http://dx.doi.org/10.5755/j01.eee.19.6.4578

A Multi-Population Genetic Algorithm for Inducing Balanced Decision Trees on Telecommunications Churn Data

V. Podgorelec[1], S. Karakatic[1]
[1]*University of Maribor, Faculty of Electrical Engineering and Computer Science,*
Smetanova 17, SI-2000 Maribor, Slovenia
vili.podgorelec@uni-mb.si

Abstract—In this paper we present a new approach to predicting telecommunications churn. Churn prediction can be considered as a multi-objective optimization problem, where the accuracy of predicting both churning and staying consumers need to be optimized simultaneously. As the existing classification methods failed to produce balanced solutions, we developed a new multi-population genetic algorithm for the induction of decision trees. By introducing multiple populations, linear ranking selection and adequate fitness function we were able to avoid overly biased solutions. The evaluation results of our algorithm's performance in comparison with the existing methods show that it was able to find highly accurate and balanced solutions.

***Index Terms*—Classification algorithms, genetic algorithms, telecommunications churn**

I. INTRODUCTION

Telecommunications churn – moving to a different telecommunications company – today is still a major deal within companies. Having to understand why a customer chooses to go for other company is crucial in finding flaws in the product-range or services. As more and more data about the consumer get stored, trying to find relations why he/she churned is becoming more and more interesting.

Data mining methods are being often used to discover interesting patterns within data. Classification is one of the most common data mining tasks, with the aim of classifying unknown cases based on a set of known examples into one of possible classes. Considering the telecommunications churn problem, the aim of classification is to learn to predict whether a consumer will move to a different company based on the consumers' data stored within company's database.

The term customer attrition simply refers to the customers leaving one business service to another. Customer or subscriber churn is similar to attrition, namely the process of customers switching from one service provider to another [1]. From a machine learning perspective, churn prediction is a supervised classification problem defined as follows: given a set of data, describing each customer's behaviour in the network (attributes) together with the information whether the customer has switched to another company, the goal is to predict the future churners using the values of this same set of attributes [2].

For the purpose of this study, a publicly available telecommunications churn problem dataset was used [3]. The input (a learning set) for this problem includes the data on past calls for each subscriber (such as day, eve and night calls, minutes, charge, etc.), together with all personal and business information maintained by the service provider (state, area code, service calls, etc) and a churn label. The aim of classification here is to build a classification model, which is able to accurately predict whether some subscriber will churn or not based on new, unseen data. To evaluate the performance of a built classifier, a test set is used containing consumers not used in the learning phase.

There are a lot of different classification methods available, including instance-based classifiers, Näive Bayes, support vector machines, neural networks, etc. One of the most popular and often used is decision tree (DT), which is easy to use, fast, quite robust classifier, providing a transparent classification model that is easy to interpret, evaluate and validate. Although there are several methods available for the statistical induction of DTs, it has been recognized that the use of alternative methods, such as evolutionary algorithms, can improve the performance of induced DTs [4], especially in the case of noisy, imbalanced or skewed datasets. As the used telecommunications churn dataset is a complex and very imbalanced dataset, the use of an alternative method for producing highly accurate solution was indeed necessary, as will be shown later.

II. MULTI-POPULATION CROSS-SELECTION GENETIC ALGORITHM FOR INDUCING DECISION TREES

Genetic algorithm (GA) is a non-deterministic evolutionary optimization method, used to solve all kinds of complex optimization problems [5]. GA is inspired by the principle of natural evolution. It uses a population of individuals (candidate solutions) which evolve through generations while being subjected to exchange of genetic material (reproduction or crossover), mutation and pressure

Manuscript received January 28, 2013; accepted April 15, 2013.

This paper was produced within the framework of the operation entitled "Centre of Open innovation and ResEarch UM". The operation is co-funded by the European Regional Development Fund and conducted within the framework of the Operational Programme for Strengthening Regional Development Potentials for the period 2007 – 2013, development priority 1: "Competitiveness of companies and research excellence", priority axis 1.1: "Encouraging competitive potential of enterprises and research excellence".

to adapt to its environment (selection with regard to fitness). Each individual is evaluated using a fitness function and at each generation fitter individuals have a higher probability of advancing to the next generation and reproducing. Throughout generations, the solution (best individual within population) is being improved towards optimum.

GAs are increasingly being used to evolve DTs because they provide accurate solutions and are able to maintain comprehensibility. Comprehensibility can be assured by evaluating candidate solutions with regard to both the accuracy and the size of the tree [6]. In this study we have designed and evaluated a special kind of GA for DT induction, namely a multi-population cross-selection GA, that was aimed at optimizing DTs with regard to overall prediction accuracy simultaneously with preserving the balance between positive (customers who churned) and negative cases (customers who stayed with a company).

After applying different classification methods for the prediction of the telecommunications churn problem, we realized that all of them produced very imbalanced results. In our previous work we have designed and evaluated various GAs for the induction of DTs [7], [8] with good results, especially on the most complex problems [9]. However, in the case of telecommunications churn problem set they were not able to produce the expected results and were not much different than the rest of the classification methods used in our experiment – they all optimized overall accuracy at the expense of low classification accuracy of positive cases (customers who churned), which were a real minority among all cases (less than 15%).

A. *Multi-population genetic algorithm*

GAs are able to explore a wide range of search space when the selection pressure is properly controlled, while crossover works as a constructive operator towards local optima and mutation as a destructive operator in order to keep the needed genetic diversity. The evolutionary search for the solution is directed towards the optimal solution based on the defined fitness function.

In our case, the optimal solution is the best DT. However, evaluation of a DT is generally multi-objective, as at least overall accuracy, average class accuracy, precision and recall for each decision class etc. influence the quality of a DT. Usually, fitness function in such cases is defined as a weighed sum of all of the above single measures. The problem here is that the improvement of one single parameter generally leads to the worsening of others. As the solution in evolutionary search needs to build up before it can reach an adequate level of quality, the problem of such aggregated multi-objective fitness functions is that the solution becomes biased for some parameters only.

To overcome this problem, evolutionary search should have the capability of preserving the building blocks needed to optimize the solution towards all of the objectives. In other words, the premature convergence should be avoided. To improve the fine-tuning capability of simple GAs, multi-population genetic algorithms (MPGA) have been used in many applications [10], including data mining [11]. MPGA is an extension of simple GAs by dividing a population into several isolated sub-populations within which the evolution proceeds and individuals are allowed to migrate from one subpopulation to another. In recent years, MPGA have been recognized as being more effective both in speed and solution quality than single-population GAs [11].

We propose a MPGA for the induction of DTs that includes two subpopulations (Fig. 1). Both subpopulations are equally sized (containing N individuals) and have the same initialization procedure, crossover, and mutation operators according to *genTrees* algorithm [7]. Each subpopulation has a different fitness function, however: while both penalize individuals for misclassification errors, size and complexity, the first optimizes DTs regarding the overall accuracy, and the second optimizes DTs regarding the balanced class accuracy.

Unique in our case is the selection operator, called cross-selection, which works in the same manner for both subpopulations (it is explained in detail in the next subsection). Before selecting individuals from the first subpopulation, a subset of $\frac{N}{8}$ individuals is selected using a linear ranking selection algorithm from the second subpopulation, which is then migrated and added to the first one. After that, a normal linear ranking selection algorithm is applied on such increased population (now containing $\frac{9N}{8}$ individuals). After selection, N new individuals are produced using crossover and mutation operator. The same principle is then used also for the second subpopulation.

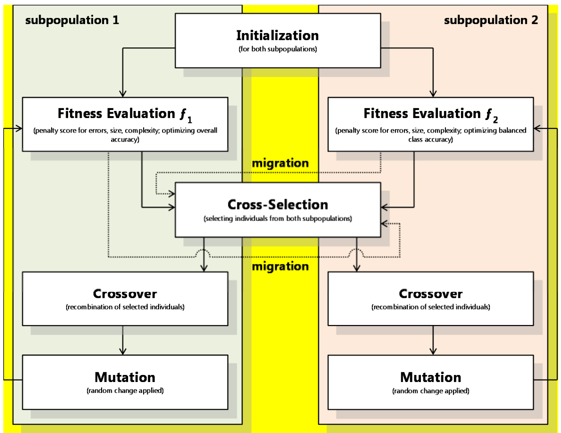
subpopulation 1
subpopulation 2
Initialization
(for both subpopulations)
Fitness Evaluation f_1
(penalty score for errors, size, complexity; optimizing overall accuracy)
Fitness Evaluation f_2
(penalty score for errors, size, complexity; optimizing balanced class accuracy)
migration
Cross-Selection
(selecting individuals from both subpopulations)
migration
Crossover
(recombination of selected individuals)
Crossover
(recombination of selected individuals)
Mutation
(random change applied)
Mutation
(random change applied)

Fig. 1. A general outline of a multi-population cross-selection genetic algorithm with two subpopulations.

B. *Linear ranking cross-selection operator*

An especially important genetic operator in the case of MPGA is selection. It has turned out that the most popular selection algorithms like tournament, proportional and roulette wheel selection are not the best choice in our case. Namely, as the improvement in the subpopulation 1 was faster than in the subpopulation 2 – it is easier to optimize the overall accuracy at the beginning – the preservation of diversity was too low; the individuals from subpopulation 1 largely displaced the individuals from subpopulation 2. For this purpose, we decided to use an adapted version of linear ranking selection operator that improved this behaviour.

Linear ranking selection has been introduced by Grefenstette and Baker [12] to overcome some deficiencies of proportional selection [13]. Individuals are sorted according to their fitness score, the best having rank 1 and the worst rank 1. The probability of selecting an individual is determined linearly according to its rank

$$p_i = \frac{1}{N}\left(\eta^- + (\eta^+ - \eta^-)\frac{i-1}{N-1}\right), \qquad i\epsilon\{1,..,N\}. \tag{1}$$

The probability of selecting the worst individual is $\frac{\eta^-}{N}$ and the probability of selecting the best is $\frac{\eta^+}{N}$. As the population size is constant, $\eta^+ = 2 - \eta^-$ and $\eta^- \geq 0$. It has to be emphasized, that all the individuals have different ranks, although some of them may be identical or may have exactly the same fitness score.

The expected fitness distribution Ω^* is the expected distribution of fitness scores of individuals within a population after using the selection operator. In the case of linear ranking selection with η^- on the distribution s, the expected fitness distribution is

$$\Omega_R^*(s,\eta^-)(f_i) = s^*(f_i) = \\ = s(f_i)\frac{N\eta^- - 1}{N-1} + \frac{1-\eta^-}{N-1}(S(f_i)^2 - S(f_{i-1})^2). \tag{2}$$

Let $\bar{s}$ be a continuous distribution of individuals' fitness scores within a population. The expected fitness distribution of individuals after performing linear ranking selection $\bar{\Omega}_R$ with η^- on the distribution $\bar{s}$ is

$$\bar{\Omega}_R^*(\bar{s},\eta^-)(f) = \bar{s}^*(f) = \\ = \eta^- \cdot \bar{s}(f) + 2\cdot\frac{1-\eta^-}{N}\cdot \bar{S}(f)\bar{s}(f). \tag{3}$$

Generally, $\eta^- = \frac{1}{N}$ is used. In our case of multi-population cross-selection, some amount of individuals from one subpopulation migrates to the other subpopulation. We experimentally decided to transfer $\frac{N}{8}$ of individuals, selected using the linear ranking selection, from one subpopulation to another. In this manner, the population size N in a single subpopulation increase to $N^* = N + \frac{N}{8} = \frac{9N}{8}$.

The reproduction rate $R(f)$ is the ratio between the number of individuals achieving a certain fitness score f before and after selection. An useful selection operator has to favour fitter individuals which should have the reproduction rate $R(f) > 1$, while the less fit individuals should have $R(f) < 1$. The reproduction rate of linear ranking selection is

$$\bar{R}_R(f) = \eta^- + 2\cdot\frac{1-\eta^-}{N}\bar{S}(f). \tag{4}$$

This equation shows that the least fit individuals have the lowest reproduction rate $\bar{R}(f_0) = \eta^-$ and the most fit individuals have the highest reproduction rate $\bar{R}(f_n) = 2 - \eta^- = \eta^+$, which can be essentially derived from the method construction as the probability of selecting the least fit individual is $\frac{\eta^-}{N}$ and the probability of selecting the most fit one is $\frac{\eta^+}{N}$.

The loss of diversity p_d is defined as a share of individuals within population who are left out during selection. The loss of diversity should be as low as possible, as the bigger loss of diversity leads towards premature convergence. As stated above, the problem of premature convergence was one of the major problems we wanted to avoid. The loss of diversity using the described linear ranking selection (regarding one subpopulation) is

$$p_{d,R}(\eta^-) = \frac{1}{4}(1-\eta^-). \tag{5}$$

III. THE EXPERIMENTS AND RESULTS

For the evaluation of the described method, a publicly available telecommunications churn problem dataset has been used [3]. It consists of 3333 cases (consumer's descriptions), described with 20 attributes (15 numeric and 5 nominal) and two decision classes (true and false – whether a consumer churned or not). More than 85% consumers did not churn, which makes the dataset an imbalanced one.

All experiments have been performed using 10-fold cross-validation. After applying different classification methods for the prediction of the telecommunications churn problem, we realized that all of them produce very imbalanced results. As all of the classifiers optimized the overall accuracy, the results were strongly biased towards accurate predictions of negative cases, which were the vast majority. This fact was the reason for us to upgrade our GA-based DT induction algorithm *genTrees* to a MPGA. For the comparison within this experiment, the best DTs from both subpopulations have been used (GT#1 and GT#2).

A comparison of a simple GA and MPGA for the induction of DTs is presented in Fig. 2. It can be easily seen how a simple GA converges prematurely (reaching its optimum after approx. 400 generations) while MPGA continues to evolve with a significant improvement after 800 generations. The isolation of second subpopulation enabled it to evolve enough to contribute to a resulting DT, which was not possible with a single-population simple GA, where the faster improvement of overall accuracy suppressed the slower evolution of balanced class accuracy.

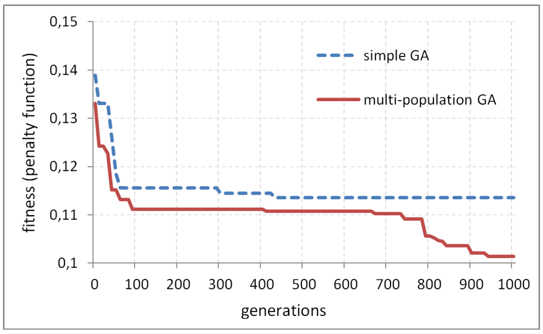

Fig. 2. Evolution comparison of a simple GA vs. multi-population GA.

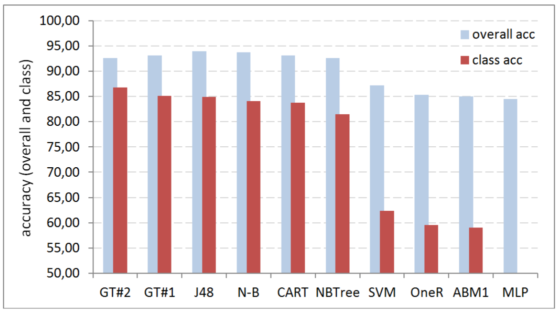

Fig. 3. Comparison of accuracy on a test set for different classifiers (overall accuracy and average class accuracy).

Comparison of classification performance for different classification methods is presented in Fig. 3. While the classifiers are sorted according to average class accuracy, it can be seen that also the overall accuracy correlates well with this measure. The best results were achieved by both MPGA solutions (GT#2 and GT#1), followed by J48, Näive Bayes, simple CART and Näive-Bayes tree. Traditionally very strong classifiers like support vector machine (SVM), neural network (MLP, multilayer perceptron) or ensemble method (ABM1, Ada Boost) achieved much worse results, both in overall and also average class accuracy.

As the problem of telecommunications churn prediction can be considered as a multi-objective optimization problem with the aim of optimizing the accuracy of predicting both the consumers that churned and the ones that stayed, a Pareto front for several classifiers, considering both TP (true positives, accuracy of classifying true cases, i.e. consumers who churned) and TN (true negatives, accuracy of classifying false cases, i.e. consumers who stayed) rates is represented in Fig. 4. It can be seen that only three solutions can be potentially considered as the best solution: SVM, J48 and GT#2. Although SVM scored slightly best TP rate, it is not a reasonable solution with its TN rate of less than 30%. From the other two, J48 was less than 3% better than GT#2 regarding TP rate but almost 7% worse regarding TN rate. Finally, GT#1 is also very near to the Pareto front.

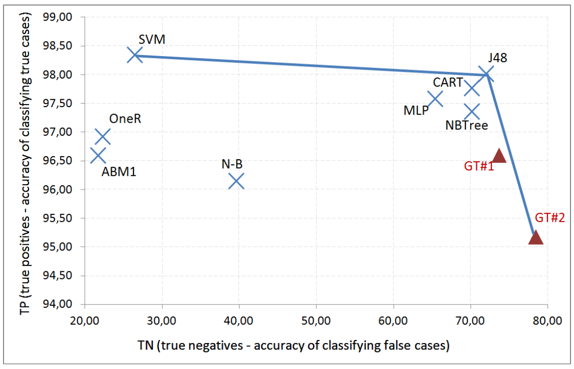

Fig. 4. Pareto front for several classifiers, considering both TP (true positives) and TN (true negatives) classification accuracy.

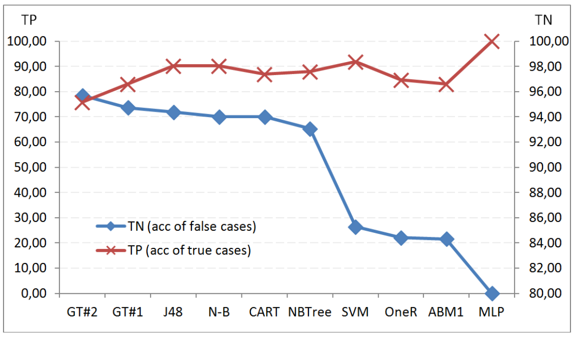

Fig. 5. Comparison of TP (accuracy of classifying true cases) and TN (accuracy of classifying false cases) rates for different classifiers.

The balance between TP and TN rates for different classifiers is presented in Fig. 5, where it can be seen that our MPGA method indeed produced the most balanced solutions (GT#2 and GT#1). While the difference in classifying true cases between best and worst solution is very small (only 4.82% between GT#2 and MLP), the difference in classifying false cases is rather huge (73.65% between GT#2 and that same MLP).

IV. CONCLUSIONS

In this paper we presented a new multi-population genetic algorithm (MPGA) for the induction of decision trees, which has been applied and thoroughly evaluated on the problem of predicting telecommunications churn. The obtained results show that our solution was able to overcome one of the major problems of the existing classification methods – providing high overall accuracy as well as balanced class accuracy. The introduction of multiple populations has enabled evolutionary search to maintain the needed diversity and thus avoiding premature convergence. In order for the MPGA to work an adaptation of linear ranking selection operator has been performed in order to cross-select individuals from both subpopulations. The evaluation results show that MPGA scored the best both in average class accuracy and in classifying true negatives, while achieving third rank in overall accuracy.

At this point the parameters for fine-tuning evolutionary search have been determined experimentally. In the future we plan to perform exhaustive analysis of MPGA performance in regard to different parameter settings. Additionally, we will test our algorithm on different datasets to gain further insights into its capabilities and performance.

REFERENCES

[1] V. Umayaparvathi, K. Iyakutti, "Applications of Data Mining Techniques in Telecom Churn Prediction", *International Journal of Computer Applications*, vol. 42, no. 20, pp. 5–9, Mar. 2012.

[2] Y. Richter, E. Yom-Tov, N. Slonim, "Predicting customer churn in mobile networks through analysis of social groups", in *Proc. of the Int. Conf. Data Mining (SIAM)*, 2010.

[3] D. T. Larose, *Discovering Knowledge in Data: An Introduction to Data Mining*, Hoboken, NJ: John Wiley and Sons, Inc., 2005.

[4] R. C. Barros, M. P. Basgalupp, A. C. P. L. F. de Carvalho, A. A. Freitas, "A Survey of Evolutionary Algorithms for Decision-Tree Induction", *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 42, no. 3, pp. 291–312, 2012. [Online]. Available: http://dx.doi.org/10.1109/TSMCC.2011.2157494

[5] J. H. Holland, *Adaptation in natural and artificial systems*. Ann Arbor, MI: University of Michigan Press, 1975.

[6] V. Podgorelec, M. Šprogar, S. Pohorec, "Evolutionary design of decision trees", *WIREs Data Mining Knowl Discov*, submitted for publication. (DOI: http://dx.doi.org/10.1002/widm.1079).

[7] V. Podgorelec, P. Kokol, "Evolutionary induced decision trees for dangerous software modules prediction", *Information Processing Letters*, vol. 82, no. 1, pp. 31–38, 2002. [Online]. Available: http://dx.doi.org/10.1016/S0020-0190(01)00284-8

[8] V. Podgorelec, "Expert-Assisted Classification Rules Extraction Algorithm", in *Proc. of the 14th East European Conf. (ADBIS 2010)*, LNCS, 2010, vol. 6295, pp. 450–462.

[9] V. Podgorelec, P. Kokol, M. Molan Stiglic, M. Hericko, I. Rozman, "Knowledge discovery with classification rules in a cardiovascular dataset", *Computer Methods and Programs in Biomedicine*, vol. 80, no. 1, pp. S39-S49, 2005. [Online]. Available: http://dx.doi.org/10.1016/S0169-2607(05)80005-7

[10] W.-Y. Lin, T.-P. Hong, S.-M. Liu, "On adapting migration parameters for multi-population genetic algorithms", in *Proc. of the IEEE Int. Conf. Systems, Man, Cybernetics*, 2004, pp. 5731–5735.

[11] H. Zhu, J. Licheng, P. Jin, "Multi-population genetic algorithm for feature selection", in *Proc. of the 2nd Int. Conf. Natural Computation*, Xi'an, China, 2006, pp. 480–487. [Online]. Available: http://dx.doi.org/10.1007/11881223_59

[12] J. J. Grefenstette, J. E. Baker, "How genetic algorithms work: A critical look at implicit parallelism", in *Proc. of the 3rd Int. Conf. Genetic Algorithms*, Morgan Kaufmann Publishers, San Matteo, CA, 1989, pp. 20–27.

[13] D. Whitley, "The GENITOR algorithm and selection pressure: Why rank-based allocation of reproductive trials is best", in *Proc. of the 3rd Int. Conf. Genetic Algorithms*, Morgan Kaufmann Publishers, San Matteo, CA, 1989, pp. 116–121.