UDC 004.4'24

PREDICTING THE PROBABILITY OF EXCEEDING CRITICAL SYSTEM THRESHOLDS

Peter Krammer, Marcel Kvassay, Ladislav Hluchý

In this paper we show how regression modelling can be combined with a special kind of data transformation technique that improves model precision and produces several "preliminary" estimates of the target value. These preliminary estimates can be used for interval estimates of the target value as well as for predicting the probability that it has or will exceed arbitrary predefined thresholds. Our approach can be combined with various regression models and applied in many domains that need to estimate the probability of system malfunctions or other hazardous states brought about by system variables exceeding critical safety thresholds. We rigorously derive the formulas for the probability of crossing an upper bound and a lower bound both separately (one-sided intervals) and together (a two-sided interval), and verify the approach experimentally on a real dataset from the electric power industry.
Key words: regression, data transformation, interval estimation, probability, statistical modelling.

У цій статті показано, як регресійне моделювання можна комбінувати зі спеціальним видом перетворення даних, яке покращує точність моделі і дає кілька «попередніх» оцінок цільового значення. Ці попередні оцінки можна використовувати для інтервальних оцінок цільового значення, а також для прогнозування ймовірності того, що воно прийме або перевищить довільні попередньо визначені порогові значення. Наш підхід можна комбінувати з різними регресійними моделями і застосовувати в багатьох областях, які повинні оцінювати вірогідність збоїв системи або інших небезпечних станів, викликаних системними змінними, що перевищують критичні пороги безпеки. Ми строго виводимо формули для ймовірності перетину верхньої та нижньої межі як окремо (односторонні інтервали), так і разом (двосторонній інтервал) і перевіряємо наш підхід експериментально на реальному наборі даних з електроенергетики.
Ключові слова: регресія, перетворення даних, оцінка інтервалів, ймовірність, статистичне моделювання.

В этой статье показано, как регрессионное моделирование можно комбинировать со специальным видом преобразования данных, которое улучшает точность модели и дает несколько «предварительных» оценок целевого значения. Эти предварительные оценки могут использоваться для интервальных оценок целевого значения, а также для прогнозирования вероятности того, что оно примет или превысит произвольные предопределенные пороговые значения. Наш подход можно комбинировать с различными регрессионными моделями и применять во многих областях, которые должны оценивать вероятность сбоев системы или других опасных состояний, вызванных системными переменными, превышающими критические пороги безопасности. Мы строго выводим формулы для вероятности пересечения верхней и нижней границы как отдельно (односторонние интервалы), так и вместе (двухсторонний интервал), и проверяем наш подход экспериментально на реальном наборе данных из электроэнергетики.
Ключевые слова: регрессия, преобразование данных, оценка интервалов, вероятность, статистическое моделирование.

Introduction

Our lives are directly or indirectly influenced by information technologies in a variety of ways, and data collection and analysis, as well as modelling and prediction of important variables, are now routinely performed in domains as diverse as electric power industry, hydrology, public health or banking. In all these areas there exists a need to increase the precision and reliability of existing models. More precise and robust models can improve not only the productivity of existing systems but also their security and safety, thus helping to save human lives in cases of emergency. Modelling tasks in these domains often include the prediction of error, risk and various hazardous states. This paper focuses on the estimation of probability that a given system variable has or will exceed predefined safety thresholds and, as a result, the operation of the system could be severely compromised. We also touch upon the problem of improving model precision since our regression model is used in conjunction with a special data transformation technique formulated in [1–3] for that purpose.

This data transformation technique was inspired by ensemble learning methods of machine learning. In machine learning, model accuracy and robustness are typically enhanced through various forms of ensemble learning [1–5], such as Boosting, Bagging, Dagging, Stacking, Additive regression, etc. These methods exploit techniques like aggregation of different types of models, multiple training phases, submodels voting and weighting of data records. First ensemble methods were intended primarily for classification; those for regression appeared later [6]. Some of the more recent ones include evolutionary ensembles [7], multiple network fusion [8] and hybrid ensembles [9]. Several studies, e.g. [4, 10] analyze the suitability of ensemble methods with respect to the type of data or properties of submodels.

Let us consider a homogeneous data table whose rows represent measurement records and whose columns represent their attributes. Each attribute describes the quantity or quality of some physical variable (pressure, throughput, voltage, etc.) and each row comprises attributes measured at the same time or place. Let us further assume that all the attributes (the input ones as well as the target one) are numerical and continuous, i.e. real-valued. Moreover, the input attributes have already been normalized and selected for their relevance with respect to the target one. The regression task then consists in modelling the target attribute as a suitable function of the input ones. It is typically approached by training various types of regression models on the available data.

In our previous work [3] we proposed a data transformation technique enhancing the precision of machine learning models and predictors for real-valued target variables. Its application to several realistic data sets

considerably improved prediction accuracy. Moreover, it could be easily combined with various types of regression models. These benefits, however, had to be "paid for" in terms of longer calculations when compared to regression on the original untransformed data. One positive side-effect of this technique is the generation of several "preliminary" target estimates in an interim step. The final target estimate is then calculated as their simple arithmetic mean although, in principle, we could use a weighted mean too. While the standard version of our technique uses only the final estimate and discards the preliminary ones, these preliminary estimates contain further valuable information besides the mean, which can be extracted through more advanced statistical principles and formulas, as we explain further below.

1. A Brief Outline of the Data Transformation Technique

The basic data transformation was already published in [3], [11] which demonstrated both its advantages and disadvantages on several synthetic and real datasets. We therefore sketch only its main idea here and do not elaborate on its properties, limitations or parameter settings. The essence of the transformation consists in the creation of all possible pairs of records from the original dataset except the identical pairs (i.e. we do not pair any record with itself). In this way we transform the original dataset (shown schematically in Tab. 1) into a new one shown in Tab. 2.

Table 1. Structure of the original data set

Record ID	***Input Attribute Z***	***Input Attribute Y***	***Target Attribute O***
{1}	z_1	y_1	o_1
{2}	z_2	y_2	o_2
{3}	z_3	y_3	o_3
{4}	z_4	y_4	o_4

Table 2. Structure of the transformed data set

ID of Used Records	***Input Attribute Z***	***Input Attribute Y***	ΔZ	ΔY	ΔO
{1}, {2}	z_1	y_1	$z_1 - z_2$	$y_1 - y_2$	$o_1 - o_2$
{1}, {3}	z_1	y_1	$x_1 - z_3$	$y_1 - y_3$	$o_1 - o_3$
{1}, {4}	z_1	y_1	$z_1 - z_4$	$y_1 - y_4$	$o_1 - o_4$
{2}, {1}	z_2	y_2	$z_2 - z_1$	$y_2 - y_1$	$o_2 - o_1$
...	...	...	...	...	...
{4}, {3}	z_4	y_4	$z_4 - z_3$	$y_4 - y_3$	$o_4 - o_3$

Each of N records in the original dataset is paired with the remaining N-1 records, and all these pairs are added to the transformed dataset. The size of the transformed dataset is then N^2–N (note that the pairing is not symmetrical, because [{i}, {k}] does not equal [{k}, {i}]). The number of input attributes in the transformed dataset doubles, because for each original one there is now added the difference in its value between the two paired records. Moreover, the difference between the two target values becomes the new target attribute for prediction (the last column ΔO in Tab. 2). This reification of attribute differences into new standalone attributes emphasizes their similarity or dissimilarity and, in effect, highlights the resemblance (or lack of it) between the original data records. Model training is thus more sensitive to attribute differences compared to classical training without transformation, in which attribute dynamics are not emphasized and attribute values from different records have no means to "meet" and influence each other. In the next step, a chosen regression model is trained on the transformed data. Since the regression model trained on the transformed data predicts the difference in the target value, we need to apply a correction (inverse transformation) in order to arrive at the prediction of the original target value. Essentially, each transformed pair involving a given original data record can be used to convert (or "correct") the prediction of the target difference into that of its original target value. Because our transformed training set contains several such pairs

for each original data record, we can produce several "preliminary" estimates of its target value. We describe this process in more detail in [3, 11].

2. Interval Estimation

As mentioned above, one advantage of our data transformation is the production of several "preliminary" estimates of the target value. Statistical principles enable us to extract from them further important information, e.g. an interval estimate or the probability that a predefined threshold has been or will be exceeded.

An interval estimate consists of the lower and the upper bounds within which the target value should stay with a given probability. It represents the uncertainty of our calculations, because even a well-trained model's predictions will have some margin of error. In some cases the predictions will be close to the real value, in others they may be quite far. A narrower interval estimate for a given probability signals lower uncertainty and vice versa.

In our context the interval estimates should *not* be calculated using the formula for *confidence intervals* [12] because these are meant for *population parameters*, such as the mean or the variance, whereas we now need to bound a certain proportion of the population itself, i.e. our individual data points or measurement records. For this purpose, *tolerance region* [13] defined by formula (1) is an appropriate method:

$$X_{RANGE} \quad \bar{X} \pm s_X . \mathrm{tinv}\left(1 \quad \frac{\alpha}{2}, M \quad 1\right) . \sqrt{1 + 1/M} \tag{1}$$

In our context, the meaning of variables and functions in formula (1) is as follows. (Please note that we also explain here some additional variables used in subsequent derivations further below.) Function symbols *tcdf()* and *tinv()* follow the convention used in Matlab [14].

X – a variable whose values are the "preliminary" estimates of the target value

X_{RANGE} – the bounds of the tolerance region

M – the number of available "preliminary" estimates of the target value

$\bar{X}$ – an average of the M "preliminary" estimates

s_X – standard deviation of the M "preliminary" estimates

α – a significance level, which determines the probability that the tolerance interval will encompass the target value. For a 95 % interval, $\alpha \quad 1 \quad 0.95 \quad 0.05$

$\beta = \Pr(X < X_{THR})$ – probability that X takes on a smaller value than the threshold X_{THR}

tcdf (Z, M-1) – Student's **t** cumulative distribution function[1] for value Z, with M-1 degrees of freedom.

tinv (p, M-1) – inverse of Student's **t** cumulative distribution function[2], with M-1 degrees of freedom, and probability p.

Tolerance regions calculated in this way provide information about the bounds for the value of X at a given significance level α. This can be tested by repeated model training and interval calculation: the ratio of "successful" cases (in which the tolerance region does encompass the actual value of X) should converge towards $1 \quad \alpha$.

Nevertheless, practical application of this information is rather limited and many domain-oriented applications tend towards the inverse task – the calculation of the probability that a certain predefined threshold level of X has been or will be exceeded. We need to keep in mind that even when the prediction model's final estimate of X (calculated as a mean of several "preliminary" estimates) does not cross the threshold, this estimate is not error-free and so, in fact, X may have crossed the threshold anyway. The calculation of the probability that X has exceeded the threshold therefore needs to consider not only the distance between the threshold and the final estimate, but also the variance of the "preliminary" estimates from which it was derived.

3. Derivation of Probability of Exceeding a Threshold

Formula (1) determines both ends of tolerance region at once. For our purposes a one-sided simplification will be more helpful: formula (2) gives only the upper bound X_{THR} for a pre-specified probability β defined as $\beta = \Pr(X < X_{THR})$:

$$X_{THR} \quad \bar{X} + s_X . \mathrm{tinv}(\beta, M \quad 1) . \sqrt{1 + 1/M} \tag{2}$$

In the next step we isolate the function *tinv()* on the right-hand side by shifting all the other terms to the left-hand side:

[1] https://www.mathworks.com/help/stats/tcdf.html
[2] https://www.mathworks.com/help/stats/tinv.html

$$\frac{x_{THR} - \bar{x}}{s_x \sqrt{1 + 1/M}} \quad tinv(\beta, M \quad 1). \tag{3}$$

We can then apply the function *tcdf()*, with $M \quad 1$ degrees of freedom, to both sides:

$$tcdf\left(\frac{x_{THR} - \bar{x}}{s_x \sqrt{1 + 1/M}}, M \quad 1\right) \quad tcdf(tinv(\beta, M \quad 1), M \quad 1). \tag{4}$$

Because the function *tinv()* is the inverse of *tcdf()* and both share the same number of degrees of freedom (M-1), they cancel each other and leave just the desired probability $\beta = \Pr(X < X_{THR})$ on the right-hand side:

$$tcdf\left(\frac{x_{THR} - \bar{x}}{s_x \sqrt{1 + 1/M}}, M \quad 1\right) \quad \beta. \tag{5}$$

This result can be more conveniently expressed as formula (6):

$$\Pr(X < X_{THR}) \quad tcdf\left(\frac{x_{THR} - \bar{x}}{s_x \sqrt{1 + 1/M}}, M \quad 1\right). \tag{6}$$

If we are interested in the lower bound, we can easily derive the formula from the probability of the complementary event:

$$\Pr(X \geq X_{THR}) \quad 1 \quad tcdf\left(\frac{x_{THR} - \bar{x}}{s_x \sqrt{1 + 1/M}}, M \quad 1\right). \tag{7}$$

Finally, by combining the two, we can derive the probability that the actual value of X is confined between a lower bound X_{THR1} and an upper bound X_{THR2}. This is expressed by formula (8):

$$\Pr(X_{THR1} < X < X_{THR2}) \quad tcdf\left(\frac{x_{THR2} - \bar{x}}{s_x \sqrt{1 + 1/M}}, M \quad 1\right) \quad tcdf\left(\frac{x_{THR1} - \bar{x}}{s_x \sqrt{1 + 1/M}}, M \quad 1\right) \tag{8}$$

In general, formulas (6), (7) and (8) enable us to calculate the probability that the actual value of some predicted variable X is confined within some predefined region. Conversely, through formulas (1) and (2) we can calculate the bounds for a given predefined probability or significance level.

In practical applications the threshold X_{THR} would typically represent some structural limit, the crossing of which might result in system damage, compromised security or danger to human lives. The formulas derived above help us to quantify the probability of such undesirable developments.

The proposed approach is primarily suited for

A.) Modelling and prediction of future values of X from its past and present values, e.g. in various early warning systems;

B.) Modelling and prediction of some variable X whose direct measurement would be dangerous, technologically demanding, or costly in terms of energy, time, money, etc. In these situations it is preferable to measure other related variables, formulate and train on them a good regression model, and use that to predict X.

We therefore do not focus on one specific scenario but try to keep a more general perspective. Our goal is to formulate an approach applicable in various situations that require modelling and prediction of one or more system variables in combination with a warning system that guards a set of predefined system constraints.

4. Experiments

We tested our approach experimentally on a publicly available dataset called Energy Efficiency [6], which contained 768 records and 9 numeric attributes including the target one. We deliberately chose this smaller dataset because our data transformation (with 15-fold repetition of experiments in order to make them more representative), took quite long to compute. For regression modelling we used a feedforward neural network of perceptrons with one hidden layer and sigmoid activation function. Learning rate was set to 0.3 and max. training epochs to 500. Table 3 shows some prediction examples for this dataset as well as the corresponding threshold-crossing probabilities and 90 % tolerance intervals.

Table 3. Sample predictions of X ($\bar{X}$) along with their corresponding threshold-crossing probabilities Pr(X < 8.0) and Pr(X > 40.0) and 90 % tolerance intervals. The real value of X is given in the last column

Record Number	$\bar{X}$	s_X	Pr(X < 8.0)	Pr(X > 40.0)	90% interval	X_{REAL}
1	7.5901	1.1125	0.638447	0.0	5.6189 9.5612	6.04
2	8.9423	4.5804	0.421507	1.241341E-6	0.8266 17.0580	8.50
3	10.4178	1.2074	0.032779	5.551115E-16	8.2784 12.5570	10.64
4	22.3606	1.3488	1.415263E-9	4.545264E-11	19.9708 24.7505	23.75
5	22.3789	4.6541	0.003560	7.687263E-4	14.1326 30.6251	24.77
6	36.9891	1.5087	5.122037E-14	0.033197	34.3159 39.6623	36.45
7	39.7402	0.9383	1.519116E-18	0.394970	38.0776 41.4028	39.04
8	39.7640	1.0177	6.820442E-18	0.411707	37.9608 41.5673	39.83
9	43.1567	0.9128	1.334947E-19	0.998410	41.5394 44.7739	41.73

We performed this particular experiment only once, because our goal was just to illustrate the process for a few concrete target values. The table lists the final estimate $\bar{X}$ calculated as an average of 20 "preliminary" estimates produced by the neural network as well as the standard deviation s_X of these preliminary estimates. It next shows the threshold-crossing probabilities Pr(X < 8.0) and Pr(X > 40.0) calculated from (6) and (7), and the corresponding 90 % tolerance interval. The real value of X in the last column (X_{REAL}) is shown just for verification – it was not available to the regression models.

By comparing the individual rows in Table 3 we can see the effect of the changing average $\bar{X}$ on the threshold-crossing probabilities, which increase or decrease sharply as $\bar{X}$ enters or leaves each guarded region. For example, as the average $\bar{X}$ changes from 36.9891 to 39.7402 between rows six and seven, the probability Pr(X > 40.0) increases more than tenfold from 0.033197 to 0.394970. Similarly, as $\bar{X}$ changes from 8.9423 to 10.4178 between rows two and three, Pr(X < 8.0) sharply decreases from 0,421507 to 0,032779.

Another important fact can be seen in rows four and five – a surprisingly large influence of the standard deviation s_X on the threshold-crossing probabilities. Both rows share a very similar value of $\bar{X}$, yet their threshold-crossing probabilities for both thresholds differ by several orders of magnitude. This is caused by the difference in the deviation s_X, since the higher the s_X, the higher the probability of exceeding the threshold.

In order to cross-check the soundness of our calculations, we have substituted the bounds of our 90 % tolerance intervals from the sixth column of the table into formula (8), expecting to get the same result (0.9) for each row. The values that we obtained were indeed very close to 0.9 – if we denote the error as ε, our results were 0.9 + ε, with the maximum absolute value of the error $|\varepsilon_{MAX}| < 2.6E\text{-}13$, which we interpret as the confirmation of sufficient precision in our calculations.

As can be seen from our derivations above, the task of predicting the probability of exceeding a pre-defined threshold is inverse to that of estimating the bounds of an interval within which the target value should reside with a given probability. Both exploit the same mathematical relationship between the bounds and the probability, but they approach it from the opposite sides. This entitled us to verify the correctness of our technique by testing the validity of our 90 % tolerance intervals, which is much easier to do than to verify the inverse task. Accordingly, we have

trained regression models on various subsets of our transformed data and repeatedly calculated the bounds for 90 % tolerance intervals, each time also noting whether or not they did contain the actual value of X (which was known to us but hidden from the regression models). We display these results in Table 4.

Table 4 lists averaged values for model precision, elapsed calculation time and the success rate of interval estimates for each size of the training set (specified in the column *NumRec*). Each row in the table shows the values averaged over 15 independent experiment runs with different random seeds. The seeds governed the inclusion of the records in the training set as well as random initialization of neural networks generating the models.

The average precision of our trained regression models is expressed through Correlation Coefficients (column *CorrCoef*) and Root Mean Squared Error (column *RMSE*). Column *Time* shows average time in seconds needed for one modelling cycle, i.e. for training the model on *NumRec* randomly selected records in the training set and then predicting the target value for the remaining records in the dataset. It should be noted that regression on the transformed data takes considerably longer than traditional regression on the original untransformed dataset.

Column *Intervals* shows the total number of interval estimates performed and column *Correct* the number of those that did encompass the real value X_{REAL}. Column *Ratio* then shows their success rate, which is defined as the ratio of Correct / Intervals. Since these estimates were meant to represent 90% tolerance intervals, the values in column *Ratio* should be no less than 0.90. We can see that it is indeed so for all the rows except the last two, where the small size of the training set (less than 100 records) negatively impacted the success rate as well as precision (lowering the Correlation Coefficients and increasing the Root Mean Squared Errors).

Nevertheless, we consider the experimental evaluation a success, because for sufficiently representative training sets (with more than 100 records) the success rate of interval estimates reached or crossed 90 %, as expected. In fact, for training sets with 120 records or more the success rate consistently exceeded 94 %. We thus feel entitled to conclude that the proposed approach can be practically deployed in various experimental scenarios. In the future we plan to test our approach on other datasets and conduct an in-depth analysis of experimental results. We also intend to investigate the lower bound for the training set size below which the interval estimates fail to reach the expected success rate.

Table. 4. Dependence of model precision (*CorrelCoef*, *RMSE*), calculation time (*Time*) and the success rate of 90 % tolerance interval estimates (*Ratio*) on the number of records in the training set (*NumRec*)

NumRec	*CorrelCoef*	*RMSE*	*Time*	*Correct*	*Intervals*	*Ratio*
260	0.999317	0.7364	67.3	7440	7620	0.9764
250	0.998838	0.7929	65.7	7572	7770	0.9745
240	0.999166	0.7501	60.8	7707	7920	0.9731
230	0.998847	0.7848	56.3	7886	8070	0.9772
220	0.998761	0.7919	51.7	7969	8220	0.9695
210	0.998387	0.8271	47.1	8065	8370	0.9636
200	0.998571	0.8058	42.8	8246	8520	0.9678
190	0.998135	0.8442	39.5	8392	8670	0.9679
180	0.997246	0.9321	35.0	8496	8820	0.9633
170	0.998015	0.8550	31.5	8615	8970	0.9604
160	0.997539	0.9104	28.4	8736	9120	0.9579
150	0.997075	0.9581	25.0	8759	9270	0.9449
140	0.997196	0.9439	22.0	8973	9420	0.9525
130	0.996817	0.9785	18.9	9079	9570	0.9487
120	0.994856	1.1462	16.2	9237	9720	0.9503
110	0.989257	1.3735	13.7	9143	9870	0.9263
100	0.989170	1.3626	11.2	9263	10020	0.9245

NumRec	*CorrelCoef*	*RMSE*	*Time*	*Correct*	*Intervals*	*Ratio*
90	0.974928	1.9311	9.6	9122	10170	0.8970
80	0.976517	1.9920	7.6	9103	10320	0.8821

Conclusions

In this paper we have presented a new approach exploiting a special data transformation for regression tasks. The transformation improves model precision through the production and utilization of a number of "preliminary" estimates of the target value. Moreover, it can be combined with different types of regression models. We have shown that this technique is suitable for tolerance interval estimates as well as for predicting the probability of exceeding arbitrary predefined thresholds. We have rigorously derived the formulas for the probability of crossing an upper bound and a lower bound both separately (one-sided intervals) and together (a two-sided interval). Our experimental evaluation confirmed a satisfactory performance of the proposed technique in terms of model precision and success rate for sufficiently large training sets. The proposed approach can be applied in various domains for predicting the probability of hazardous situations brought about by important system variables exceeding predefined safety thresholds.

In the future we plan to investigate in more detail and for various datasets how the minimum required size of the training set depends on the total size of the dataset and on the required model precision.

This work was supported by projects: VEGA 2/0167/16 (2016 - 2019) and PROCESS EU H2020-777533 (2017-2020).

References

1. Hastie T., Tibshirani R., Friedman J. The elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition 2009, Springer. P. 463–470, 605–622. http://web.stanford.edu/~hastie/Papers/ESLII.pdf
2. Ian H. Witten, Eibe Frank: Data Mining: Practical Machine Learning Tools and Techniques, Second Edition, Elsevier. P. 315–334, 414–418.
3. Peter Krammer, Marcel Kvassay, Ladislav Hluchý: Improved regression method with interval estimation. In ICNC-FSKD 2017: 2017 13th international conference on natural computation, fuzzy systems and knowledge discovery. - Guilin, China: IEEE, 2017. P. 2402–2408.
4. Jain Anil K., Robert P. W. Duin, Jianchang Mao: Statistical Pattern Recognition: A Review. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000. P. 4–37.
5. Dietterich T.G. An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization, Machine Learning 2000, 40(2). P. 139–157.
6. UCI Machine Learning Repository: Energy Efficiency, Center for Machine Learning and Intelligent Systems, https://archive.ics.uci.edu/ml/datasets/Energy+efficiency
7. K. Krishnamoorthy: Statistical Tolerance Regions: Theory, Applications, and Computation. 2009. John Wiley and Sons. P. 1–6.
8. Liu Y., Yao X., Higuchi T. Evolutionary Ensembles with Negative Correlation Learning. IEEE Transactions on Evolutionary Computation. 2000. P. 380–387.
9. CHO Sung-Bae, and Jin H. KIM: Multiple Network Fusion Using Fuzzy Logic. IEEE Transactions on Neural Networks. 1995, 6(2). P. 497–501.
10. Chandra Arjun, and Xin Yao: Evolving Hybrid Ensembles of Learning Machines for Better Generalisation, Neurocomputing. 2006. 69(7–9). P. 686–700.
11. Krammer Peter, Habala Ondrej, Hluchý Ladislav. Transformation regression technique for data mining. In IEEE International Conference on Intelligent Engineering Systems. 2016, vol., art. no. 7555134. P. 273–277.
12. Prasanna Sahoo: Probability and Mathematical Statistics, University of Louisville. 2013. P. 497–584. http://www.math.louisville.edu/~pksaho01/teaching/Math662TB-09S.pdf
13. Krishnamoorthy K. Statistical Tolerance Regions: Theory, Applications, and Computation, 2009, John Wiley and Sons. P. 1–6.
14. Matlab, Statistics and Machine Learning Toolbox Functions, https://www.mathworks.com/help/stats/functionlist-alpha.html

Література

1. Hastie T., Tibshirani R., Friedman J. The elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition 2009, Springer. P. 463–470, 605–622. http://web.stanford.edu/~hastie/Papers/ESLII.pdf
2. Ian H. Witten, Eibe Frank: Data Mining: Practical Machine Learning Tools and Techniques, Second Edition, Elsevier. P. 315–334, 414–418.
3. Peter Krammer, Marcel Kvassay, Ladislav Hluchý: Improved regression method with interval estimation. In ICNC-FSKD 2017: 2017 13th international conference on natural computation, fuzzy systems and knowledge discovery. - Guilin, China: IEEE, 2017. P. 2402–2408.
4. Jain Anil K., Robert P. W. Duin, Jianchang Mao: Statistical Pattern Recognition: A Review. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000. P. 4–37.
5. Dietterich T.G. An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization, Machine Learning 2000, 40(2). P. 139–157.
6. UCI Machine Learning Repository: Energy Efficiency, Center for Machine Learning and Intelligent Systems, https://archive.ics.uci.edu/ml/datasets/Energy+efficiency
7. K. Krishnamoorthy: Statistical Tolerance Regions: Theory, Applications, and Computation. 2009. John Wiley and Sons. P. 1–6.
8. Liu Y., Yao X., Higuchi T. Evolutionary Ensembles with Negative Correlation Learning. IEEE Transactions on Evolutionary Computation. 2000. P. 380–387.

9. CHO Sung-Bae, and Jin H. KIM: Multiple Network Fusion Using Fuzzy Logic. IEEE Transactions on Neural Networks. 1995, 6(2). P. 497–501.
10. Chandra Arjun, and Xin Yao: Evolving Hybrid Ensembles of Learning Machines for Better Generalisation, Neurocomputing. 2006. 69(7–9). P. 686–700.
11. Krammer Peter, Habala Ondrej, Hluchý Ladislav. Transformation regression technique for data mining. In IEEE International Conference on Intelligent Engineering Systems. 2016, vol., art. no. 7555134. P. 273–277.
12. Prasanna Sahoo: Probability and Mathematical Statistics, University of Louisville. 2013. P. 497–584. http://www.math.louisville.edu/~pksaho01/teaching/Math662TB-09S.pdf
13. Krishnamoorthy K. Statistical Tolerance Regions: Theory, Applications, and Computation, 2009, John Wiley and Sons. P. 1–6.
14. Matlab, Statistics and Machine Learning Toolbox Functions, https://www.mathworks.com/help/stats/functionlist-alpha.html

***About Authors*:**

Peter Krammer,
graduated from the Faculty of Electrical Engineering and Information Technology,
Slovak University of Technology in Bratislava,
and is currently Researcher at the Institute of Informatics of the Slovak Academy of Sciences.
His research interests include data mining and machine learning.
He is (co-)author of several scientic papers and has participated in international and national research projects.

Marcel Kvassay,
is a research scientist at the Institute of Informatics of the Slovak Academy of Sciences.
He graduated from the Faculty of Electrical Engineering and Information Technology in 1991 and in 2017 earned his PhD in applied informatics from the Faculty of Informatics and Information Technologies of the Slovak University of Technology in Bratislava.
Prior to joining the Institute in 2009 he worked at various positions as a software engineer, software design coach and software process improvement manager. His research interests include causal analysis, complex systems, intelligent and knowledge-based technologies, data mining and machine learning.

Ladislav Hluchý,
is the Head of the Department of Parallel and Distributed Information Processing
at the Institute of Informatics of the Slovak Academy of Sciences.
He received M. Sc. and Ph.D. degrees, both in Computer Science.
He is R&D Project Manager and Work-package Leader in a number of 4FP, 5FP, 6FP and 7FP projects,
as well as in Slovak national R&D projects.

Organization:

Institute of Informatics
Slovak Academy of Sciences
Dúbravská cesta 9
845 07 Bratislava, Slovakia.
E-mails: peter.krammer@savba.sk,
marcel.kvassay@savba.sk,
ladislav.hluchy@savba.sk