Computing and Informatics, Vol. 29, 2010, 929–946

CORRECTION OF REGRESSION PREDICTIONS USING THE SECONDARY LEARNER ON THE SENSITIVITY ANALYSIS OUTPUTS

Zoran Bosnić, Igor Kononenko

University of Ljubljana
Faculty of Computer and Information Science
Tržaška cesta 25
1000 Ljubljana, Slovenia
e-mail: {zoran.bosnic, igor.kononenko}@fri.uni-lj.si

Manuscript received 11 January 2008; revised 23 March 2009
Communicated by János Fodor

Abstract. For a given regression model, each individual prediction may be more or less accurate. The average accuracy of the system cannot provide the error estimate for a single particular prediction, which could be used to correct the prediction to a more accurate value. We propose a method for correction of the regression predictions that is based on the sensitivity analysis approach. Using predictions, gained in sensitivity analysis procedure, we build a secondary regression predictor whose task is to predict the signed error of the prediction which was made using the original regression model. We test the proposed methodology using four regression models: locally weighted regression, linear regression, regression trees and neural networks. The results of our experiments indicate significant increase of prediction accuracy in more than 20 % of experiments. The favorable results prevale especially with the regression trees and neural networks, where locally weighted regression was used as a model for predicting the prediction error. In these experiments the prediction accuracy increased in 60 % of experiments with regression trees and in 50 % of experiments with neural networks, while the increase of the prediction error did not occur in any experiment.

Keywords: Regression, predictions, correction of predictions, sensitivity analysis, prediction error, prediction accuracy

Mathematics Subject Classification 2000: 68T05, 68T37, 93D25

1 INTRODUCTION

When using supervised learning for modeling data we aim to achieve the best possible prediction accuracy for the unseen examples which were not included in the learning process [1]. For evaluation of the prediction accuracies, the averaged accuracy measures are most commonly used, such as the mean squared error (MSE) and the relative mean squared error (RMSE). Although these estimates evaluate the model performance by summarizing the error contributions of all test examples, they provide no local information about the expected error of individual prediction for a given unseen example. To provide information of individual prediction accuracy/error, *reliability estimates* are of a greater use.

Reliability. In engineering, reliability is defined as the ability of a system or a component to perform its required functions under stated conditions for a specified period of time [3, 2]. In machine learning, we can define reliability as a qualitative property or ability of the system which is related to a critical performance indicator (positive or negative) of that system, such as accuracy, inaccuracy, availability, downtime rate, responsiveness, etc. Since reliability is in most cases defined qualitatively, the *reliability estimate* is therefore an estimate for quantitative measuring of reliability. According to the particular field of use, a reliability estimate can therefore be an accuracy estimate, error estimate, availability estimate, etc.

Having information about single prediction reliability available provides two important benefits in the risk-sensitive areas where acting upon predictions may have critical consequences (e.g. medical applications, stock market, navigation, control applications). First, such estimates can provide the degree of confidence in predictions, which is relevant for experts to easier decide whether to trust the prediction or not. Second, if reliability estimate is quantitatively related to the prediction error of individual examples, one could use such information to aim at correcting the prediction to the more accurate value.

Motivation. In our previous work [4] we used the sensitivity analysis technique to develop reliability estimates (estimates of prediction error in our context). We defined these estimates using repetitive modification of the learning set (expanding with an additional learning example) with the goal to observe how the prediction of a regression model for a particular example changes with respect to this influence. In such manner we obtained for each example its *initial prediction* and many *sensitivity predictions* (one for each controlled modification of the learning set), which we afterwards used to define our reliability estimates. Some of the defined reliability estimates empirically showed strong correlation to the prediction error.

Since *sensitivity predictions*, as results of the sensitivity analysis, have shown to be a good tool for prediction error estimation, in the present work we aim at discovering whether it is possible to use these predictions directly for correcting

the regression predictions. We propose an approach that beside an initial regression model (*primary model*, used for predicting) uses an additional, *secondary model*, with the task of only predicting the signed prediction error of the primary model, based on the attributes defined using the sensitivity predictions. We use the predicted correction to correct the initial predictions and compare whether we statistically gained an improvement with the correction.

The paper is organized as follows. Section 2 summarizes previous work from related areas of individual prediction reliability estimation and Section 3 summarizes our previous work which is further extended in this paper. In Section 4 we describe the proposed method and explain our experimental protocol. We describe the testing and interpret the results in Section 5. Section 6 provides conclusions and ideas for further work.

2 RELATED WORK

In the related work, most effort has been put into improving the accuracy of regression models, and very little into subsequent correction of computed predictions. For our contribution, the most relevant related work is the field of individual prediction reliability estimation, since it provides estimates of individual prediction error/accuracy, which can be utilized for the purpose of correcting the initial predictions. In the following we shortly present some important contributions in this field and provide criteria for differentiation of their conceptual foundations:

1. *Model-dependent reliability estimation* is a mechanism which is implemented as an extension of a particular predictive model. It exploits particular model characteristics (e.g. Lagrange coefficients in Support Vector Machine optimization procedure, splits in regression trees) and are as such intended for use only with those particular models. Examples:
   - Gammerman, Vovk, and Vapnik [5] and Saunders, Gammerman and Vovk [6] proposed an extension of Support Vector Machine (SVM), which produces the reliability estimates of *confidence* (probability of the correct classification) and *credibility* (1 – probability of the second most probable class) for each prediction.
   - The *confidence* values, implemented for ridge regression [7], implemented as a further work of the former approach.
   - Extension of the multi-layer perceptron with an additional output neuron, intended to predict the variance in the neighborhood of the input example, serving as reliability estimate [8].
   - *prediction intervals* for the ensembles of neural networks [9, 10]. Being defined as a degree of agreement between predicted value and example's label value, the prediction interval is therefore also an estimate of the individual prediction reliability.

2. *Model-independent reliability estimation* is an approach which uses an arbitrary model and deals with it as with the black box (wrapper-like method). This approach is more general than model-dependent approaches since it utilizes only general parameters, available in the supervised learning framework. However, its performance can depend on the properties of the used model. Examples:

   - Local modeling of prediction error based on input space properties and local learning [11, 12, 13, 14]. Most frequently the local cross validation is applied to compute the prediction and the prediction error for the example of interest using a local model.
   - Transductive methods for estimation of classification reliability [16]. Transduction is an inference principle that reasons from particular to particular [15] in contrast to inductive learning, which aims at inferring a general rule from a finite set of data. Transductive methods may therefore use only selected examples of interest and not necessarily the whole input space and aim at modeling reliability information using selected examples.
   - Use of sensitivity analysis for development of reliability estimates [4, 17] (our previous work).

The work in this paper continues our previous work [4] of using sensitivity analysis for the purpose of estimating prediction error. Instead of defining reliability estimates (our previous work), we directly use the building blocks of our former reliability estimates (i.e. sensitivity predictions) and model the prediction error as an independent regression task. We describe the relevant prerequisites from our previous work in the next section.

3 ESTIMATION OF INDIVIDUAL PREDICTION RELIABILITY USING THE LOCAL SENSITIVITY ANALYSIS

An approach which enables us to analyze the *local* particularities of learning algorithms is the *sensitivity analysis* [18, 19, 20, 21], which aims at determining how much the variation of input can influence the output of a system. The idea of putting the reliability estimation in the context of the sensitivity analysis framework is therefore to observe the changes in model outputs (i.e. predictions) by modifying its input (i.e. learning data set). Note that by influencing only the learning set and observing model outputs, the sensitivity analysis approach utilizes regression model only as its parameter and is therefore model-independent.

In the previous work [4] we defined the framework for locally modifying the learning set in a controlled manner in order to explore the sensitivity of the regression model in a particular part of the problem space. Lead by the conclusions of three related research fields we utilized the following ideas:

- the *sensitivity analysis* approach allows us to observe the behavior of the regression model as the black box and therefore enables us to design an approach that is independent of the regression model;

- approaches which generate *perturbations of learning data* (bagging [22] and boosting [23, 24, 25] are the best known in this field) do so to achieve greater prediction accuracy or study the stability of predictive models [26, 27, 28]. This motivated us to combine perturbing with the sensitivity analysis framework;
- the co-training approaches that *use unlabeled data in supervised learning* utilize the additional unlabeled learning examples by labeling them using a predictive model, which was built on labeled data, and including them into the learning set. It was also shown that the unlabeled data can be used to improve the performance of a predictor [29, 30]. Therefore it may be reasonable to use the same approach for reliability estimation of individual examples by observing the change in predictor's performance.

For influencing the input of the system (i.e. regression model) we therefore expanded the learning set with an additional learning example. By adding or removing an example from the learning set, thus making a minimal change to the input of the learning algorithm, one can expect that the change in output prediction for the modified example will also be small. Big changes in output prediction that result from making small changes in learning data may be a sign of instability in the generated model. We assume at this point that the additional example will indeed influence the prediction model to cause a change in the generated model, which is a requirement for the applied sensitivity analysis approach to work. We base this assumption on the fact that we evaluate the new prediction value in the same local neighborhood as we influence with the additional example, and provide some intuitive arguments in the following.

We expanded the learning set with additional example as follows. Let $(x, \_)$ be the unseen and unlabeled example, for which we wish to estimate the reliability of its prediction K (the *initial prediction*). K is computed using regression model M, therefore $f_M(x) = K$. Since the learning example $(x, \_)$ is unlabeled, we first had to label it prior to inserting it into the learning set. We labeled the example using its initial prediction K, which was modified by some small value of δ:

$$y = K + \delta. \tag{1}$$

In previous work we defined δ which was proportional to known bounds of label values. In particular, if the interval of learning examples' labels is denoted by $[a, b]$ and if ε denotes a value that expresses the relative portion of this interval, then $\delta = \varepsilon(b - a)$. The related works that provided ideas to expand the learning set in such manner were:

1. Using the Minimum Description Length principle [31] as a general formalism based on the probabilistic and information theory, we showed that it is possible to obtain additional information if we expand the learning data set with an additional example. We showed that *most information is achieved if we expand*

the learning set with an example that is not well covered by the initial hypothesis [4]. It is therefore advisable to use such label, which is different from the covered predictor's knowledge, i.e. the initial prediction itself (thus we modify prediction K by some chosen δ).

2. By expanding of the initial learning set with an additional example we aim at inducing a change in the output prediction for the particular example. To achieve such change, a new learning example must be positioned into a problem space in such way that it indeed affects the computation of the prediction for that particular example. This will more likely be achieved, if the new example is placed locally close to the particular example, since the local change will have greater influence to local predictions than changes made farther in the problem space. Note that it depends on the underlying regression model how the change in prediction values will propagate through the problem space – the influence may remain local or become global. However, since our approach is focused on observing the influence of local changes to local outputs, it requires the influence to be at least local, global influence thus represents no obstacle. This reasoning is illustrated in Figure 1, which illustrates the influence of the additional learning example in local area only.

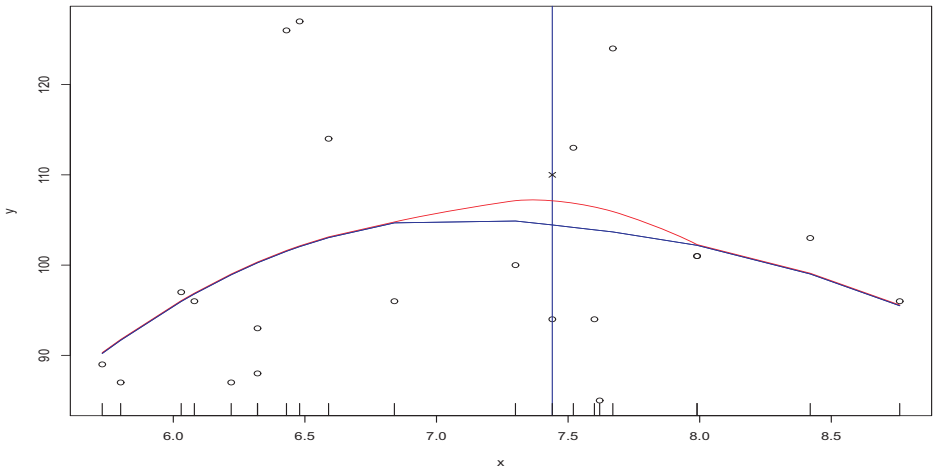

Fig. 1. The example of local impact to the change of regression model in two-dimensional problem space, where x is the only attribute. The learning examples are denoted with circles and the additional learning example is denoted with the cross. The top curve shows how the initial model (bottom curve) changes if the additional learning example is inserted into the learning set.

After selecting ε and labeling the new example, we expanded the learning data set with example (x, y). We referred to the newly built model as the *sensitivity regression model* M' and it's prediction K_ε of example $(x, \_)$ $(f_{M'}(x) = K_\varepsilon)$ as the

sensitivity prediction. Since ε is the parameter of the procedure, by selecting different values $\varepsilon_k \in \{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_m\}$ we iteratively obtained a set of sensitivity predictions

$$K_{\varepsilon_1}, K_{-\varepsilon_1}, K_{\varepsilon_2}, K_{-\varepsilon_2}, \ldots, K_{\varepsilon_m}, K_{-\varepsilon_m}. \tag{2}$$

Before computing each sensitivity prediction, we always start by the original data set; i.e., the examples are not incrementally added into the learning set, but added as the query points of interest in the problem space. The described procedure is illustrated in Figure 2.

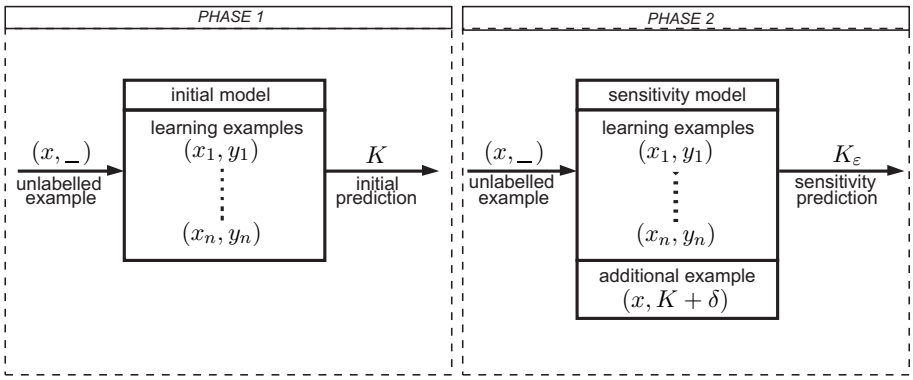

Fig. 2. The sensitivity analysis process. The figure illustrates the obtaining of initial prediction (phase 1) and the sensitivity model with sensitivity prediction K_ε (phase 2).

The obtained sensitivity predictions serve as output values in the sensitivity analysis process. In the previous work we showed that the magnitudes of changes in output $K_\varepsilon - K$ may be combined into reliability estimates and used as a measure of model instability for a modified example. Instead of focusing on the reliability estimates, proposed in the previous work, in this paper we focus on using the magnitudes of changes in output for the correction of the initial predictions.

Additionally, we showed that *more complex* learning algorithms that partition the input space (e.g. regression trees and neural networks) prior to modeling are more interesting for the sensitivity analysis approach than the other *more simple* algorithms (linear regression, locally weighted regression, etc.). Using such complex regression learners, the additional example can namely cause different space partitioning, leading to a considerably different hypothesis. This may also result in a big difference between initial and sensitivity predictions, which indicates that the initial hypothesis for tested example was unstable or unreliable. In our experimental work we therefore expect that the correction of regression predictions will be more successful for complex regression models than for simple regression models.

4 CORRECTION OF INITIAL PREDICTIONS

The basic idea of the proposed system for the correction of the initial predictions is to utilize an additional regression predictor (denoted by *secondary predictor*) which learns on the outputs of the main regression predictor (denoted by the *primary predictor*). The purpose of the secondary predictor is to predict the signed error of the predictions, computed by the primary predictor, i.e. to compute the necessary correction of the initial prediction.

For construction of the learning set for the secondary predictor we use the sensitivity predictions (2), based on which we form the following differences between the sensitivity predictions and the initial prediction:

$$K_{\varepsilon_1} - K, K_{-\varepsilon_1} - K, \ldots, K_{\varepsilon_m} - K, K_{-\varepsilon_m} - K. \tag{3}$$

We formed the above differences, since they represent the prediction sensitivity information which is independent of the particular prediction values. Since the differences are signed values, it is reasonable to assume that they may include the information about the direction and magnitude of the prediction error, making them suitable for predicting the necessary prediction correction. We therefore build the secondary predictor on the learning set, comprised by attributes (3) and the target value $C - K$ (signed prediction error of the primary predictor).

To assure the unbiasedness of the proposed system we used the cross-validation procedure to process the original testing data sets (divide them to learning and testing parts). In each iteration, the sensitivity predictions for all examples were computed and the attributes for the secondary predictor were formed using only the learning examples. After computing the secondary model, the predictions of the signed errors $\overline{C - K}$ were computed for the test examples. After combining the predicted signed error with the initial prediction K, we aim to achieve the corrected prediction

$$\overline{K} = K + (\overline{C - K}). \tag{4}$$

Having computed the initial prediction K and the corrected prediction $\overline{K}$ for all examples, we compare their accuracies by calculating the relative mean squared error (RMSE) for the whole data set. Both relative mean squared errors were statistically compared using paired t-test. This procedure for correction of the initial prediction and statistical evaluation is illustrated in Figure 3.

Since we assumed that the magnitudes of attributes 3 are proportional to the predicted error, we may also expect that the secondary predictor will achieve better results if one of the simple models is used (linear regression, locally weighted regression). We therefore expect the experimental results to confirm our hypotheses that the correction of regression predictions will work better for complex primary predictor models (H1), while the simple models will be the most appropriate models for the secondary predictor (H2).

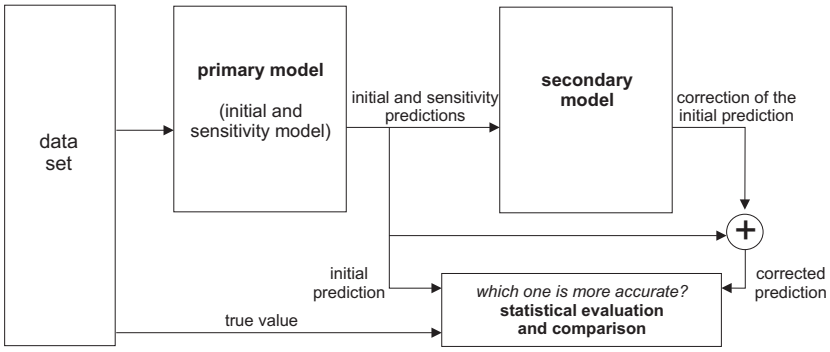

Fig. 3. The procedure for correction of the initial predictions using secondary regression model and statistical evaluation of the results

5 EXPERIMENTAL RESULTS

The proposed method for the correction of regression predictions was tested on 10 standard benchmark data sets, which are used across the whole machine learning community. Each data set is a regression problem. The application domains vary from medical, ecological and technical to mathematical and physical domains. The number of examples in these domains varies from 20 to over 6 500. Most of the data sets are available from UCI Machine Learning Repository [32] and from StatLib DataSets Archive [33]. All data sets are available from authors upon request. A brief description of data sets is given in Table 1.

Data set	# examples	# disc. attr.	# cont.attr.
auto_price	159	1	14
autompg	398	1	6
housing	506	1	12
cos4	1 000	0	10
cpu	209	0	6
grv	123	0	3
photo	858	2	3
pwlinear	200	0	10
servo	167	2	2
wsc	198	0	32

Table 1. Basic characteristics of testing data sets

We experimented with four regression models: regression trees (abbreviated as RT in the results), linear regression (LR), artificial neural networks (NN) and locally weighted regression (LW). Since the choices of regression models for the primary and the secondary predictor are independent, in our experimental work we test all possible 16 (4x4) combinations of them. We denote various combinations of models

by PP-SS, where PP and SS stand for the abbreviations of the primary and the secondary regression model, respectively (e.g. RT-NN). Some key properties of used models are:

- Regression trees: the mean squared error is used as the splitting criterion, the value in leaves represents the average label of examples, trees are pruned using the m-estimate [34].
- Linear regression: prediction with $(n-1)$-dimensional linear hyperplane in n-dimensional attribute space.
- Neural networks: multilayered feedforward artificial neural networks with back-propagation [35], one hidden layer of neurons, the learning rate was 0.5, the stoping criterion for learning is based on the change of MSE between two back-propagation iterations.
- Locally weighted regression: local regression using a distance function with Gaussian kernel for weighting the contributions of neighboring examples.

As described in Sections 3 and 4, we use a set of ε parameters to produce the attributes of examples for the secondary predictor. We used five different values of the ε parameter: $E = \{0.01, 0.1, 0.5, 1.0, 2.0\}$, thus defining 10 attributes for secondary predictor. Since some of these attributes can be more descriptive for predicting the signed error than the others, we approach this issue in two ways. First, with exhaustive search of all possible attribute combinations we test how the different attribute combinations influence the different results. Then, we test a method for automatic selection of best attributes for the secondary predictor and present its results.

5.1 Exhaustive Search for Optimal Set of Attributes

To test if the proposed system has the capability to correct the initial regression predictions at all and being faced with a difficulty which particular subset of attributes to optimally choose, we tested the system first using all possible combinations of attributes for the secondary predictor. In this way we focused on testing the systems functionality and postponed the problem of selecting the optimal attributes to a later time (see next subsection). Since the secondary predictor attributes are determined by the choice of ε parameters, we repeated the experiment for every number (of maximum five) and every combination of ε parameters in set $E = \{0.01, 0.1, 0.5, 1.0, 2.0\}$, which gives $2^5 - 1 = 31$ possible attribute sets for the secondary predictor (the power of the power set without the empty set). For each combination of selected ε parameters we therefore included the attributes $K - K_{\varepsilon}$ and $K - K_{-\varepsilon}$ in the secondary predictor learning set.

The results of described exhaustive search for the optimal attribute set are shown in Table 2 (for *locally weighted regression* and *linear regression* as the primary predictor) and in Table 3 (for *regression trees* and *neural networks*). The results in Table 2 confirm our findings from previous work that the simple regression models

(locally weighted regression and linear regression) are not suitable for use with the sensitivity analysis. Namely, among the experiments where the change of prediction accuracy was significant, the table shows the prevalent number of experiments where the RMSE increased.

In contrast to the above results, the results in Table 3 for more complex models (models that partition the input space, i.e. regression trees and neural networks) show the majority of achieved significant reductions of the RMSE and experiments with mixed changes (mixed change means that the use of some attribute sets resulted in reducing the RMSE, while the use of the other attribute sets increased the RMSE). High number of achieved results (50 % on the average with regression trees and 63 % on the average with neural networks) confirm our general expectation that it is possible to employ the sensitivity analysis for correction of the regression predictions.

The results also indicate that by selecting different attribute sets for the secondary predictor, it is possible to achieve different results, without having a consistent rule which values of parameters ε are the most suitable for defining the secondary attributes. This motivates us to exploring the possibilities for automatical selection of attributes, on which we focus in the following section.

5.2 Automatic Selection of Attributes for the Secondary Predictor

The results from the previous subsection showed that the proposed system is feasible to achieve the reduction in RMSE by correcting the regression predictions. However, we also concluded that there is no general rule in selecting the optimal sets of attributes for the secondary predictor. In this subsection we focus on extending the proposed approach with automatical selection of the best attribute set using RReliefF estimate [36, 37].

After producing a set of attributes using all values of parameter ε, we select only those attributes that have the value of RReliefF estimate greater than 0.0. Namely, Kononenko [36] has shown that estimates of ReliefF for classification problems (RReliefF is an adaptation of ReliefF for regression) are highly correlated with Gini index gain, which is a non-negative measure. In contrast to Gini index gain which assumes the independence of attributes, ReliefF estimates the gain in the context of other attributes. If the estimate is zero (or less than zero due to variations in data distributions), the attribute has zero gain and is therefore useless.

Having applied automatical selection of the optimal attributes, we gained two benefits. First, by limiting ourselves to the particular set of optimal attributes, we do not need to exhaustively test the performance of all attribute combinations, which reduced time complexity of testing. Second, the attribute selection using ReliefF allows greater flexibility, since it evaluates the usefulness of attributes $K-K_\varepsilon$ and $K - K_\varepsilon$ and includes them into the learning set of the secondary predictor independently of each other. In the exhaustive search, these two attributes appeared in the learning set at the same time, since the attributes for the secondary predictor

	LW	LW-LW	LW-LR	LW-RT	LW-NN	LR	LR-LW	LR-LR	LR-RT	LR-NN
autompg	0.240	0.241 [45]	0.240 [1]	0.265 [1]	0.295 [5]	0.192	0.192 [345]	0.199 [1]	0.192 [2]	0.200 [12]
autoprice	0.328	0.265 [5]	0.266 [1]	0.270 [5]	0.219 [35]	0.219	0.224 [1]	0.225 [14]	0.224 [45]	0.221 [12]
cos4	0.835	0.842 [1]	0.831 [1]	0.874 [23]	0.873 [345]	1.012	1.044 [1]	1.011 [1]	1.045 [1]	1.052 [1]
cpu	0.417	0.372 [15]	0.386 [1]	0.466 [3]	0.340 [12345]	0.261	0.255 [34]	0.279 [125]	0.276 [12]	0.347 [4]
grv	0.568	0.545 [1]	0.552 [1]	0.571 [13]	0.536 [123]	0.400	0.416 [1]	0.420 [1]	0.420 [1]	0.401 [34]
bhouse	0.492	0.467 [345]	0.481 [1]	0.482 [245]	0.525 [12345]	0.291	0.271 [5]	0.279 [134]	0.272 [35]	0.299 [124]
photo	0.331	0.330 [1]	0.318 [1]	0.334 [1234]	0.322 [125]	0.556	0.527 [1]	0.519 [2]	0.577 [124]	0.529 [25]
pwlinear	0.608	0.584 [1]	0.575 [1]	0.619 [135]	0.588 [345]	0.382	0.395 [1]	0.389 [1]	0.425 [1]	0.379 [1]
servo	0.633	0.547 [145]	0.555 [2]	0.553 [3]	0.585 [125]	0.638	0.536 [1]	0.554 [1]	0.709 [3]	0.560 [135]
wsc	0.956	0.969 [124]	0.979 [1]	1.108 [345]	0.959 [3]	0.988	1.001 [12]	0.991 [1]	1.195 [34]	1.004 [12345]
positive		30 %	30 %	0 %	10 %		20 %	10 %	0 %	10 %
negative		0 %	0 %	30 %	30 %		30 %	40 %	30 %	50 %
mixed		0 %	0 %	0 %	0 %		0 %	0 %	0 %	0 %

Table 2. The lowest relative mean square error (RMSE) achieved with *locally weighted regression* and *linear regression* as the primary regression predictors. The columns denoted by the single regression model designation (LW and LR) display the RMSE of the initial predictions of that model and the columns in PP-SS format display the lowest RMSE achieved using primary predictor model PP and secondary predictor model SS. The numbers $[n_1 \dots n_k]$ represent the consecutive numbers of used ε parameters in E giving the attribute set with which the displayed result was achieved. The cell shading indicates the results with significant changes in RMSE ($\alpha \leq 0.05$). Light grey shading denotes the positive results (reduction of RMSE), dark grey shading denotes the negative results (increase of RMSE) and the medium grey denotes the mixed changes (the use of some attribute sets resulted in reducing the RMSE, while the use of the other attribute sets increased the RMSE).

were selected based on the values of parameter ε (to maintain time complexity). We expect that this independent consideration of the two attributes may potentially lead to even greater reduction of the RMSE.

The results for the described approach are shown in Table 4. The results show that developed method managed to significantly reduce the RMSE in 36 tests, while significant increase occurred only in 5 tests. The lowest number of significant reductions of the RMSE in locally weighted regression and linear regression again confirm that the use of sensitivity analysis approach is not appropriate for these two methods. However, it can be seen from the comparison of Tables 2 and 4 that the automatic attribute selection using RReliefF managed to reduce the RMSE in some experiments even further (e.g. domain grv, LW-LW models). In addition, the proposed procedure in some cases managed avoid the attribute

	RT	RT-LW	RT-LR	RT-RT	RT-NN	NN	NN-LW	NN-LR	NN-RT	NN-NN
autompg	0.207	0.178 [13]	0.165 [23]	0.177 [13]	0.189 [35]	0.148	0.141 [4]	0.144 [13]	0.139 [12345]	0.145 [5]
autoprice	0.214	0.196 [245]	0.160 [12345]	0.194 [5]	0.183 [2345]	0.156	0.135 [1]	0.130 [4]	0.120 [4]	0.130 [45]
cos4	0.243	0.173 [23]	0.211 [25]	0.207 [24]	0.175 [235]	1.033	1.016 [5]	1.010 [5]	1.076 [45]	1.088 [4]
cpu	0.642	0.610 [14]	0.598 [14]	0.732 [2345]	0.797 [1]	0.254	0.245 [45]	0.400 [2345]	0.229 [1]	0.242 [1]
grv	0.447	0.429 [1245]	0.397 [1234]	0.439 [12]	0.427 [1345]	0.564	0.421 [25]	0.426 [1235]	0.434 [15]	0.405 [3]
bhouse	0.272	0.250 [1235]	0.216 [24]	0.243 [4]	0.293 [134]	0.336	0.260 [25]	0.279 [1]	0.266 [125]	0.288 [5]
photo	0.179	0.167 [345]	0.170 [3]	0.205 [2345]	0.187 [235]	0.490	0.387 [3]	0.441 [1]	0.395 [4]	0.396 [45]
pwlinear	0.254	0.235 [235]	0.204 [3]	0.287 [3]	0.219 [35]	0.169	0.162 [24]	0.151 [13]	0.202 [14]	0.166 [5]
servo	0.400	0.383 [145]	0.367 [5]	0.244 [24]	0.356 [1245]	0.215	0.200 [4]	0.172 [135]	0.282 [5]	0.207 [5]
wsc	1.211	1.053 [45]	1.003 [5]	1.570 [14]	1.000 [145]	1.660	0.988 [3]	1.014 [145]	1.117 [235]	0.978 [345]
positive		20 %	60 %	10 %	10 %		80 %	70 %	40 %	60 %
negative		0 %	0 %	40 %	30 %		0 %	10 %	30 %	10 %
mixed		50 %	0 %	10 %	40 %		0 %	10 %	10 %	0 %

Table 3. The lowest relative mean square error (RMSE) achieved with *regression trees* and *neural networks* as the primary regression predictors. The columns denoted by the single regression model designation (RT and NN) display the RMSE of the initial predictions of that model and the columns in PP-SS format display the lowest RMSE achieved using primary predictor model PP and secondary predictor model SS. The numbers $[n_1 \dots n_k]$ represent the consecutive numbers of used ε parameters in E giving the attribute set with which the displayed result was achieved. The cell shading indicates the results with significant changes in RMSE ($\alpha \leq 0.05$). Light grey shading denotes the positive results (reduction of RMSE), dark grey shading denotes the negative results (increase of RMSE) and the medium grey denotes the mixed changes (the use of some attribute sets resulted in reducing the RMSE, while the use of the other attribute sets increased the RMSE).

set which resulted in the increase of the RMSE as well (e.g. domain autompg, LW-RT).

Similar conclusions can also be drawn for experiments in which the regression trees and neural networks were used as the primary regression model. The results that stand out most with these two regression models are where the locally weighted regression was used as the secondary regression model. In model combinations RT-LW and NN-LW we namely managed to reduce the RMSE in 60 % and 50 % of domains, respectively, while the increase of the RMSE did not occur in any experiment.

We conclude that the results confirm both hypotheses: Best results were achieved by using complex primary predictor models (regression trees and neural networks) and by using a simple secondary predictor model (locally weighted regression).

	LW	LW-LW	LW-LR	LW-RT	LW-NN	LR	LR-LW	LR-LR	LR-RT	LR-NN
autompg	0.240	0.240	0.238	0.254	0.281	0.192	0.191	0.198	0.202	0.203
autoprice	0.328	0.246	0.261	0.342	0.237	0.219	0.223	0.250	0.226	0.221
cos4	0.835	0.840	0.830	0.891	0.865	1.012	1.012	1.012	1.012	1.012
cpu	0.417	0.369	0.357	0.487	0.320	0.261	0.251	0.264	0.262	0.311
grv	0.568	0.537	0.554	0.569	0.545	0.400	0.410	0.411	0.430	0.400
bhouse	0.492	0.466	0.480	0.487	0.507	0.291	0.273	0.309	0.274	0.298
photo	0.331	0.330	0.318	0.351	0.325	0.556	0.526	0.519	0.579	0.525
pwlinear	0.608	0.580	0.573	0.626	0.597	0.382	0.394	0.387	0.423	0.384
servo	0.633	0.538	0.546	0.655	0.580	0.638	0.602	0.546	0.692	0.568
wsc	0.956	0.956	0.956	0.956	0.956	0.988	0.988	0.988	0.988	0.988
positive		40 %	30 %	0 %	0 %		10 %	10 %	0 %	10 %
negative		0 %	0 %	10 %	10 %		0 %	0 %	10 %	10 %

	RT	RT-LW	RT-LR	RT-RT	RT-NN	NN	NN-LW	NN-LR	NN-RT	NN-NN
autompg	0.207	0.178	0.191	0.196	0.202	0.143	0.138	0.136	0.154	0.147
autoprice	0.214	0.198	0.157	0.244	0.169	0.091	0.088	0.090	0.101	0.097
cos4	0.243	0.180	0.169	0.183	0.183	1.096	1.063	0.982	1.109	1.104
cpu	0.642	0.601	0.621	0.694	0.674	0.220	0.221	0.246	0.229	0.235
grv	0.447	0.424	0.410	0.437	0.430	0.712	0.398	0.403	0.484	0.401
bhouse	0.272	0.248	0.215	0.262	0.266	0.200	0.181	0.182	0.207	0.203
photo	0.179	0.168	0.170	0.175	0.179	0.685	0.382	0.388	0.418	0.411
pwlinear	0.254	0.236	0.305	0.239	0.220	0.156	0.152	0.155	0.179	0.155
servo	0.400	0.377	0.748	0.410	0.339	0.103	0.108	0.120	0.138	0.108
wsc	1.211	1.049	1.001	1.195	1.005	1.673	1.176	1.182	1.226	1.286
positive		60 %	30 %	10 %	30 %		50 %	30 %	20 %	20 %
negative		0 %	0 %	0 %	0 %		0 %	10 %	0 %	0 %

Table 4. Relative mean square error (RMSE) before (columns LW, LR, RT and NN) and after the correction with the algorithm for automatical attribute selection for secondary predictor using RReliefF. The column names in PP-SS format denote the abbreviations of the primary predictor model PP and the secondary predictor model SS, respectively. Cell shading represents the p-values. The data with significance level $\alpha \leq 0.05$ is marked by light grey (significant reduction of the RMSE) and dark grey (significant increase of the RMSE) background. The columns emphasized by rectangles denote the subsets of the results referring to our hypotheses.

6 CONCLUSION

In the paper we presented a new method for correction of regression predictions. Our method is based on the sensitivity analysis, which is an approach that observes a change of the model outputs depending on the change in its inputs. In our approach we use an additional regression model, which has a task of modeling the signed prediction error for predictions, made by the main regression predictor. The secondary predictor predicts the prediction error based on the attributes, which we compute by the adapted sensitivity analysis procedure. By combining the initial predictions and the predicted signed error we gained the corrected predictions and statistically compared their accuracy to the accuracy of the initial predictions.

We performed the experiments using 4 regression models (regression trees, neural networks, linear regression and locally weighted regression) and by testing all

16 combinations of model pairs for primary and secondary predictor. The experiments with the exhaustive search for the secondary predictor optimal attribute set showed that the proposed approach is capable of correcting the regression predictions. They also confirmed our expectations based on our previous work that the proposed methodology is not suitable for simple regression models (linear regression and locally weighted regression) that do not partition the problem space prior to performing data modeling.

The testing of the proposed method for selection of the optimal attributes using RReliefF estimate showed that in most experiments the method successfully selects the set of attributes which lead to the reduction of the initial prediction error. The promising results showed the potential of using the proposed method with neural networks and regression trees, where locally weighted regression is employed as the secondary predictor model.

Besides extending and analyzing the performance evaluation of the proposed method also with other regression models, the ideas for further work include comparisons with alternative techniques for prediction correction. The performance of the proposed approach shall be compared to the predictor cascade, where the secondary predictor learns only from the initial predictions of primary predictions. The idea for further work lies also in eliminating the need for the secondary regression model by correcting regression predictions directly using the reliability estimates from our previous work [4].

Acknowledgements

We thank Matjaž Kukar and Marko Robnik-Šikonja for their contribution to this study.

REFERENCES

[1] Alpaydin, E.: Introduction to machine Learning. The MIT Press, Cambridge, Massachussetts 2004.

[2] Crowder, M. J.—Kimber, A. C.—Smith, R. L.—Sweeting, T. J.: Statistical Concepts in Reliability. Statistical Analysis of Reliability Data, Chapman & Hall, London, UK 1991, pp. 1–11.

[3] Bosnić, Z.—Kononenko, I.: An Overview of Advances in Reliability Estimation of Individual Predictions in Machine Learning. Intelligent Data Analysis, Vol. 13, 2008, No. 2, pp. 385–401.

[4] Bosnić, Z.—Kononenko, I.: Estimation of Individual Prediction Reliability Using the Local Sensitivity Analysis. Applied intelligence, Vol. 29, 2007, No. 3, pp. 187–203.

[5] Gammerman, A.—Vovk, V.—Vapnik, V.: Learning by Transduction. In: Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, Madison, Wisconsin1998, pp. 148–155.

[6] Saunders, C.—Gammerman, A.—Vovk, V.: Transduction with Confidence and Credibility. Proceedings of IJCAI, Vol. 2, 1999, pp. 722–726.

[7] Nouretdinov, L.—Melluish, L.—Vovk, V.: Ridge Regression Confidence Machine. In: Proc. 18th International Conf. on Machine Learning, Morgan Kaufmann, San Francisco 2001, pp. 385–39.

[8] Weigend, A.—Nix, D.: Predictions with Confidence Intervals (Local Error Bars). In: Proceedings of the International Conference on Neural Information Processing (ICONIP '94), Seoul, Korea 1994, pp. 847–852.

[9] Heskes, T.: Practical Confidence and Prediction Intervals. In: Michael C. Mozer, Michael I. Jordan, Thomas Petsche (Eds.): Advances in Neural Information Processing Systems, 1997, Vol. 9, The MIT Press, pp. 176–182.

[10] Carney, J.—Cunningham, P.: Confidence and Prediction Intervals for Neural Network Ensembles. In: Proceedings of IJCNN '99, The International Joint Conference on Neural Networks, Washington, USA 1999, pp. 1215—1218.

[11] Birattari, M.—Bontempi, H.—Bersini, H.: Local Learning for Data Analysis. In: Proceedings of the 8th Belgian-Dutch Conference on Machine Learning 1998, pp. 55–61.

[12] Schaal, S.—Atkeson, C. G.: Constructive Incremental Learning from Only Local Information. In: Neural Computation, Vol. 10, 1998, No. 8, pp. 2047–2084.

[13] Woods, K.—Kegelmeyer, W. P.—Bowyer, K.: Combination of Multiple Classifiers Using Local Accuracy Estimates. In: IEEE Transactions on PAMI, Vol. 19, 1997, No. 4, pp. 405–410.

[14] Giacinto, G.—Roli, F.: Dynamic Classifier Selection Based on Multiple Classifier Behaviour. In: Pattern Recognition, Vol. 34, 2001, No. 9, pp. 1879–881.

[15] Vapnik, V.: The Nature of Statistical Learning Theory. Springer Verlag 1995.

[16] Kukar, M.—Kononenko, I.: Reliable Classifications With Machine Learning. In: Proc. Machine Learning: ECML-2002, Helsinki (Finland) 2002, Springer Verlag, pp. 219–31.

[17] Bosnić, Z.—Kononenko, I.—Robnik-Šikonja, M.—Kukar, M.: Evaluation of Prediction Reliability in Regression Using the Transduction Principle. In: Proc. of Eurocon 2003, Ljubljana, pp. 99–103.

[18] Bousquet, O.—Elisseeff, E.: Stability and Generalization. Journal of Machine Learning Research, Vol. 2, pp. 499–526, 2002.

[19] Bousquet, O.—Elisseeff, A.: Algorithmic Stability and Generalization Performance. In: NIPS, 2000, pp. 196–202.

[20] Bousquet, O.—Pontil, M.: Leave-One-Out Error and Stability of Learning Algorithms With Applications. In: Suykens, J. A. K. et al. (Eds.): Advances in Learning Theory: Methods, Models and Applications, IOS Press 2003.

[21] Kearns, M. J.—Ron, D.: Algorithmic Stability and Sanity-Check Bounds for Leave-One-Out Cross-Validation. In: Computational Learning Theory, 1977, pp. 152–162.

[22] Breiman, L.: Bagging Predictors. In: Machine Learning, Vol. 24, 1996, pp. 123–140.

[23] Schapire, R. E.: A Brief Introduction to Boosting. Proc. IJCAI, pp. 1401–1406, 1999.

[24] Drucker, H.: Improving Regressors Using Boosting Techniques. In: Machine Learning: Proceedings of the Fourteenth International Conference 1977, pp. 107–115.

[25] Ridgeway, G.—Madigan, D.—Richardson, T.: Boosting Methodology for Regression Problems. In: Proc. Artificial Intelligence and Statistics 1999, pp. 152–161.

[26] Tibshirani, J. R.—Knight, K.: The Covariance Inflation Criterion for Adaptive Model Selection. In: Journal of the Royal Statistical Society, Series B 61, 1999, pp. 529–546.

[27] Rosipal, R.—Girolami, M.—Trejo, L.: On Kernel Principal Component Regression with Covariance Inflation Criterion for Model Selection. Technical report, University of Paisley 2000.

[28] Elidan, G.—Ninio, M.—Friedman, N.—Shuurmans, D.: Data Perturbation for Escaping Local Maxima in Learning. In: Proc. AAAI/IAAI, 2002, pp. 132–139.

[29] Mitchell, T.: The Role of Unlabelled Data in Supervised Learning. In: Proceedings of the 6$^{\text{th}}$ International Colloquium of Cognitive Science, San Sebastian, Spain, 1999.

[30] Blum, A.—Mitchell, T.: Combining Labeled and Unlabeled Data with Co-training. In: Proceedings of the 11$^{\text{th}}$ Annual Conference on Computational Learning Theory 1998, pp. 92–100.

[31] Li, M.—Vitányi, P.: An Introduction to Kolmogorov Complexity and its Applications. Springer-Verlag, New York 1993.

[32] Newman, D. J.—Hettich, S.—Blake, C. L.—Merz, C. J.: UCI Repository of Machine Learning Databases. University of California, Irvine, Dept. of Information and Computer Sciences 1998.

[33] Department of Statistics at Carnegie Mellon University: StatLib – Data, Software and News from the Statistics Community. 2005.

[34] Cestnik, B.—Bratko, I.: On Estimating Probabilities in Tree Pruning. In: Proceedings of European working session on learning (EWSL-91), Porto, Portugal 1991, pp. 138–150.

[35] McCulloch, W. S.—Pitts, W.: A Logical Calculus of the Ideas Imminent in Nervous Activity. In: Bull. of Math. and Biophysics, Vol. 5, 1943, pp. 115–133.

[36] Kononenko, I.: Estimating Attributes: Analysis and Extensions of RELIEF. In: Proc. Machine learning: ECML-94, European conference on machine learning, ISBN 3-540-57868-4, Catania, Sicily 1994, Springer-Verlag, pp. 171–182.

[37] Robnik-Šikonja, M.—Kononenko, I.: An Adaptation of Relief for Attribute Estimation in Regression. In Proc. Int. Conf. on Machine Learning ICML-97, ISBN 1-55860-486-3, Nashville, pp. 296–304.

Zoran Bosnić obtained his Master and Doctor degrees in computer science at University of Ljubljana (Slovenia) in 2003 and 2007, respectively. Since 2006 he has been employed at Faculty of Computer and Information Science and currently he works as Assistant Professor in the Laboratory of Cognitive Modelling. His research interests include artificial intelligence, machine learning, regression, and reliability estimation for individual predictions, as well as applications in these areas.

Igor Kononenko received his Ph. D. in 1990 from University of Ljubljana, Slovenia. He is a Professor at the Faculty of Computer and Information Science in Ljubljana and the Head of the Laboratory for Cognitive Modeling. His research interests include artificial intelligence, machine learning, neural networks and cognitive modeling. He is a member of the editorial board of Applied Intelligence Journal and Informatica Journal. He is a (co)author of 180 papers and 10 textbooks. Recently he co-authored the book "Machine Learning and Data Mining: Introduction to Principles and Algorithms" (Hoorwood, 2007).