p, y^q) leads to tht *minimum zero-norm inversion* problem of the neural network $F(., w)$, which is a difficult nonsmooth optimization problem. Suitable modifications based on concave programming are introduced to make the problem smooth, a standard penalty approach is adopted to properly manage the nonlinear equalities, thus obtaining a smooth global optimization problem with "simple" box constraints (associated to the input data).

The attempt of the paper is that of providing a feature ranking technique starting from a formal notion of relevance of a variable. We remark that there are several definitions in the literature for what it means for features to be relevant (see, e.g., [4], [22]). The peculiarity of our approach is to formally state the notion in terms of a well-defined optimization problem involving both the mathematical model underlying the data and the training instances. The paper is organized as follows. In Section 2 the notion of relevant feature is introduced. It involves the problem of inverting a feedforward neural network and, in particular, we consider the inversion nearest (according to the zero-norm) to the reference point. In Section 3 we recall a known approach for transforming a zero-norm minimizing problem into a smooth, concave optimization problem. The new feature ranking method, involving the solution of smooth, global optimization problems, is presented in Section 4. The results of computational experiments and the comparison with existing algorithms are shown in Section 5.

2. Relevant features and inversion of a neural network

A Feedforward Neural Network (FNN) is a nonlinear mapping from the input space to the output space. A FNN consists of the input layer, one (for simplicity) hidden layer, and the output layer. The dimension of the input and output layers is fixed and depends on the considered input-output relationship. The number of hidden units and the weights connecting the layers are determined by the training process involving the minimization of an error function on the training data. The most popular FNNs are the Multi-Layer Perceptron (MLP) and the Radial Basis Function (RBF) networks, which differ in the activation function of the hidden units. Let $x \in R^n$ be the input of a FNN, $w \in R^m$ be the adjustable parameters determined by the training process. A trained FNN can be viewed as a forward mapping $y = F(x; w)$, where $F : R^n \to R$ (for sake of simpilicity we

consider one dimensional outputs) is a continuously differentiable function. FNNs are *universal approximators*, i.e., they are capable of approximating any continuous function on a compact set [18]. We want to exploit this important theoretical property of FNNs and for this reason we focus on regression problems.

The problem of *inverting* a FNN consist in determining an input $\bar{x}$ which yields a desired output $\bar{y}$. The inverse problem is an ill-posed problem since the inverse mapping is usually a one-to-many mapping.

Following [15], the problem of inverting a trained FNN can be formulated as follows:

$$\begin{aligned} &\min_x \quad f(x) \\ &s.t. \quad F(x;w) - \bar{y} = 0 \\ &\qquad l \leq x \leq u, \end{aligned} \tag{1}$$

where $l, u \in R^n$ are the lower and upper bounds on the variables, $f : R^n \to R$ is the objective function defining the kind of inversion we are going to compute. For instance, taking $f(x) = \|x-c\|^2$, where $c \in R^n$ is the *reference point*, a solution of (1) is the *inversion nearest to the reference point.*

We consider here the following formulation

$$\begin{aligned} &\min_x \quad f(x) = \|x - c\|_0 \\ &s.t. \quad F(x;w) - \bar{y} = 0 \\ &\qquad l \leq x \leq u, \end{aligned} \tag{2}$$

where $\|\cdot\|_0$ is the *zero-norm*, i.e., $\|z\|_0 = card\{i : z_i \neq 0, i = 1, \ldots, n\}$, $c \in R^n$ is the reference point.

Let $X^\star$ be the set of solutions of (2). We say that a feature $i \in \{1, \ldots, n\}$ is *relevant w.r.t. the input-output pair* $(c, \bar{y})$ if there exists a vector $x^\star \in X^\star$ such that $x_i^\star - c_i^\star \neq 0$.

In practice, we may expect to determine a finite subset $\tilde{X} \subset X^\star$ and to define, for each feature i, the set

$$\tilde{X}_i = \left\{ x \in \tilde{X} : \; x_i - c_i \neq 0 \right\},$$

to be used for computing a score $r_i(c, \bar{y})$, which can be viewed as a measure of the relevance of the feature i with respect to the pair $(c, \bar{y})$. For instance, $r_i(c, \bar{y})$ could be the cardinality of the set $\tilde{X}_i$. As we will see in the next section, by varying the pair $(c, \bar{y})$ and by computing the corresponding scores $r_i(c, \bar{y})$, for $i = 1, \ldots, N$, we can perform an overall ranking of the features.

Finally, we observe that, from an optimization point of view, problem (2) presents the following difficulties:

(i): the objective function is discontinuous;
(ii): there is a highly nonlinear constraint.

Concerning point (i), we will adopt a concave-optimization based approach [16], [19]. As regards point (ii), the constraint will be handled by adding a quadratic penalty term to the objective function related to the violation of the constraint.

3. Concave programming for minimizing the zero norm

In this section, we recall an approach for transforming a zero-norm minimization problem, which is nonsmooth, into an equivalent (in some cases) smooth concave optimization problem. Let us consider the problem

$$\begin{array}{l} \min_{x \in R^n} \|x\|_0 \\ x \in S \end{array} \tag{3}$$

where $S \subset R^n$.

In order to illustrate the idea underlying the concave approach, we observe that the objective function of problem (3) can be written as follows:

$$\begin{array}{l} \min_{x \in R^n} \sum_{i=1}^{n} s(|x_i|) \\ \\ x \in S. \end{array} \tag{4}$$

where $s : R \to R^+$ is the *step function* such that $s(t) = 1$ for $t > 0$ and $s(t) = 0$ for $t = 0$. The approach was originally proposed in [16] and is based on the idea of replacing the discontinuous step function by a continuously differentiable concave function $1-e^{-\alpha t}$, with $\alpha > 0$, thus obtaining a problem of the form

$$\begin{array}{ll} \min\limits_{x,z} & \sum_{i=1}^{n} \left(1 - \exp\left(-\alpha z_i\right)\right) \\ \\ & -z_i \leq x_i \leq z_i \qquad i = 1, \ldots, n \\ \\ & x \in S, \end{array} \tag{5}$$

It has been shown in [16] that, by assuming that S is a polyhedral set, the approximating problem (5) is equivalent to the given nonsmooth problem (4), that is, for α sufficiently large, there exists a solution of (5) which yields a solution of the original problem (4). A similar concave optimization-based approach has been proposed in [24], where the idea is that of using other smooth functions to approximate the step function. We remark that the formal equivalence between a smooth concave problem of the form (5) with the original zero norm problem (3) has been proved under suitable assumptions on the feasible set S. These assumptions are not satisfied by the feasible set of problem (2). However, we will adopt in the sequel the concave-based approach just described as heuristic to manage the zero-norm function in problem (2).

Finally, we remark that the nature of the features influences the structure of the feasible set S. In particular, if a feature is discrete or categorical, the

problem becomes more complicated, and the solution algorithm needs to tackle the integer nature of some variables. For this reason, in this paper we consider only continuous features, and leave the handling of discrete features as subject of future work.

4. The feature ranking method

In order to solve problem (2), we relax the constraint on the desired output value and introduce a penalization term in the objective function, and substitute the zero-norm with a concave function as described in the preceding section, getting the following global optimization problem:

$$\begin{aligned} \min_{x,z} \quad & \textstyle\sum_{i=1}^{n} \left(1 - \exp\left(-\alpha z_i\right)\right) + \frac{1}{2} C \left(F(x;w) - \bar{y}\right)^2 \\ & -z_i \leq x_i - c_i \leq z_i, \quad i = 1, \ldots, n \\ & l \leq x \leq u, \end{aligned} \tag{6}$$

Problem (6) is a nonconvex global optimization problem. Let $X^\star$ be the finite set of *putative global minima* found by a global minimization algorithm. Note that, in general, we can not guarantee that a global solution of (6) is neither a feasible point nor a solution of (2). We consider the subset of *quasi-feasible minima*, that is, the set $\hat{X} \subseteq X^\star$ such that

$$|F(x^\star; w) - \bar{y}| \leq \epsilon \qquad \forall x^\star \in \hat{X}, \tag{7}$$

where $\epsilon > 0$ is a given tolerance parameter. Then, we introduce the set $\tilde{X} \subset \hat{X}$ of *feasible zero-norm minima* defined as

$$\tilde{X} = \left\{ x^\star \in \arg\min_{x \in \hat{X}} \|x - c\|_0 \right\}. \tag{8}$$

Finally, for each feature i, we define the set $\tilde{X}_i$ of *minima reached by feature* i, that is

$$\tilde{X}_i = \left\{ x \in \tilde{X} : \; x_i - c_i \neq 0 \right\}, \tag{9}$$

and we compute the score

$$r_i(c, \bar{y}) = |\tilde{X}_i|. \tag{10}$$

In order to get the overall ranking of the features we vary the input-output pair $(c, \bar{y})$. In particular, we choose input-output pairs (x^p, y^q) from the training set with $p \neq q$.

This choice ensures that we are selecting output values that are "reachable", that is, by assuming that the model $F(x; w)$ is reliable, there exists at least one quasi-feasible point, i.e., the point x^q, such that

$$|F(x^q; w) - y^q| \leq \epsilon.$$

Below we report a summary of the Concave-Optimization BASed (COBAS) algorithm.

:

S0: Set $r_i = 0$ for $i = 1, \ldots, n$.

S1 : Select randomly M pairs (x^p, y^q) with $p \neq q$ and $p, q \in \{1, \ldots, P\}$. For each pair (p, q):

: Set $c = x^p$ and $\bar{y} = y^q$ in problem (6), compute the set $X^\star$ of *putative global minima* by a global minimization algorithm;

: Define the set $\hat{X}$ of *quasi-feasible minima* by (7), and the set $\tilde{X}$ of *feasible zero-norm minima* by (8).

: For each *feature* $i \in \{1, \ldots, n\}$, let $\tilde{X}_i$ the set of *minima reached by feature* i defined by (9), compute $r_i(x^p, y^q)$ by (10), and set $r_i = r_i + r_i(x^p, y^q)$.

S2: Build the ranking by ordering the feature with respect to the numbers r_i, $i = 1, \ldots, n$.

5. Computational experiments

In this section we experimentally evaluate the performance of the proposed ranking method. We consider both artificial and real datasets inherited from the literature. The experiments on artificial data sets allow us to assess the effectiveness of the method in detecting the features correlated with the output. The results on real data sets are used to evaluate the quality of the ranking provided by the proposed method.

5.1. **Implementation details.** Concerning the machine learning model $F(x; w)$, in all the experiments we adopt a RBF network with inverse multiquadric as activation function. More specifically, the neural network model is

$$F(x; w) = \sum_{i=1}^{h} \lambda_i \left(\|x - v_i\|^2 + \sigma^2 \right)^{-1/2},$$

here h is the number of hidden neurons, $\lambda_i \in R$, $v_i \in R^n$, with $i = 1, \ldots, h$, define the vector $w \in R^{(1+n)h}$ of adjustable parameters, and $\sigma^2 = 0.1$. The training of the RBF network has been formulated as the problem of minimizing a standard least-squares error function that measures the error on the outputs, in correspondence to a given set of training pairs. The training has been performed by a gradient-based batch strategy with early stopping, and the number h of hidden neurons has been determined by a cross validation technique [3].

The parameters of COBAS algorithm have been chosen as follows: the number M of randomly selected training pairs has been set equal to 500; the parameter ϵ defining *quasi-feasible* minima (see (7)) has been set equal to 10^{-4}. As global optimization algorithm applied to problem (6) we have employed a multistart algorithm using 1000 starting points, and MINOS 5.51 [17] as local solver.

5.2. **Results on synthetic data.** Five synthetic data sets have been used to analyse the performances of the proposed method to detect the features correlated with the output. The data sets have been generated as follows. Given the interval $[-a, a]^n$, we sampled P training points $x \in R^n$ randomly drawn from the uniform distribution. The output labels are computed using a regression function $f(x)$ that depends nonlinearly only on some variables x_j, with $j \in J \subset \{1, \ldots, n\}$. The analytic expression of the considered functions are the following

$$f_1(x) = (x_1^4 - x_1^2) \cdot (3 + x_2) \tag{11}$$

$$f_2(x) = 2(x_1^3 - x_1) \cdot (2x_2 - 1)(x_2 + 1) + (x_2^3 - x_2 + 3) \tag{12}$$

$$f_3(x) = -2(2x_1^2 - 1) \cdot (x_2) \cdot e^{-x_1^2 - x_2^2} \tag{13}$$

$$f_4(x) = x_1 + (x_2 > 0.5)(x_3 > 0.5) \tag{14}$$

$$f_5(x) = 10 \sin(x_1) x_2 + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5 \tag{15}$$

Concerning function f_4, the term $(x_i > 0.5)$ is equal to 1 when $x_i > 0.5$ and is equal to 0 otherwise. The interval $[-a, a]^n$ is $[-3, +3]$ for $f_1,\ f_2$, $[-1, +1]$ for f_3, and $[0, 1]$ for $f_4,\ f_5$. Functions (11), (12) and (13) were originally proposed in [21], while functions (14) and (15) draw inspiration from the ones used in [9].

We performed several tests by varying both the number $n \in \{6, 15\}$ of features and the number $P \in \{100, 200, 500, 1000\}$ of available training samples used to build the RBF network.

Let R_t be the number of features correlated with the output (relevant features), and let I_t be the number of features uncorrelated with the output (irrelevant features). A feature is selected by the method as relevant feature provided that its score is in the first R_t positions, otherwise the feature is selected as irrelevant feature. We indicate by R_s and I_s the number of *true* relevant features and the number of *true* irrelevant features, respectively, selected by the method as relevant features. We adopted the following performance index (defined in [5]):

$$Suc. = \left[\frac{R_s}{R_t} - \alpha \frac{I_s}{I_t} \right] \times 100 \tag{16}$$

This index is a measure that attempts to reward the selection of relevant features and to penalize the inclusion of irrelevant ones. The parameter α weights the penalties of choosing an irrelevant features and missing a relevant one, and is defined as $\min\{\frac{1}{2}, \frac{R_t}{I_t}\}$. If all the relevant features are ranked before irrelevant features the index of success is 100.

To compare our method with current techniques, we consider the implementation of the feature ranking method RReliefF ([20]) that is available through the package AttributeSelection of the software WEKA ([11]). The obtained results are reported in Tables (1-5) and, for each test, averages are taken over ten different training sets.

	COBAS	RReliefF	COBAS	RReliefF
n. training data	6 features		15 features	
100	100.000	98.462	97.500	93.846
200	100.000	97.500	100.000	79.231
500	100.000	100.000	100.000	97.692
1000	100.000	100.000	100.000	100.000

TABLE 1. Comparison on function f_1

	COBAS	RReliefF	COBAS	RReliefF
n. training data	6 features		15 features	
100	95.000	95.000	82.308	97.692
200	100.000	97.500	94.615	96.923
500	100.000	100.000	100.000	100.000
1000	100.000	100.000	100.000	100.000

TABLE 2. Comparison on function f_2

	COBAS	RReliefF	COBAS	RReliefF
n. training data	6 features		15 features	
100	100.000	100.000	43.846	100.000
200	100.000	100.000	86.923	100.000
500	100.000	100.000	100.000	100.000
1000	100.000	100.000	100.000	100.000

TABLE 3. Comparison on function f_3

	COBAS	RReliefF	COBAS	RReliefF
n. training data	6 features		15 features	
100	100.000	100.000	100.000	95.833
200	100.000	100.000	100.000	100.000
500	100.000	100.000	100.000	100.000
1000	100.000	100.000	100.000	100.000

TABLE 4. Comparison on function f_4

From the results reported in Tables (1-5) we get that COBAS algorithm is competitive with RReliefF algorithm in distinguishing relevant and irrelevant features. The results of Tables 2 and 3 (and other results not here reported) show that COBAS algorithm needs a sufficient number of training samples to have a reliable analytical model, and this represents a fundamental issue for the proposed feature ranking strategy.

This can be observed, for instance, in the columns corresponding to $n = 15$ features of Tables 2 and 3, where the performance index *Suc.* relative to COBAS increases with the number of training samples.

	COBAS	RReliefF	COBAS	RReliefF
n. training data	6 features		15 features	
100	100.000	100.000	100.000	85.000
200	100.000	100.000	100.000	88.000
500	100.000	100.000	100.000	100.000
1000	100.000	100.000	100.000	100.000

TABLE 5. Comparison on function f_5

Data set name	N. of training data	N. of features
Poland	1370	30
diabetes	442	10
Santa Fe laser	10081	12
abalone	4177	8
housing	506	13
cpusmall	8192	12

TABLE 6. Information on real datasets

5.3. **Results on real datasets.** For our experiments we have considered the following datasets

(1) Poland electricity load dataset ([13, 14])
(2) diabetes dataset ([8])
(3) Santa Fe laser dataset ([12, 23])
(4) housing dataset
(5) abalone dataset
(6) cpusmall dataset.

The last three data sets have been downloaded from `http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets`. The first three data sets are the ones used in [9], with the exception of the anthrokids data set that for confidentiality reasons could not be made available. We note that in case of Poland and Santa-Fe, the datasets that are available for download are a bit different: those are the raw time series. The datasets we use were converted in [9] into regression datasets by using the last q values ($q = 30$ for Poland and $q = 12$ for Santa Fe) as features to predict the current value. Information about the data sets are reported in Table 6. The quality of the feature rankings obtained by the compared methods can be evaluated using a nonlinear regressor with growing number of most important features. For each ranking method we evaluate the prediction accuracy on the test set as a function of the best features. Therefore, for every dataset a total of 70% of examples are used for training and 30% of examples for testing. This splitting has been randomly performed 10 times. As nonlinear regressor we have employed Support Vector Machines with Gaussian kernel and we have used the LIBSVM package [6]. The SVM parameters have been selected by the grid search tool provided by LIBSVM.

COBAS Algorithm has been compared again with RReliefF in terms of feature subsets and test error (averaged on ten instances randomly generated) using all the six data sets. Moreover it has been compared with

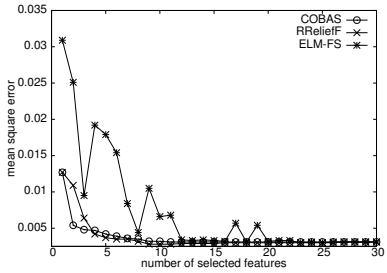

FIGURE 1. Results for the Poland dataset: mean square error for COBAS, RReliefF and ELM-FS.

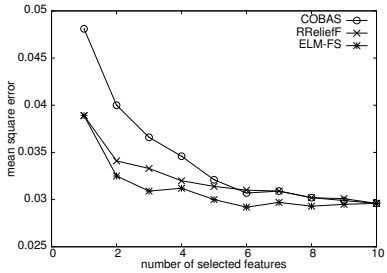

FIGURE 2. Results for the diabetes dataset: mean square error for COBAS, RReliefF and ELM-FS.

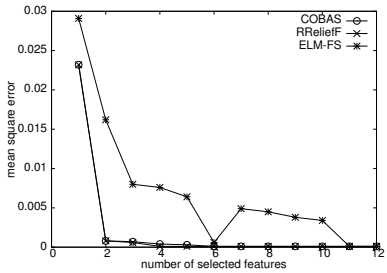

FIGURE 3. Results for the Santa Fe laser dataset: mean square error for COBAS, RReliefF and ELM-FS.

ELM-FS [9], which is a feature selection method, for the first three data sets. The code of ELM-FS is not available, so that we can perform the comparison only on the datasets used in [9], and we can not run the method on the other datasets. The results are plotted in figures (1-6), where we report the mean square error on the test set obtained by feature subsets with increasing sizes. For COBAS and RReliefF, the feature subsets are automatically defined by the overall ranking provided by the method, whereas for ELM-FS we use the subsets of selected features reported in [9]. From Figures (1-6), with the exception of Fig. 2, we can observe that COBAS either outperforms or is competitive with both ELM-FS and RReliefF, and

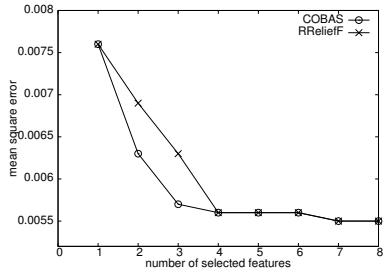

FIGURE 4. Results for the abalone dataset: mean square error for COBAS and RReliefF.

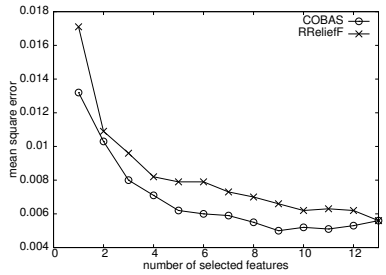

FIGURE 5. Results for the housing dataset: mean square error for COBAS and RReliefF.

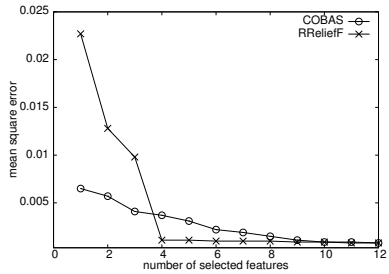

FIGURE 6. Results for the cpusmall dataset: mean square error for COBAS and RReliefF.

this shows the validity of the proposed feature ranking method. The bad performance on the diabetes dataset are due to the "low" quality of the available training data that leads to a poor fitting model. The low quality of the available data is confirmed by the fact that, as shown in Fig. 2, the test error of the nonlinear SVM regressor is of the order of 10^{-2}, whereas the test error on the other data sets is at most of the order of 10^{-3}.

In conclusion, the numerical results show the effectiveness of the proposed feature ranking method, provided that the available data permit to build a good-fitting model.

Acknowledgements

We would like to thank Benoît Frénay for providing us the three datasets Poland electricity load, diabetes, and Santa Fe laser.

References

[1] R.E. Abdel-Aal. GMDH-based feature ranking and selection for improved classification of medical data. *Journal of Biomedical Informatics*, 38:456–468 2005.

[2] T. Abeel, T. Helleputte, Y. Van de Peer and Y. Saeys. Robust biomarker identification for cancer diagnosis with ensemble feature selection methods. *Bioinformatics*, 26:392–398, 2010.

[3] C. M. Bishop. *Neural networks for pattern recognition*. Oxford: Clarendon Press., 1995.

[4] A.L. Blum and P. Langley. Selection of Relevant Features and Examples in Machine Learning. *Artificial Intelligence*, 97:245–271, 1997.

[5] V. Bolón-Canedo, N. Sánchez-Maroño and A. Alonso-Betanzos. A review of feature selection methods on synthetic data. *Knowledge and information systems*, 34:3, 483–519, 2013.

[6] C.-C. Chang and C.-J. Lin. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)* 2:3, 27:1–27:27.

[7] F. Chen and F. Li. Combination of feature selection approaches with SVM in credit scoring. *Expert Systems with Applications*, 37:4902–4909, 2010.

[8] B. Efron, T. Hastie, Trevor, I. Johnstone and R. Tibshirani. Least angle regression. *The Annals of Statistics*, 32:2, 407–499, 2004.

[9] B. Frénay, M. Van Heeswijk, M. Yoan, M. Verleysen and A. Lendasse. Feature selection for nonlinear models with extreme learning machines. *Neurocomputing*, 102:111–124, 2013.

[10] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[11] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann and I. H. Witten. The WEKA data mining software: an update. *ACM SIGKDD explorations newsletter* 11:1, 10–18, 2009.

[12] U. Hübner, N. B. Abraham, and C. O. Weiss. Dimensions and entropies of chaotic intensity pulsations in a single-mode far-infrared NH_3 laser. Physical Review A, 40, 6354–6365, 1989.

[13] A. Lendasse, J. A. Lee, V. Wertz, and M. Verleysen. Time series forecasting using CCA and kohonen maps - application to electricity consumption. ESANN'2000 proceedings, 329–334, 2000.

[14] A. Lendasse, J. A. Lee, V. Wertz, and M. Verleysen. Forecasting electricity consumption using nonlinear projection and self-organizing maps. *Neurocomputing*, 48:1-4, 299–311, 2002.s

[15] B.L. Lu, H. Kita and Y. Nishikawa. Inverting Feedforward Neural Networks Using Linear and Nonlinear Programming. *IEEE Transactions on Neural Networks*, 10:1271–1290, 1999.

[16] O.L. Mangasarian. Machine learning via polyhedral concave minimization. in *Applied Mathematics and Parallel Computing*, Festschrift for Klaus Ritter, H. Fischer, B. Riedmueller, and S. Schaeffler, eds., Physica-Verlag, Germany, 1996, pp. 175188.

[17] B. A. Murtagh and M. A. Saunders. MINOS 5.51 user's guide. *Technical Report SOL 83-20R, Stanford University, CA, USA*, Revised 1987.

[18] A. Pinkus. Approximation theory of the mlp model in neural networks. *ACTA NUMERICA*, 8:143–195, 1999.

[19] F. Rinaldi, F. Schoen and M. Sciandrone. Concave programming for minimizing the zero-norm over polyhedral sets. *Comuptational Optimization and Applications*, 46:467–486, 2010.

[20] M. Robnik-Šikonja and I. Kononenko. An adaptation of Relief for attribute estimation in regression. *Machine Learning: Proceedings of the Fourteenth International Conference (ICML)*, 296–304, 1997.

[21] L. Rosasco, M. Santoro, S. Mosci, A. Verri and S. Villa. A regularization approach to nonlinear variable selection *International Conference on Artificial Intelligence and Statistics*, 653–660, 2010.

[22] L. Yu and H. Liu. Efficient Feature Selection via Analysis of Relevance and Redundancy. *Journal of Machine Learning Research*, 5:1205–1224, 2004.

[23] A. Weigend and N. Gershenfeld. Results of the time series prediction competition at the santa fe institute *Neural Networks, 1993., IEEE International Conference on*, 3, 1786–1793, 1993.

[24] J. Weston, A. Elisseef and B. Scholkopf. Use of the zero-norm with linear models and kernel model. *Journal of Machine Learning Research*, 3:1439–1461, 2003.