International Journal of Scientific Research & Engineering Trends
Volume 4, Issue 6, Nov-Dec-2018,ISSN (Online): 2395-566X

Supervised learning in data mining using Transformation Regression Technique

M. Tech. Scholar Karan Bahal
Dept. of Mechnical Engg.
Bharati Vidyapeeth's College of Engineering
Delhi, India

Abstract - **The presented paper focuses on supervised learning in data mining and machine learning areas for small data sets. In the paper, the precision of data mining regression model is increased by special transformation technique, which transforms the original regression task into a new regression task, equivalent with the original. In the paper, transformation was successfully applied on synthetic and real data sets with positive results.**

Keywords-**data mining, regression, supervised learning, data transformation, ensemble learning, model**

I. INTRODUCTION

In the real world information plays an important role, influencing safety, production efficiency and determining the behavior of various subjects. Today big data are collected in many domains, and modern database systems allow effective storing and fast access to this data. Also, the ever increasing performance of computers allows the execution of complex processing and calculations in very short time. These are very good conditions for the application of data mining and information retrieval in many areas of industry and science.

Big data (with several structures and types - graph data[6], time series, texts, images, records) are taking a lead, which poses a number of problems - their processing is time-consuming, memory demanding, and requires complex parallelization strategies. Also, a common problem is a classes-balancing [13], which is not only problem in big data. In mining with big data, it is possible to choose a representative training set of appropriate size; remaining data will be included into testing set. Train and data manipulation is so much easier and faster.

In some real cases, there is available only a limited amount of data records. The measurement can in fact be technologically difficult or time consuming. But even in these cases we need the most accurate modeling and prediction. This may lead to a higher profit and a higher level of security. In such cases, we face problems related to the representativeness of the data set, or a small number of records for the creation of training and testing sets.

The second stated problem could be partially solved by N-fold cross validation. However, these problems have a completely different character than the problems of processing big data. Because when dealing with big data, we could select a smaller representative data set, which partially avoids some of the problems with big data. But a small data set cannot be cloned to create a larger dataset with higher representativeness and better capability for generalization. Therefore, it is necessary to realize deep data analysis and sensitive pre-processing for getting a full potential contained in the data in case of smaller data sets.

II.ENSEMBLE LEARNING OVERVIEW

In data mining, there is a significant effort to maximally increase model precision. There are several methods for improvement of model quality; very often used are ensemble learning methods. The most common ensemble learning methods are Bagging [7], Boosting (Gadabouts [11]), Stacking [8], Dagging [9] and Additive Regression [10]. Many of them are using methods such as voting, record weighting, or multiple model training. Several ensemble learning methods (such as Boosting, Bagging) were originally designed for classification tasks only.

Later they were extended to applications in the regression task [12, 14]. Some more modern methods, such as Evolutionary Ensembles [1], Multiple Network Fusion [2] and Evolving Hybrid Ensembles of Learning Machines [3], were inspired by original ensemble learning methods. Selected studies [4, 5] analyze the suitability of ensemble methods according to partial models or data characteristics.

These methods benefit from a composite model consisting of several sub -classifiers. Individual sub -classifiers are of different types and they mutually offset their weaknesses. Alternatively, same type sub-classifiers are trained in a row, with changing weights of records, thus better adapting them to problematic records. The presented technique uses a different approach. It uses only one trained model, which predicts several values based on available data. These predictions could be averaged; also probability interval of target value could be estimated.

III. TRANSFORMATION TECHNIQUE

The improvement of modeling is based on transformation of the original regression task into a new regression task by using data transformation. This technique is usable if all attributes are continuous numerical attributes. Application of this technique gives us many advantages - simple idea, possibility of target value interval estimation, increasing the count of records and application of different types of models for regression. Presented data transformation is primary focused on cases without large number of available records; it is not suitable for one hundred thousand records or more.

However, this transformation is useful in cases where several hundreds or thousands of records are available, and we would like to use full information potential of the data. Presented data transformation is significantly increasing the number of records; original data set with *N* records, will be transformed into a new data set with N^2 - *N* records. Similarly, count of input attributes (variables) from original data set will be multiplied by 2. The idea of such data transformation is based on the following principle. Traditional machine learning is used to quantify the relation between input attributes and the target attribute. Relations, which are between two inputs attributes, are an alysed during pre-processing phase, usually by correlation analysis.

But the data set could contain relations which are more complex. It could be relation between changing one attribute with changing target attribute. The incremental learning in the training process uses only one record in one moment. So training process with incremental learning quantifies only relation between input attributes and the target attribute Using 2 records together in same cycle of training process allows us to take into account not only values of 2 records, but also a difference between their values. In a classification task, we are able to observe pairs of records and their classes. It could help to identify attributes with large influence on the target class. Calculation of attribute difference of two records could be used for quantifying the measure of distance or similarity of records in the specified attribute. In regression task, it is also possible to calculate difference of the target attribute.

So we are able to observe influence of input attributes and their changes on the target attribute and its change. Similar approach is used in some lazy models, such as k-nearest neighbors (KNN) model. KNN calculates the distance for a pair of records; smaller distance of records represents higher similarity level. High similarity level for record pair indicates high probability that the analyses records are in the same class. In our case of regression task, the target attribute is a continuous variable, so high similarity level (small distance) indicates small difference calculated from target attribute for the analyzed pair of records. However, other regression model types usually do not use differences of values between 2 records at the same time in the training process. It is understandable because comparison of each record pair is very time-consuming, especially for large-scale data. If we do not have too large data set, we are able to tolerate this, especially if we need to maximize the quality of the model. The presented data transformation therefore uses the principle of taking into account two records from the original dataset in one training cycle.

Differences between the same attribute of 2 records are used to represent changes of record pairs. Small difference indicates high level of similarity. Also, differences are better suitable for representation of small relative changes in attributes. It allows us to train a more sensitive model.

1. Definition of Transformation

The definition of a data transformation is very easy to understand. Let us have the original data set, in the form of records for the regression task, with continuous numerical attributes only. This data set represents data after integration process and attributes selection process. So, we expect that all input attributes are relevant to the target attribute. Structure of this original data set is shown in Table I. In Table I, data set contains only 2 input attributes - X and Y; it is shown only as a simple demonstration of the transformation. Of course, real data sets contain usually many more input attributes and records (N = 4 in Table I).

From the original data set (structure in Table I) were taken all possible record pairs, except pairs with same, duplicated records. For each pair of records are calculated differences of all attributes. Calculated values are used in the transformed data form, which is shown in Table II.Each pair of records from original data set defines one record in the new data set. So, count of records will be increased from *N* to N^2 - *N* as mentioned above. Also, count of input attributes will be doubled by the transformation.

Table 1 Structure of Original Available Data Set For Regression Task.

Record Id	Input Attribute X	Input Attribute Y	Target Attribute O
{1}	X1	Y1	O1
{2}	X2	Y2	O2
{3}	X3	Y3	O3

{4}	X4	Y4	O4

Table 2 Structure of Transformed Data Set For Regression Task

Used Record Id	Input Attribute X	Input Attribute Y	vX	vY	vO
{1}, {2}	X1	Y1	X1- X2	Y1 -Y2	O1- O2
{1}, {3}	X1	Y1	X1- X3	Y1- Y3	O1- O3
{1}, {4}	X1	Y1	X1- X4	Y1-Y4	O1- O4
{2}, {1}	X2	Y2	X1-X1	Y2- Y1	O2- O1
--	--	--	--	--	--
{4}, {3}	X1	Y4	X4- X3	Y4- Y3	O4- O3

Data transformation can be easily by pseudo -code. N represents count of records in original data set D, T is the new, transformed data set.

```
for i := 1 to N {
    for j := 1 to N {
        if (i • j) {
            Insert into T a record {xi, yi, …, vi, xi-xj, yi-yj, …, zi-zj} where x, y,… , v, z are attributes in D.
        }
    }
}
Return T.
```

For regression model training we have used the transformed data set, represented by the structure in Table 2As a model any regression model type working with continuous input attributes could be used. Our technique allows using many types of artificial neural networks or regression trees. The trained model could be defined as f() function (1), p represents predicted value, which approximates a new target attribute. The new target attribute is the difference ¨O rather than the original attribute O. This causes several changes in comparison with traditional model training, primarily in the prediction phase.

$$p = f (X, Y,X,Y) \qquad (1)$$

Model, with symbolic representation (1), which was trained on data with structure in Table II, gives a prediction about the change of attribute O between two records.

2. Prediction by Model

It is important to note that the trained model will predict a difference of variable ¨O instead of original target variable O. To apply our model to record {A} (shown in Table III), it is necessary to apply the same data transformation. Original record from Table III marked as {A}, will be transformed into form shown in Table IV.

Table3 One Record {A}, Given To Prediction Process

Record ID	Input Attribute X	Input Attribute Y	Target Attribute O
{A}	X_A	Y_A	0_A

Table 5.One Record {A} Transformed Into Specified Form for Prediction Process

Used Records Id	Input Attribute X	Input Attribute Y	NX	NY	NO
{A}, {1}	X_A	Y_A	$X_A - X_1$	$Y_A - Y_1$	P_{A1}
{A}, {2}	X_A	Y_A	$X_A - X_2$	$Y_A - Y_2$	P_{A2}
{A}, {3}	X_A	Y_A	$X_A - X_3$	$Y_A - Y_3$	P_{A3}
{A}, {4}	X_A	Y_A	$X_A - X_4$	$Y_A - Y_4$	P_{A4}
{1}, {A}	X_1	Y_1	$X_1 - X_A$	$Y_1 - Y_A$	P_{1A}
{2}, {A}	X_2	Y_2	$X_2 - X_A$	$Y_2 - Y_A$	P_{2A}
{3}, {A}	X_3	Y_3	$X_3 - X_A$	$Y_3 - Y_A$	P_{3A}
{4}, {A}	X_4	Y_4	$X_4 - X_A$	$Y_4 - Y_A$	P_{4A}

After data transformation, we are able to apply the trained prediction model f(), which will give us estimations of variable O. Outputs of trained prediction model f() are values p, which approximate values o. For i = 1, 2, ..., N:

$$Pia \cong Oia$$

Or

$$Pia = Oia \pm Er$$

Also, E_r Represents The Error Of The Regression Model F().

$$Oia = Oi - Oa$$

$$Oai = Oa - Oi$$

It is important that $o_{Ai} = - o_{iA}$, however, it is not so exact for approximations p_{iA} and p_{Ai}, because model f() could be nonlinear. We are able to calculate o_A value from (2), (4) and (5), in form (6) and (7).

$$oA = oi - oiA \cong oi - piA \qquad (6)$$

$$oA = oi + oAi \cong oi + pAi \qquad (7)$$

Data transformation generates *2N* new records from one original record {A}. A very positive aspect of this process is the computation of several independent estimations of o_A. It allows us using several strategies for final calculation of the o_A value.

3.Transformation Properties

Presented transformation presents us with several advantages. It increases the number of attributes and records in the training set. Also, it is possible to use any kind of regression model to describe a relation between input attributes and the target attribute. So, the transformation does not restrict choice of model type.

From one record used for prediction {A}, we get several value estimations p_{A1}, p_{A2}, ..., p_{4A}, which approximate the unknown target value o_A. So, we are able to calculate *2N* independent estimations of value o_A. It allows calculation of final o_A value by several strategies.

- Using usual arithmetic average from all *2N* estimations (the most intuitive strategy)
- Using weighted arithmetic average from all *2N* estimations; weights are chosen as by inverse record distance from record {A}.
- Elimination of extremes from estimations (for example 1 minimum, 1 maximum), and calculation of arithmetic average from rest *2N-2* estimations.
- Calculation of record distance, selection k nearest records only for averaging.

Appropriately chosen strategy can greatly improve the accuracy of the model. Also it is possible to estimate the interval of predicted value from several independent predictions. However, presented data transformation has several disadvantages such as higher time and memory requirements. So it is not appropriate for larger data sets.

Two rows in Table II (which are marked {1},{2} and {2},{1}), seem to be collinear; however variables in columns X and Y contain different values. This prevents the linear dependence of rows in the transformed data set.

IV. EXPERIMENTAL RESULTS

The presented transformation technique was tested on generated synthetic data. The synthetic data we have used contain 3 input attributes (marked as $Attr_1$, $Attr_2$ and $Attr_3$). Target attribute O was defined by (8) for training and testing data sets. As a strategy for calculation of final predicted value intuitive arithmetic average of estimations from 2N records was used.Testing data set contains 1000 records. Each input attribute includes integer values 1, 2, ..., 10 evenly (data set contains all combinations of values). Training data set contains random generated real numbers from interval <1, 10>. Training set has 60 records; this number was step by step reduced for performance comparison. In this comparison, we have focused on the performance of the trained models. Our model was an artificial neural network (NN) with one hidden layer and with sigmoid activation function. Learning rate was 0.3 and maximum count of epochs was set to 500. All models were trained in Weka [15].

In the first phase, neural networks were trained by traditional machine learning from original data set with structure corresponding to Table I. Each training model was repeated 5 times with different seed values for initialization of NN.Model performances were averaged from 5 measures with different seeds. Training process was realized for 60 records in training set, and was repeated for smaller count of records.

In second phase, the same strategy (5 repetitions, 60 records and reducing) was used for modelling with data transformed by our transformation. The only difference was the use of the data transformation before model training and also during the prediction phase. Comparisons of observed performances are shown in Figure 1 and Figure 2. Figure 1 shows comparison of model performance (which is represented by correlation coefficient) shown on vertical axis and depending on record count in training set, shown on horizontal axis. Red markers with square shape represent performance of models using data

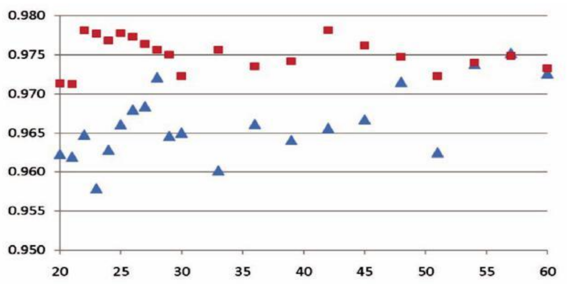

Fig.1Comparison of model performance represented by correlation coefficient on vertical axis depends on number of records in training data set on horizontal axis.

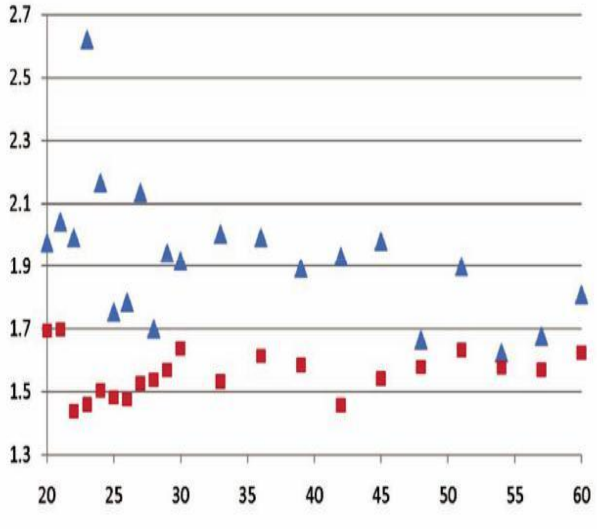

Fig. 2 Comparison of model performance represented by mean squared error on vertical axis depends on number of records in training data set on horizontal axis.

Transformation. Blue markers with triangle shape in Figure 1 represent performance of regression model without data transformation. These models were trained directly on data with structure in Table I for original regression task.

Also, Figure 2 shows comparison of performance depending on number of records in the training data set. However, in this graph, performance is represented by mean squared error. So, smaller value of this criterion represents better model performance.

Overall, models with data transformation reached better performance in both observed criteria. Some isolated points from figures give very similar performance of models with and without data transformation (for example in Figure 1, where record count is 54 or more).

Same approach was used for real data set from energy domain. Energy efficiency data set is available [16]; variable Heating Load was used as target attribute. In this case, 10 repetitions were applied with different seeds; training set has 150 records maximally. Number of records must be higher for generalization in real data case, because real data usually contain significant level of noise.

As a models were used neural network again, with same settings. Reached model performances are compared in Figure 3 and Figure 4 for energy data set. As mention above, red markers with square shape represent performance of models using data transformation. Blue markers with triangle shape represent performance of regression model without data transformation.

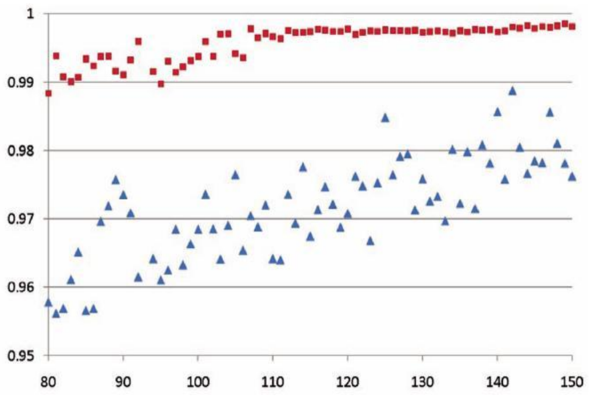

Fig. 3 Comparison of model performance represented by correlation coefficient in real data set set on horizontal axis.

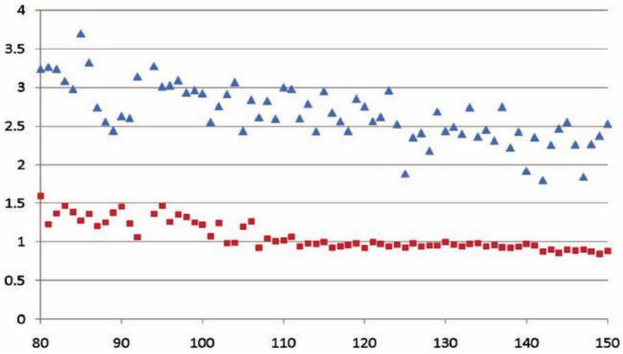

Fig. 4 Comparison of model performance represented by mean squared error in real data set

In this real case, data transformation is stabilizing the performance of model significantly. Even, data transformation gives more significant improvement of results in case for real data set, in compare with synthetic data. Also, it is caused by higher count of records in training data set.

V.CONCLUSIONS

We present a transformation technique usable for increasing of precision of a regression model. Technique is suitable for cases with small data sets and attributes in a real number form. The presented data transformation was so far only applied to a one synthetic and one real data set, but the results show promise. Models with data transformation reached better performance in both correlation coefficient and mean squared error criterions. Also, the proposed data transformation has several advantages. It allows calculation of interval estimation of target values, and supports any type of regression models and provides 4 solid strategies for calculation of the final predicted value. Currently we are working on further experiments with the transformation on other real data sets. Results of these experiments look promising. In future we are planning to apply the presented transformation technique to more real data sets. It allows us to estimate the improvement of model quality more objectively.

REFERENCES

[1] Liu, Y., X. Yao, and T. Higuchi, 2000: Evolutionary Ensembles with Negative Correlation Learning, IEEE Transactions on Evolutionary Computation, 380–387.

[2] Cho, Sung-Bae, and Jin H. Kim, 1995: Multiple Network Fusion Using Fuzzy Logic, IEEE Transactions on Neural Networks, 6(2), 497–501.

[3] Chandra, Arjun, and Xin Yao, 2006: Evolving Hybrid Ensembles of Learning Machines for Better Generalisation, Neurocomputing, 69(7–9), 686–700.

[4] Jain, Anil K., Robert P. W. Duin, Jianchang Mao: Statistical Pattern Recognition: A Review, 2000, IEEE Transactions on Pattern Analysis and Machine Intelligence, 4 –37.

[5] Dietterich, Thomas G.: An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization, Machine Learning 2000, 40(2), 139–157.

[6] Mojžiš, J., Laclavík, M.: Relationship Discovery and Navigation in Big Graphs, In Intelligent Systems in Science and Information 2014, pp. 109-123, Springer International Publishing 2015.

[7] Leo Breiman: Bagging predictors. Machine Learning, 1996, 24(2): 123-140.

[8] David H. Wolpert: Stacking - Stacked generalization, Neural Networks 1992, 5:241-259.

[9] Ting, K. M., Witten, I. H.: Stacking Bagged and Dagged Models, In Fourteenth international Conference on Machine Learning, San Francisco, CA, 367-375, 1997.

[10] J.H. Friedman: Stochastic Gradient Boosting, 1999.

[11] Yoav Freund, Robert E. Schapire: Experiments with a new boosting algorithm. In: Thirteenth International Conference on Machine Learning 1996, San Francisco, 148-156.

[12] Jane Elith, John Leathwick: Boosted Regression Trees for ecological modeling, 2016, https://cran.r-project.org/web/packages/dismo/vignettes/brt.pdf

[13] Zhuoyuan Zheng, Yunpeng Cai, Ye Li: Oversampling method for imbalenced classification, Computing and Informatics, Vol. 34, 2015, 1017-1037.

[14] Matthias Schonlau R.: Boosted Regression (Boosting), The Stata Journal 5, Number 3, 2005, pp. 330 - 3654, http://www.stata-journal.com/sjpdf.html?articlenum=st0087

[15] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten: WEKA Data Mining Software, 2009; SIGKDD Explorations, Volume 11, Issue 1.

[16] UCI, Machine Learning Repository; Energy efficiency Data Set, available: https://archive.ics.uci.edu/ml/datasets/Energy+efficiency