ATLANTIS PRESS
Advances in Computer Science Research (ACSR), volume 72
IV International Research Conference "Information Technologies in Science, Management, Social Sphere and Medicine" (ITSMSSM 2017)

Combined Optimization and Modified Performance Metrics for Automated Model and Parameter Selection in Telecom Customer Churn Prediction

Stanislav Alkhasov
Department of Information Security Systems
Southern Federal University
Taganrog, Russia
alkhasov@sfedu.ru

Alexey Tselykh
Department of Information Security Systems
Southern Federal University
Taganrog, Russia
tselykh@sfedu.ru

***Abstract*— In this paper, we consider the construction of an optimization algorithm designed to identify, in automatic way, the optimal parameters and algorithms for binary classification in the task of customer churn prediction in telecommunication. The currently supported methods for classification include Decision Trees, k-Nearest Neighbors, Support Vector Machines, and Back-Propagation Artificial Neural Networks. It is shown that in certain cases the standard textbook metrics of classification model quality (e.g. accuracy, precision, recall, and AUC) are not descriptive enough. Thus, we evaluate the algorithms using the modified recall metrics: Weighted Recall metric, and Weighted Recall and Run-Time metric. These metrics are appropriate for use in automatic mode without continuous expert supervision. We use genetic algorithms as optimization algorithms. To improve standard implementations, we introduce a combined optimization approach based on D.H. de Vries model of disaster evolution and the island model.**

Keywords—binary classification; artificial neural network; genetic algorithm; combined optimization; performance metric.

I. INTRODUCTION

Data mining methods have been extensively applied in science, engineering, and business. While binary (two-class) classification problem is quite well studied, challenges remain. No-free-lunch (NFL) theorems have shown that learning algorithms cannot be universally good. One of the challenges is a fast and efficient automatic classification that is robust to changes in data structure as well as to missing values, duplicates, and outliers. One approach to tackle these issues is an optimization algorithm that automatically determines optimal algorithms and parameters as well as data pre-processing techniques for maximum model accuracy.

Each classification task is quite unique. In this paper, we focus on the problem of customer churn prediction in telecommunication. When assessing mobile subscriber behavior, we often deal with class imbalance problem as normal churn rate in telco rarely exceeds 3-5 per cent. Besides, the costs of a type 2 error are dramatically higher than the costs of a type 1 error. [1, 2]

The demand for combined optimization algorithm stems from the following reasons: (i) periodic changes in the structure of analyzed data, (ii) a need for incorporating new algorithms, (iii) an impossibility to build a universal classifier due to NFL-theorem, and (iv) a growing need for automated binary classification.

II. PREVIOUS RESEARCH AND PROBLEM STATEMENT

The recent trends in classification methods for customer churn prediction have previously been highlighted in papers by M.A.H. Farquad [3], A. Rodan [4], Huang Bingquan [5], T. Vafeiadis [6], Huang Ying and T. Kechadi [7], A. Keramati [8] et al. In the earlier stages of our research, we have studied the following basic methods used for binary classification: Decision Trees, k-Nearest Neighbors, Support Vector Machines, and Back-Propagation Artificial Neural Networks. We have evaluated model quality over different sets of parameters, such as the number of neurons in a hidden layer in artificial neural networks (Fig. 1). Ultimately, we obtained the following characteristics of classifiers that are optimal with respect to standard recall measure:

- *C4.5 Decision Tree:* no maximum depth or pruning;
- *K-Nearest Neighbors:* k = 3, Manhattan distance;
- *Support Vector Machines:* polynomial kernel, p = 3, $\gamma = 0$, C = 1000;
- *Artificial Neural Network:* 25 neurons in the first hidden layer, 15 neurons in the second hidden layer, momentum parameter $\alpha = 0.8$.

We found out that accuracy and recall measures alone were not enough to draw any definite conclusions. Precision, F-measure, and AUC are among other basic performance measures used for classification model evaluation. Searching for optimal parameters of a classification model requires manual analysis of all standard metrics. Besides, we need to continuously handle class imbalance, to consider the costs of errors and the running time of algorithms.

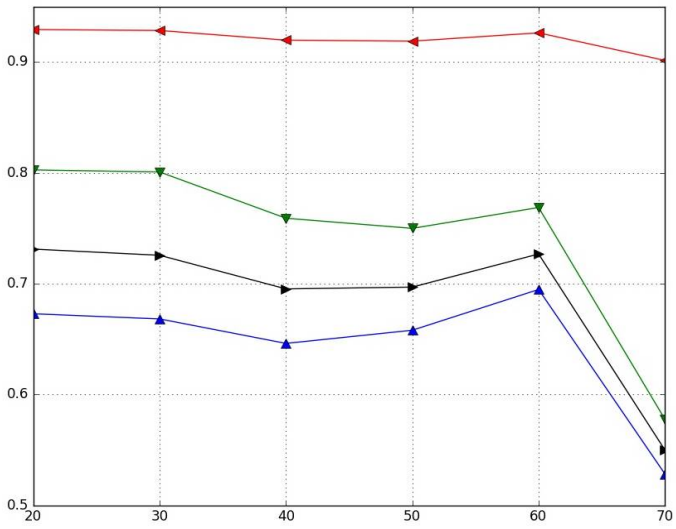

Fig. 1. ANN classification model: graph of dependencies between the number of neurons in a hidden layer and classification quality: red line (◄) – accuracy, green line (▼) – precision, black line (►) – F-measure, blue line (▲) – recall

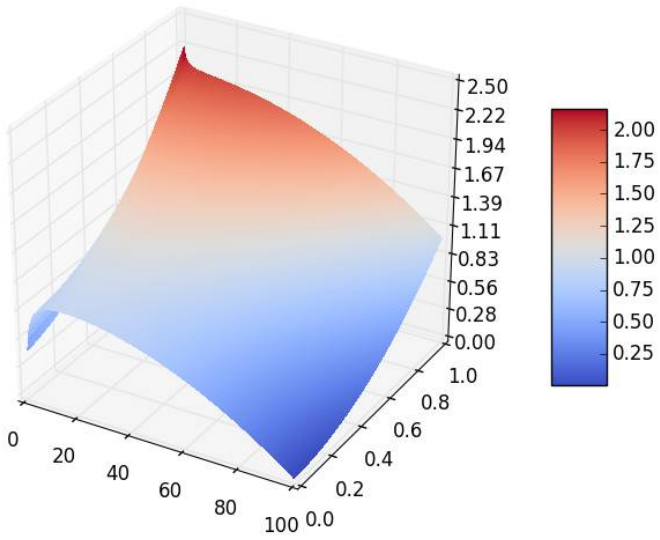

Fig. 2. Graph of dependencies between WRAR metric value, running time of algorithm (scale from 0 to 100) and weighted recall value (scale from 0.0 to 1.0)

III. Introducing Modified Metrics of Classification Model Quality

In this paper, we present modified recall metrics: Weighted Recall metric, and Weighted Recall and Run-Time metric.

Weighted Recall (WR) metric captures the number of False Positives (FP) as well as the ratio of positive to negative training examples. The metric is calculated according to the formula:

$$\mu' = \mu\left(1-\left(\frac{\mathrm{FP}}{\mathrm{FP}+\mathrm{TP}}\right)^{10\delta}\right)$$

$$\delta = \frac{\mathrm{TP}+\mathrm{FN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}}$$

where FP is the number of False Positives, TN is the number of True Negatives.

Weighted Recall and Algorithm Runtime (WRAR) metric additionally considers the runtime of binary classification algorithm (Fig. 2). The metric is calculated according to the formula:

$$\omega' = \exp\frac{\mu'-\mu_0}{\max(t,t_m)}+\mu'^2-\left(\frac{t}{t_0}\right)^2$$

where t is a running time of an algorithm (in seconds), μ_0 is a cutoff value of recall (generally accepted at level 0.8), t_0 is a maximum allowed running time (accepted at level 100), t_m is an adjustment to minimum running time (accepted at level 1).

We evaluated classification methods based on the values of standard recall and modified metrics (Table 1). One can see that the greatest differences in the values of standard recall and modified metrics are observed in cases of kNN classification (1NN is unstable under certain scenarios, e.g. noise in the input data) and polynomial kernel SVM when $\gamma = 0$ (making it a constant). Here, the introduced metrics are more informative in comparison to standard recall.

Data preprocessing included dummy encoding, feature selection using Add-Del algorithm, and data shuffling. The obtained results are presented in Table 2.

IV. Automatic Parameter Selection Using GA

Optimal parameter selection based on theoretical and empirical knowledge as well as depth-limited exhaustive search has certain drawbacks: a) it's time- and resource-consuming, b) it requires constant expert supervision to interpret interim results, and c) there's a considerable risk of premature convergence on a local extremum. To overcome these limitations, we propose an automated approach

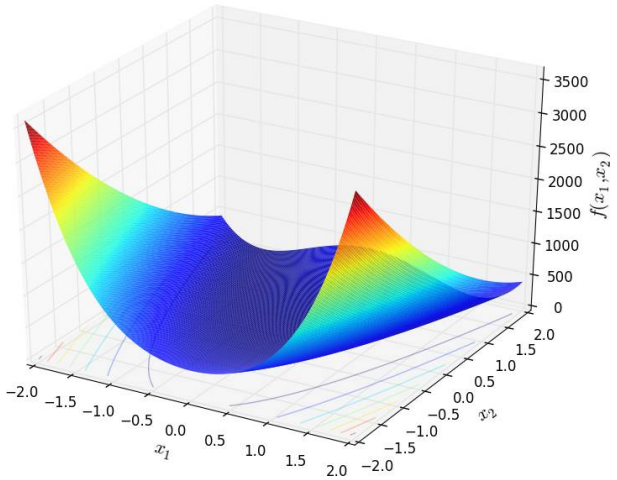

Fig. 3. Example of a test function: Rosenbrock function of two independent variables

TABLE I. EVALUATION OF CLASSIFICATION METHODS BASED ON THE STANDARD RECALL AND MODIFIED METRIC

Method		**C4.5 Decision Tree**	**kNN**		**Support Vector Machine**	**ANN**
Parameters		***No maximum depth or pruning***	***Manhattan distance***		***Polynomial kernel, $p = 3$, $C = 10^3$, $\gamma = 0$***	***25 and 15 neurons in hidden layers, $\alpha = 0.8$***
			k = 1	***k = 3***		
Metrics	μ	70.77%	45.29%	39.90%	85.09%	72.69%
	μ'	55.29%	28.98%	31.79%	19.94%	63.85%
	ω'	2.12	0.80	0.90	1.15	4.26

TABLE II. WEIGHTED RECALL METRIC FOR CLASSIFIERS USING INITIAL SAMPLING AND MODIFIED SAMPLING USING ADD-DEL ALGORITHM

Method / **Sampling**	**C4.5 Decision Tree**	***k*NN: $k = 3$**	**SVM-RBF: $C = 55$, $\gamma = 150$**	**ANN: $\alpha = 0{,}8$**
Initial	55.29%	31.79%	46.05%	63.85%
Modified using Add-Del	58.68%	38.17%	41.48%	68.72%

to parameter selection using GA. One of its main advantages is there are no substantial mathematical requirements to the type of objective function [9-12].

To assess GA algorithms on the development stage, we used benchmark functions, i.e. sphere, Rastrigin, Rosenbrock, and Ackley [13-15]. The main advantage of these functions is their fast performance. Ackley function is specifically appropriate to assess basic trends during GA modification. The number of independent variables in a benchmark function was equal to the number of classification model parameters being optimized (Fig. 3).

We analyzed optimization efficiency of modified genetic algorithms from simple to complex. Initially, we considered a Standard GA [16, 17]. We used the following selection operators: (i) tournament, selection, (ii) simple two-point cross with probability 0.75, and (iii) Gaussian mutation with probability 0.8. Standard GA was not effective in finding global extrema of Rastrigin and Rosenbrock functions.

Further, we modified Standard GA based on the exploration and exploitation strategy:

- Exploration is based on the mutation operator and allows finding previously unknown search space (avoiding local extrema).
- Exploitation is based on crossover operator and allows improving initial results (reaching optima with a high accuracy) [18, 19].

Hence, we modified Standard GA in the way that probabilities of crossover and mutation change after the one half of desired generations. The modified algorithm proceeds from high probability of mutation in the beginning to high and modified metrics are observed in cases of kNN probability of crossover in the end. This algorithm has shown effectiveness for all the functions but Rosenbrock.

Furthermore, we suggested another approach to Standard GA modification based on the shift of the dominant genetic operator depending on a fitness function. This algorithm has reached a global extremum of Rosenbrack function in 6.9 per cent of tests. We did not aim at to build a universal optimization algorithm that effectively optimizes all the test functions. We concluded, this algorithm was not good enough for our classification problem. [16, 20].

The main problem with the preceding algorithms is that as a result of the evolution the species are starting to look the same. Consequently, the algorithms prematurely converge to a local minimum. To tackle this problem, we turned to D.H. de Vries model of disaster evolution and the island model.

Genotypic catastrophism, a modified model of disaster evolution, relies on k-means++ method suggested by D. Arthur and S. Vassilvitskii [21] to cluster species into several clusters. We determined an optimal ratio of the clusters to the species in the population at 1/4-1/5 that corresponds to the minimum average complexity of the algorithm. This algorithm performs better than phenotypic catastrophism but is not that effective in comparison to non-GA approaches.

Another approach to prevent premature convergence is to divide species into several separate parts, i.e. islands. In its standard implementation, an island algorithm does not show high capability for optimization in the task of binary classification.

Furthermore, we suggested the following modification of the island algorithm. For each island, the probabilities for mutation and crossover are given as a set of random values. In this way, each island has a specialization within the exploration-exploitation strategy. It was proved through experiment, that higher frequency of migrations between islands leads to higher average complexity of the algorithm (under reduced diversity of species, the island model degenerates into a Standard GA).

Modified algorithm (73.4%) outperformed both the basic island algorithm (61.21%) and the best of non-GA approaches (68.72%). To study an effect of changes in the structure if classified data, we finally considered the combination of island and catastrophic models.

The integration of these two approaches can be interpreted as non-equiprobable occurrence of disasters (i.e. removal of some species) in the separate parts (islands) of the discontinuous habitat. This approach minimizes the impact of the death of most species that could be the predecessors of the solutions close to optimal. During the experiment, we determined an optimal ratio of disasters to the islands at 1/10.

In the binary classification task, we used a combined algorithm with the value of minimum frequency of disasters set to 60 and the number of disasters in the generation set to 10. By applying the combined optimization algorithm, we obtained an optimal selection of parameters for binary classification leading to the most efficient churn prediction.

The set of parameters is as follows:

Data preprocessing: ANN classifier, min-max normalization, data shuffling in a training set, ninefold cross-validation.

Feature selection for churn prediction: «The total number of voice messages», «The total amount of calling time in the afternoon», «The total number of calls to customer services», «International calling plan», «Calling rates in the evening », etc.

Parameters for the classification algorithm: 27 and 19 neurons in the first and second hidden layers, momentum of 0.73, learning rate of 0.0018, and hyperbolic tangent as an activation function.

V. Conclusions

In this study, we obtained the following results.

We surveyed a set of methods for binary classification to predict churn. We have also shown the specificity of the task due to the dominance of true positive recognitions of non-loyal customers. It is shown that the basic standard performance metric in this case is recall.

We introduced two new performance metrics – Weighted Recall and Weighted Recall metric and Algorithm Runtime metric. The first one provides an assessment of the influence of the number of false positive recognitions of non-loyal consumers and the general imbalance of the analyzed data. The second one considers the running time.

We proposed a strategy for automating the selection of the main parameters of the classifiers, as well as characteristics of the analyzed data (normalization, shuffle of objects of the sample, etc.) using genetic algorithms.

We developed an improved version of D.H. de Vries model of the evolution of disasters that is used for the modification of the Standard GA in order to maintain an effective genetic diversity of the population during iterations (generations).

We developed a combined genetic algorithm incorporating catastrophic and island models. This algorithm allows building classifiers in order to solve, more efficiently, the churn problem compared to traditional methods of data analysis.

The resulting set of parameters allows to provide a qualitative assessment of customer loyalty. The architecture of the classifier in the analysis of customer loyalty of telecommunication services is the following: back propagation artificial neural network with 27 and 19 neurons in hidden layers, hyperbolic tangent as the activation function, learning rate and momentum equal to 0.73 and 0.0018, respectively. We used min-max data normalization and data shuffling. The number of folds in the cross-validation set is equal to 9. We also identified the optimal subset of 9 features.

In the process of development, we used IPython Notebook as IDE, Python 2.7 as a programming language, Scikit Learn as a library for machine learning and DEAP as a library for evolutionary computation [22].

Acknowledgment

The reported study was supported by the grant of Southern Federal University, research project No. 07/2017-28.

References

[1] N.B. Paklin and V.I. Oreshkov, Biznes-analitika: ot dannykh k znaniyam: Uchebnoe posobie. SPb: Piter, 2013.

[2] R.R. Veynberg, Intellektual'nyy analiz dannykh i sistem upravleniya biznes-pravilami v telekommunikatsiyakh: Monografiya. Moscow: INFRA-M, 2016.

[3] M.A.H. Farquad, V. Ravi, and S. Bapi Raju, “Churn prediction using comprehensible support vector machine: An analytical CRM application,” Applied Soft Computing, 2014, vol. 19, pp. 31-40.

[4] A. Rodan, A. Fayyoumi, H. Faris, J. Alsakran, and O. Al-Kadi, “Negative Correlation Learning for Customer Churn Prediction: A Comparison Study,” The Scientific World Journal, 2015, pp. 1-7.

[5] B. Huang, M.T. Kechadi, and B. Buckley, “Customer churn prediction in telecommunications,” Expert Systems with Applications, 2012, vol. 39, pp. 1414-1425.

[6] T. Vafeiadis, K.I. Diamantaras, G. Sarigiannidis, and K.Ch. Chatzisavvas, “A comparison of machine learning techniques for customer churn prediction,” Simulation Modelling Practice and Theory, 2015, vol. 55, pp. 1-9.

[7] Y. Huang and T. Kechadi, “An effective hybrid learning system for telecommunication churn prediction,” Expert Systems with Applications, 2013, vol. 40, pp. 5635-5647.

[8] A. Keramati, R. Jafari-Marandi, M. Aliannejadi, I. Ahmadian, M. Mozaffari, and U. Abbasi, “Improved churn prediction in telecommunication industry using data mining techniques,” Applied Soft Computing, 2014, vol. 24, pp. 994-1012.

[9] Yu.V. Chernukhin and M.A. Belyaev, “Osobennosti ispol'zovaniya geneticheskikh algoritmov pri obuchenii pertseptronov,” Izvestiya TSURE, 2001, vol. 4(22), pp. 134-140.

[10] V.I. Bozhich, L.A. Gladkov, V.M. Kureychik, and Yu.L. Shnitser, “Razrabotka sistemnykh printsipov postroeniya ehvolyutsionnykh instrumental'nykh sredstv formirovaniya i obucheniya neyronnykh setey,” Izvestiya TSURE, 2001, vol. 4(22), pp. 182-186.

[11] L.M.L. de Campos, R.C.L. de Oliveira, and M. Roisenberg, “Optimization of neural networks through grammatical evolution and a genetic algorithm,” Expert Systems with Applications, 2016, vol. 56, pp. 368-384.

[12] Nauchnaya sessiya MEPhI–2007. IX Vserossiyskaya nauchno-tekhnicheskaya konferentsiya «Neyroinformatika–2007»: Lektsii po neyroinformatike. Chast' 2. MEPhI, 2007.

[13] L.A. Gladkov, V.V. Kureychik, and V.M. Kureychik, Geneticheskie algoritmy / Pod red. V.M. Kureychika. Moscow: FIZMATLIT, 2006.

[14] D. Rutkowska, M. Piliński, and L. Rutkowski, Neyronnye seti, geneticheskie algoritmy i nechetkie sistemy. Moscow: Goryachaya liniya – Telekom, 2006.

[15] M.V. Burakov, Geneticheskiy algoritm: teoriya i praktika: ucheb. posobie. SPb: GUAP, 2008.

[16] V.V. Kureychik, V.M. Kureychik, and S.I. Rodzin, Teoriya ehvolyutsionnykh vychisleniy. Moscow: FIZMATLIT, 2012.

[17] Bionicheskie informatsionnye sistemy i ikh prakticheskie primeneniya / Pod red. L.A. Zinchenko, V.M. Kureychika, V.G. Red'ko. Moscow: FIZMATLIT, 2011.

[18] D. Whitley, "A genetic algorithm tutorial," Statistics and Computing, 1994, vol. 4(2), pp. 65-85.

[19] H.M. Pandey, A. Chaudhary, and D. Mehrotra, "A comparative review of approaches to premature convergence in GA," Applied Soft Computing, 2014, vol. 24, pp. 1047-1077.

[20] S.S. Alkhasov, A.N. Tselykh, and A.A. Tselykh, "An Integrated ANN-GA Approach to Data Classification," Proceedings of the 2016 Conference on Information Technologies in Science, Management, Social Sphere and Medicine (ITSMSSM 2016), 2016, pp. 172–176.

[21] D. Arthur and S. Vassilvitskii, "k-means++: The Advantages of Careful Seeding," Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms (SODA '07), 2007, pp. 1027-1035.

[22] L.P. Coelho and W. Richert, Building Machine Learning Systems with Python. Packt Publishing, 2015.