Handling the Problem of Unbalanced Data Sets in the Classification of Technical Equipment States

Obukhov Egor
Perm National Research Polytechnic University - Electrotechnical Department Komsomolsky Ave. 29, 614990, Perm, Russia
E-Mail: obuhov2014@bk.ru

***Abstract*—Questions of handling unbalanced data considered in this article. As models for classification, PNN and MLP are used. Problem of estimation of model performance in case of unbalanced training set is solved. Several methods (clustering approach and boosting approach) considered as useful to deal with the problem of input data.**

***Keywords:* unbalanced data, probabilistic neural net, multilayer perceptron, classification, evaluation of performance, preparation of data.**

I. Introduction

This article deals with comprehensive estimation of the technical state of complex technical systems, based on a structure-approach and further analysis and dynamic monitoring of the structure elements. Most of all a result of technical state estimation of the equipment is determined by selecting the most informative parameters of monitoring: vibration, pressure, temperature etc. Obviously, it is important to make a correct diagnostic model to develop good methods for the recognition of different states of the technical equipment [1].

Thereby, to develop a high-quality model it is necessary to use the most significant parameters of monitoring and to have a representative data set that means the data set have to give comprehensive information about possible states of the monitored equipment.

II. About Data and Problem of It

The technical equipment has special detectors to track and monitor parameters. The values of the parameters are taken from detectors at a certain frequency and are transmitted to a database. Therefore, the values transmitted in the database are used as dataset to train model of prediction future state of the equipment.

The data set has around 2000 instances and a very large dimension. It has advantages such as absence of missing values, but also disadvantages. The main disadvantage is that the data set is unbalanced. This means there is a big bias in the amount of instances of the classes. For example if we have two classes, the first one contains 100 instances and the second one 1000 instances, so the first class is a minority class of rare instances, the other is a majority class.

Most machine learning algorithms work better when both classes of the training set has an equal number of elements. If the number of instances of one class is far different from the other, then problems appear. This is the best illustrated with following example.

If consider the current data set (parameter monitoring) and train a machine learning algorithm on the data, suppose there are two possible outputs as follows:

1) 10 instances of minority class and 20 of the majority class are misclassified.
2) 2 instances of minority class and 60 of the majority class are misclassified.

If we calculate the performance of the model by the amount of misclassified examples, then obviously that first case is better. However in terms of correct classification of minority class (especially when the class has priority), then second case is the best choice. Thereby it is important to have a correct metric for efficient estimation of the model performance.

Unbalanced data sets [2] are a special case for classification problems. This type of sets supposes a new challenging problem for Data Mining, since standard classification algorithms usually consider a balanced training set. So the question is how to handle with it?

The goal of the article is to find a way of handling with unbalanced data sets and improve the performance of the unbalanced data sets classification.

For that goal, it is necessary to create a representative and high-quality training set.

III. Approaches to Handle Unbalanced Data

There are different ways to handle unbalanced data. Let's consider several of those, which could be useful for the specific task:

1) In the first approach, it is necessary to divide the major class into L distinct clusters, then train L classifiers, where each classifier is trained on only one of the distinct clusters, but on all of the data from the minority class. To be clear, the data from the minority class are used in the training of all L classifiers. Finally, use ensemble of the L learned classifiers as a final classifier.

2) This is similar to number (1), but a little different. Let N be the number of samples in the minority class. Cluster the majority class into N clusters (agglomerative, K-means

clustering etc.), and use the resulting cluster mediods as the training data for the majority class. To be clear, you throw out the original training data from the majority class, and use the mediods instead. Finally, the classes are balanced.

3) The third one is based on boosting. The algorithm trains the first learner, L1, on the original data set. The second learner, L2, is trained on a set on which L1 has around 50% chance to be correct. The third learner, L3, is trained on the cases on which L1 and L2 disagree. As output, return the majority of the classifiers. Why it improves, the classification can be found in [3].

In case of application this method on the unbalanced data. Let L1 return always true (majority class). L2, is being trained, where L1 inconvenient. L3 trained, when L1 and L2 disagree, that is L2 predicts false (minority class). Therefore, false (minority class) prediction can be only when both L2 and L3 predicts false.

Whatever method one can use will help in some ways, but hurt in others. To improve the efficiency, one can train separate models using all of the methods listed above, and then perform model averaging over all of them.

The methods are not applied yet, and on the stage of research, the task was to find out the ideas and approaches, which could be useful to the task. In future, it is necessary to apply the methods to prove the efficiency and suitability of it.

IV. Data preparation (Normalization and feature selection)

The data are recorded from detectors of the technical equipment and has a large dimensionality around 80 attributes. In order to reduce the dimensionality of the data and find out the effective number of variables, we applied the following approaches:

- Correlation analysis

Take our 80 attributes, one of them is label, so we have around 80 pairs (label + one, every time different attribute). A correlation is a number between -1 and +1 that measures the degree of association between two attributes, which allows us to estimate the degree of association between label and one certain attribute. The table 1 shows the interpretation of correlation values.

TABLE I
Interpretation of correlation values

Negative Value (V)	Positive Value (V)	Interpretation
-0,2 <V< 0,2		Very low correlation
-0,5<V<-0,2	0,2<V<0,5	Low correlation
-0,7<V<-0,5	0,5<V<0,7	Middle value of correlation
-0,9<V<-0,7	0,7<V<0,9	High correlation
V < -0,9	V > 0,9	Very high correlation

By calculating the correlation coefficient for each of the pairs, it is possible to make some filtering and delete less correlated attributes.

- Principal component analysis (PCA)

PCA is method to reduce the dimensionality of the variable (attribute) set, by using a new coordinate system that is lesser in dimension than the number of original variables. This transformation will usually be accompanied by a loss of information. The goal of PCA is to preserve as much information contained in the data as possible.

- Global sensitivity analysis

It is a tool from Statsoft "Statistica" [4], which gives information about the relative importance of the variables used in a neural network. In sensitivity analysis, one determines how the neural net will response (increasing or decreasing error rates) to some changes of its input variables. During the analysis, this tool exclude an attribute and make training of classifier without the attribute. If an important attribute excluded then error increase significantly. If an unimportant attribute excluded, the error will not increase very much.

- Normalization

As a classifier to solve, the task of determination of states of technical equipment was chosen a neural net. In this case, it is necessary to make some normalization of the training data, because in such type of classifier small values of a variable (like 0.5) and big values (like 100) have different influence on the final prediction. It means one has to reduce the range of data to a limit between 0 to 1. The limit depends of activation function type.

The method we applied to normalize the data: Statistical normalization.

The main idea of the method is to convert the data into a set with normal distribution with mean=0. The formula of statistical normalization is $Z=(x-u)/s$ (x-current value, u-mean value of the variable, s- standard deviation).

V. Choosing classification algorithms (modeling)

As models for classification we have chosen two neural nets: probabilistic neural net and multilayer perceptron. We will consider their distinctive features and find out why they are well suited for this kind of task [5].

MLP:

A multilayer perceptron (MLP) is a typical common kind of neural nets and a good fit to almost all types of tasks. It is feedforward neural net, so we have several successive connected layers and each current neuron gets and processes signals from neurons of the previous layer. The input layer is used just to transfer information into a hidden (computing) layer. The amount of neurons in the input and output layers is determined by the dataset parameters (number of attributes and type of target attribute). More information about working of this type of neural nets available in [6].

PNN:

A probabilistic neural net (PNN) is some kind of a neural net which similar to a radial basis function (RBF) network. The main difference is that a PNN has one neuron (in hidden layer) for each point in training set, that means the kind of net requires more memory space to store the model then MLP. It is not a good match to work with large training sets. PNN is quite accurate with small to medium size data sets and it has modifications, such as matrix of loss, which add an additional layer with rates (price) of classification errors. The modification allows to make some accent on one class, the possibility fits well to the current task of determination state of an engine. More information about the structure of hidden layers and activation functions available in [7].

If we consider the question of tuning the model's parameters [8], then compared to PNN (where is only one parameter to choose), a MLP has the following parameters:

- Amount of neurons of hidden layer

A way used to calculate amount of neurons in the layer: (number of attributes + number of classes) / 2 + 1

- Training cycles
- Learning rate
- Momentum

For automatically tuning the parameters and finding the best combination of them, we used the statistica program tools. It allows to set some limits for parameters and to train specified number of nets. As a result, we have nets ranked by performance.

VI. Metric to Estimate the Performance of Models

As was said before, such typical metric as classification accuracy is not a good metric, because if a model correctly classify just instances of majority class, then the model have high accuracy by using the metric of estimation.

In the case, when the minority class is a class which represents fault states of the technical equipment and it is more important to accurately classify the vectors of the class, than vectors of the other class.

That is why, the confusion matrix was chosen as a metric to determine the quality of the model. The matrix allows to estimate the recall of a specific class and getting a clearer representation of the model efficiency.

To calculate the recall of the minority class in the data set (monitoring parameters of technical equipment) it is necessary to split the dataset into 3 parts: training set, validation set, testing set. The first step is to train the classifier with data from the major part, which is more than 50 % of whole data set. The second step is to make some intermediate validation of model efficiency following by tuning the model. The last step is a final one-time testing. It allows to make more accurate estimation of the model performance, because testing on a validation set is some kind of training (changing parameters of a model and retraining the model). The results of the final testing are shown in table 2.

Calculation of classification accuracy:

$$Accuracy = \frac{tp+tn}{tp+tn+fp+fn} = \frac{323+2130}{323+2130+132+57} = 0{,}92$$

Therefore, the result is an accuracy equal to 92%, which is a good result for classification. However, the method of estimation does not represent the whole efficiency of the model.

Calculation of the recall of class True:

$$Recall = \frac{tp}{tp+fn} = \frac{323}{323+57} = 0{,}85$$

Comparison of classification results PNN and MLP:

The result is 85 percent, that means class True is recognized with probability of 85 percent. So with propability of 15 percent a failure or failts will not be recognized, that's not enough for effective operation of a system.

Training the models (MLP and PNN) on unbalanced data/ unrepresentative data showed, performance/accuracy of the models are equal, however it is not enough for effective performing of the current task of technical state of technological equipment determination.

TABLE II
Confusion Matrix

		Real values / classification	
		True	False
Predicted values / classification	True	tp = 323	fp = 132
	False	fn = 57	tn = 2130

VII. Conclusion

In the paper it was explained what unbalanced data are and what influence it has on the performance of a system to classify different states of a technical equipment. Principal approaches of handling unbalanced data are discussed. Making a representative training set by using k-means clustering is shown. Several machine-learning algorithms such as probabilistic neural net and multilayer perceptron have been chosen as models to deal with the task of determination of different states of the technical equipment.

The traditional methods of diagnosis and control of parameters of the technological equipment are time-consuming and it doesn't present a possibility for express analysis. The results of the approaches presented in the article can be useful in some difficult situations as additional information to make a good decision.

Future research will be concentrated on applying all of that approaches and methods of handling with unbalanced data. The main issue is to figure out how much does this affect the final performance of the model. It is important to know which type of neural net fits to the case most.

References

[1] Basmanov M., Menshikov S., Morozov I., Strebkov A.," System of parameter's diagnostic of GTU: modern approach" Delovaya Rossia N7, 2011.-42-43 p. (In Russian)

[2] Icamaan B. Viegas da Silva; Paulo J. L. Adeodato, "PCA and Gaussian noise in MLP neural network training improve generalization in problems with small and unbalanced data sets" Proceedings of International Joint Conference on Neural Networks, San Jose,

[3] Robert E. Schapire, "The Strength of Weak Learnability" Machine Learning, 5(2):197–227, 1990

[4] Statsoft program "Statistica". [Online]. Available: http://www.statsoft.ru/

[5] Dolina O.N. and Kyzmin A.K. Particularities of developing expert systems based on neural net modeling" Journal of Saratov technic state university, 2009.-266-272 p. (In Russian)

[6] S. Sathyanarayana, "A gentle introduction to backpropagation", July 22, 2014

[7] Specht, D. F. "Probabilistic neural networks". Neural Networks 3:1990.- 109–118 p.

[8] A.I. Gavrilov and P.V. Evdokimov, "Neural Network optimum parameters determining under industrial process mathematical model construction" Vestnik of ISPU N4, 2007(In Russian)