HYPERPARAMETER TUNING WITH CARET FOR AUTHOR NAME DISAMBIGUATION

George TALABĂ
Al. I. Cuza University of Iași, Romania
george.talaba.edu@gmail.com

Abstract. *Author identification is a complex problem that affects the quality of content in the knowledge management systems and digital libraries. The main challenge is to correctly assign the author of scientific papers for ensuring correctness and reliability of any analysis. The difficulties come from: the lack of a universally accepted standard, authors sharing the same name, or the use of short names and initials. In this paper we present a Machine Learning-based method for tuning the hyperparameters on predictive models used for eliminating the author ambiguity. The models are built using the publishing profile extracted from the existent work, academic affiliation, research domain and meta attributes like email address, ORCID, or ResearchID. Initial experiments were obtained using CARET for tuning the models created by Classification And Regression Tree and Conditional Inference Tree methods.*

Keywords: CART, hyperparameter, predictive models, tuning, text mining, CTREE
JEL classification: C38
DOI: 10.12948/ie2019.03.08

1. Introduction

In digital libraries is at the same time important and difficult, from both practical and theoretical perspectives, to correctly identify the author of a particular paper. Trustworthy query results are needed for measuring publications' impact, individual promotions, or approving grants. The main problems are the lack of a universally accepted standard, the decentralized generation of content, or polysemy of the names.

Using the taxonomy defined by Ferreira [1] we can group the methods for eliminating the ambiguity into two main categories, depending on the type of approach: author grouping methods and author assignment methods. In this paper we will focus on the author assignment method, which attempts to directly assign bibliographic references to a particular author using an automated classification method [2], [3].

The hyperparameter tuning technique is used for improving the precision of such methods. The most common hyperparameter optimization techniques are: grid search, random search, differential evolution, and Bayesian optimization.

In this paper we propose a method for hyperparameter optimization on the predictive classification methods using an automatic parameter tuning technique. The optimization is performed for two classification tree algorithms, CART **C**lassification **A**nd **R**egression **T**ree (CART) and **C**onditional inference **TREE**s (CTREE).

2. Related work

According to Nair [4], the hyperparameter optimization is very useful, but at the same time it is not clear when one hyperparameter optimizer is better than another. Each method has several parameters which can impact the computed model. It is very difficult to tune only one parameter, since it might be efficient for a training set and unreliable for another [5].

Being impractical to explore all the possible combinations of parameters and settings in a classification technique, only the top performing classifiers should be evaluated. An automated parameter optimization technique can be used for exploring the impact of several parameters with different settings [6]. An off-the-shelf automated parameter optimization technique is CARET [7], which is analyzing multiple settings in the parameter space, proposing an optimal combination.
One advantage of such an automated system is the possibility of parallel predictor evaluation across different worker nodes in a grid environment. This offers better scalability for large data sets, with increased training efficiency [8].
Another option is to use a framework like ANOVA [9], that can detect the importance of both individual hyperparameters and interaction between different parameters. This framework also allows the evaluation of one hyperparameter prediction power across multiple datasets [10].
Not always using a grid search automated tuning method delivers the optimal results. Bergstra [11] proved that random experiments are more efficient than grid experiments for hyperparameter optimization. Random experiments are easier to be used, being asynchronous and with a better scalability and survivability.

3. Setup and data

Our data was extracted from Web of Science - Clarivate (WOS) database and has 98,926 records. From this data we've extracted attributes like author's data, category data, activity domain information, affiliation, and the journal/proceedings volume where the paper was published.
For building the training data set we used several heuristic methods. The meta attributes like email, Open Researcher & Contributor ID (ORCID) and ResearcherID (RI) were considered as having a high plausibility. The author affiliation and the name uniqueness were other attributes used to construct the training set.
Based on this data, we built predictive models for identifying the researchers affiliated to Al. I. Cuza University of Iași (UAIC). Models were built using CART and CTREE classification algorithms.

4. Methodology, platforms, tools

The entry data for each researcher had some metadata like email address, Open Researcher and Contributor ID (ORCID), or ResearchID and a publishing profile based on unigrams and bigrams found in the paper's titles.
The experiments were executed and visualized with R [12] and RStudio (https://www.rstudio.com) platforms using several packages. The predictive models were built using two classification methods: CART from *rpart* package [13] and CTREE from *partykit* package [14].
We evaluated the hyperparameter tuning using CARET [7], identifying the setting which achieves the highest performance during model building. The parameters setting tuning was performed using the *train* function from CARET package. The discrimination power was measured using *Area Under the receiver operator characteristic Curve* (AUC) [15].
The testing process included five steps.
(Step 1) Create the baseline prediction model using optimal value from theoretical perspective.
(Step 2) Generate the control parameter setting for CARET *train* method using a number of different values for each parameter.
(Step 3) Evaluate the parameter setting using CARET, which will explore all the parameter combinations and a specified number of repetitions.
(Step 4) Identify the optimal setting with the highest estimated performance using CARET

(Step 5) Compare the classifier performance using the AUC method for the predictive models created as baseline and with the CARET-optimized setting.
In the case of *CART*, the parameter tuned parameter is CP (complexity parameter) that imposes a penalty on decision-trees with too many splits. The baseline value for this parameter was obtained using the "one standard error rule" [16], that finds the minimum number of splits for all trees having the cross-validation error within one standard deviation from the minimum value.
For CTREE we tuned the maximum depth of the tree and for baseline we use the default value "Inf" (-inite), which implies that the resulted model will have no limitation on the number of splits.

5. Results

Using these optimization methods, several prediction models were compared from a performance point of view using the Confusion Matrix produced by the *CARET* package [7]. We compared several scores for identifying the best tuning strategy. The scenario 1 is the baseline and it was built using the theoretical method of "one standard error rule".

Table 1. Tuning methods and hyperparameters used for CART

Scenario	Optimization	Method	Parameters	Accuracy	Sensitivity	Balanced Accuracy
1	Theoretical	-	CP=0.000540	0.9488	0.8379	0.8996
2	Repeated k-fold Cross Validation	rpart	CP=0.002159361	0.95	0.8123	0.8889
3	Repeated k-fold Cross Validation	rpart2	maxdepth=4	0.9332	0.7036	0.8313
4	Leave One Out Cross Validation	rpart	CP=0.002375297	0.9392	0.8123	0.8829
5	Leave One Out Cross Validation	rpart2	maxdepth=4	0.9332	0.7036	0.8313
6	k-fold Cross Validation	rpart	CP=0.002159361	0.95	0.8123	0.8889
7	k-fold Cross Validation	rpart2	maxdepth=4	0.9332	0.7036	0.8313

Based on the results, we can observe that some of the tuned models have a better performance than the baseline. However, the tuned models are having a worse balanced accuracy on the validation training set that was used. Another observation is that tuning the depth of the classification tree is producing the same results for the different optimization techniques tested. Also, the performance of the models built using the depth hyperparameter is worse than the performance of models built using the complexity parameter.
For validating the results we've used the ROC (Receiver operating characteristic) analysis. The comparison was done between the model selected by using the theoretical approach and the one having the best performance after tuning (scenario 2 and 6).

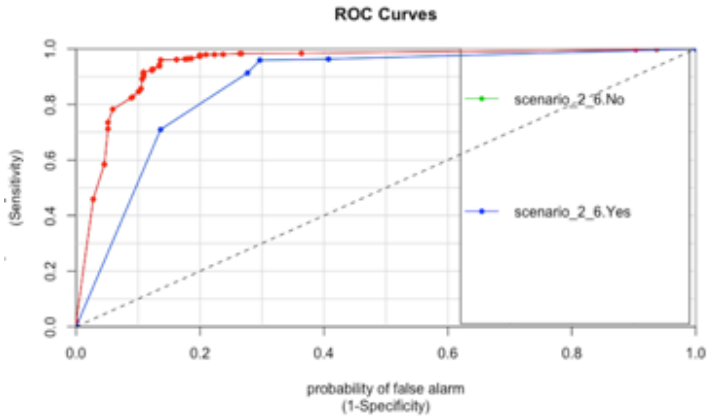

Figure 1. ROC analysis for selected models (where red is the theoretical model)

Based on this this analysis, the prediction model created using the theoretical method has better performance than the ones resulted from hyperparameter optimization. Considering that the ROC analysis decouples the classifier performance from class skew and error costs, it can produce more accurate results.
As last step we compared the variable importance for the selected models. We can see on both models that the research domain has the biggest predictivity, followed by the short name of the author. The differences are marginal between the two selected models on the importance of each variable.

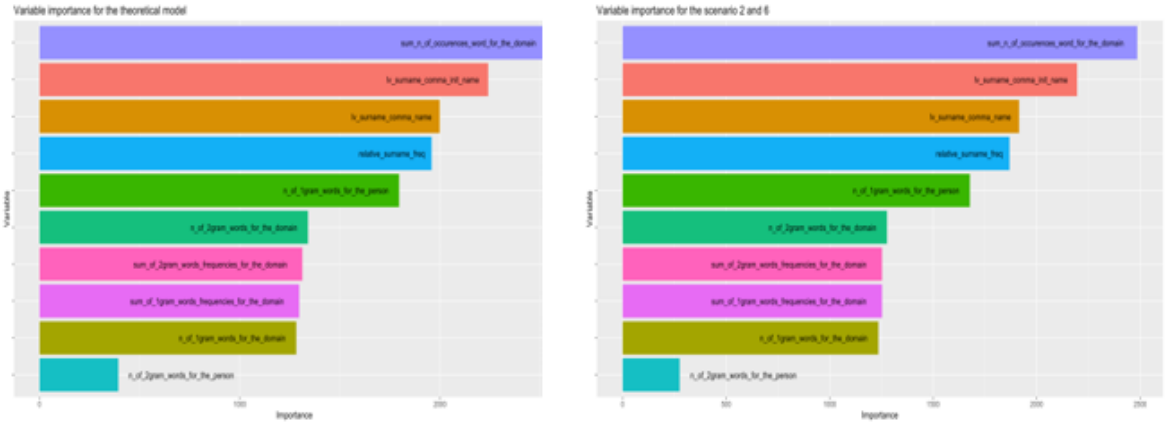

Figure 2. Variable importance for the selected models

On the other hand, the models created using the hyperparameter tuning with CTREE algorithm had worse performance, per Confusion Matrix, than the ones created with CART. Also, tuning the depth of the tree is not producing better predictive models than using the default values. To overcome this limitation, we've used a different tuning method which is selecting a different hyperparameter.

Table 2. Tuning methods and hyperparameters used for CTREE

Scenario	Optimization	Method	Parameters	Accuracy	Sensitivity	Balanced Accuracy
1	Theoretical	-	maxdepth=Inf	0.9486	0.9626	0.8934
2	Repeated k-fold Cross Validation	ctree2	maxdepth=10	0.9486	0.9626	0.8934
3	Leave One Out Cross Validation	ctree	mincriterion=0.99	0.9454	0.9582	0.8951
4	k-fold Cross Validation	ctree	mincriterion=0.01	0.946	0.9504	0.9287

The ROC analysis invalidates these findings, since the optimal model created with the CTREE method has better performance than the one created with CART.

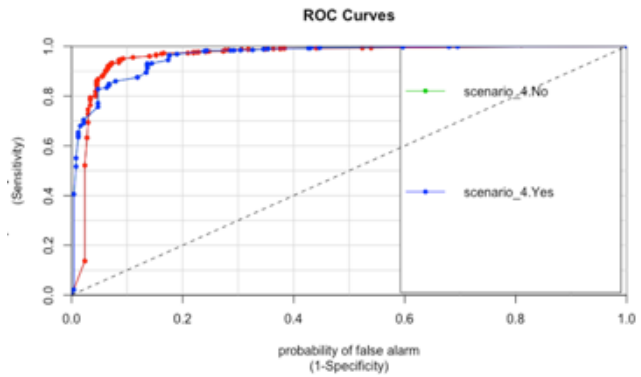

Figure 3. ROC analysis for selected models (where red is the theoretical model)

This can be partly explained if we check the variable importance for these models where the author name has the biggest impact.

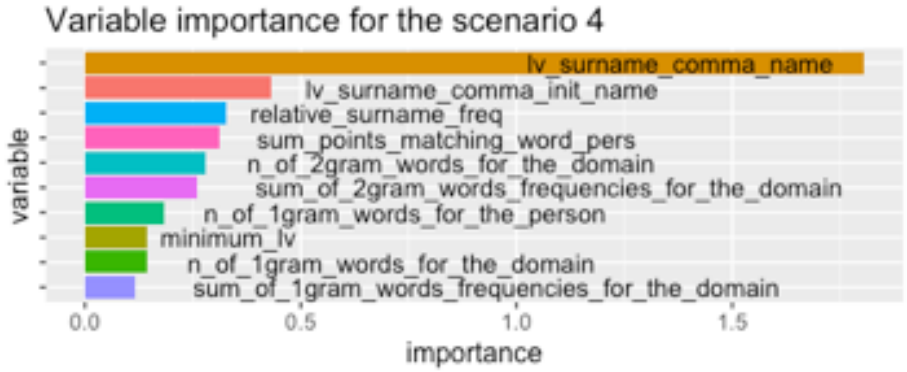

Figure 4. Variable importance for the scenario 4

6. Conclusions and further research

There is a plethora of work on hyperparameter optimization, using all kinds of methods and algorithms. This paper argues that the tuning methods can produce better results than using the random or theoretical approaches. By using an algorithmic approach, we can produce predictive models that can accommodate the change in the data set structure or volume. Also, this offers the ability to test more variables for identifying a better predictive model.

The results showed that the tuning of hyperparameters can result in better predictive models. But the theoretical approach should not be ignored, since it is based on years of research and can be a good starting point.

We cannot conclude that one method is better than the other due the different analysis' results. While some models have better accuracy, they can exhibit a worse ROC curve or sensitivity.

Therefore, based on the existent results, we cannot pick one hyperparameter as being the one which will improve the prediction performance for most of the models.

One conclusion is that the training data set might not have sufficiently diverse data. The variables related to author name have a high importance in all models, leading to the conclusion that we need to extend the author coverage. By extending the training data set to cover several institutes we can build a larger training data set. This will result in models more stable and less sensitive to missing data problems or larger data sets.

In future work, we plan to further increase the list of hyperparameters using various configurations specific to R implementation of these optimizations. We also plan to check the impact of using combinations of hyperparameters instead of isolated parameters.

Acknowledgment

Al. I. Cuza University of Iasi, GRANT 12521: Platforma de servicii software privind centralizarea raportarii activitatii stiintifice la nivel UAIC, prin integrarea datelor referitoare la publicatii si proiecte de cercetare.

References

[1] A. A. Ferreira, M. A. Gonzcalves and A. H. Laender, "A Brief Survey of Automatic Methods for Author Name Disambiguation", SIGMOD Record, vol. 41, no. 2, pp. 15–26, August 2012.

[2] A. A. Ferreira, A. Veloso, M. A. Goncalves and A. H. Laender, "Effective Self-Training Author Name Disambiguation in Scholarly Digital Libraries" in Proc. of Joint Conf. on Dig. Lib., 2010.

[3] A. A. Ferreira, R. Silva, M. A. Goncalves, A. Veloso and A. H. Laender, "Active Associative Sampling for Author Name Disambiguation" in Proc. of Joint Conf. on Dig. Lib., 2012
[4] Huy Tu and Vivek Nair, “Is one hyperparameter optimizer enough?” in Proc. of the 4th ACM SIGSOFT International Workshop on Software Analytics (SWAN 2018). ACM, New York, NY, USA, pp. 19-25.
[5] James Bergstra, Rémi Bardenet, Yoshua Bengio and Balázs Kégl, “Algorithms for hyper-parameter optimization” in Proc.of the 24th International Conference on Neural Information Processing Systems (NIPS 2011), Curran Associates Inc., USA, pp. 2546-2554.
[6] C. Tantithamthavorn, S. McIntosh, A. E. Hassan and K. Matsumoto, "Automated Parameter Optimization of Classification Techniques for Defect Prediction Models," 2016 IEEE/ACM 38th International Conference on Software Engineering (ICSE), Austin, TX, 2016, pp. 321-332.
[7] M. Kuhn. “caret: Classification and regression training”. [Online]. Available: http://CRAN.R-project.org/package=caret, 2015. [Accessed March, 2019].
[8] Patrick Koch, Oleg Golovidov, Steven Gardner, Brett Wujek, Joshua Griffin and Yan Xu, “Autotune: A Derivative-free Optimization Framework for Hyperparameter Tuning” in Proc. of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD 2018). ACM, New York, NY, USA, pp. 443-452.
[9] F. Hutter, H. H. Hoos and K. Leyton-Brown, “An efficient approach for assessing hyperparameter importance” in Proc. of ICML 2014, pp. 754–762.
[10] Jan N. van Rijn and Frank Hutter. “Hyperparameter Importance Across Datasets” in Proc. of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD 2018). ACM, New York, NY, USA, pp. 2367-2376.
[11] James Bergstra and Yoshua Bengio, “Random search for hyper-parameter optimization”. J. Mach. Learn. Res. 13 (February 2012), pp. 281-305.
[12] R Core Team, “R: A language and environment for statistical computing”, R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/, January 13, 2019
[13] rpart package [Online]. Available: https://cran.r-project.org/web/packages/rpart/rpart.pdf, January 21, 2019. [Accessed February, 2019].
[14] partykit package. [Online]. Available: https://cran.r-project.org/web/packages/partykit/partykit.pdf, January 21, 2019. [Accessed February, 2019].
[15] D. J. Hand and C. Anagnostopoulos, “When is the area under the receiver operating characteristic curve an appropriate measure of classifier performance?”, Pattern Recogn. Lett. 34, 5 (April 2013), pp. 492-495
[16] Breiman, L., Classification and Regression Trees. New York: Routledge, 1984. Available https://doi.org/10.1201/9781315139470