Voting with Random Classifiers (VORACE)*

Extended Abstract

Cristina Cornelio
IBM Research
Rüschlikon, Zürich, Switzerland
cor@zurich.ibm.com

Michele Donini†
Amazon
Seattle, USA
donini@amazon.com

Andrea Loreggia
European University Institute
Firenze, Italy
andrea.loreggia@gmail.com

Maria Silvia Pini
Department of Information Engineering
University of Padova, Padova, Italy
pini@dei.unipd.it

Francesca Rossi
IBM Research
Yorktown Heights, NY, USA
francesca.rossi2@ibm.com

ABSTRACT

We propose an innovative ensemble technique that uses voting rules over a set of randomly-generated classifiers. Given a new input sample, we interpret the output of each classifier as a ranking over the set of possible classes. We then aggregate these output rankings using a voting rule, which treats them as preferences over the classes. We show that our approach obtains good results compared to the state-of-the-art, both providing a theoretical analysis and an empirical evaluation of the approach on several datasets.

KEYWORDS

Multi-agent learning; Machine learning; Social choice theory

ACM Reference Format:
Cristina Cornelio, Michele Donini, Andrea Loreggia, Maria Silvia Pini, and Francesca Rossi. 2020. Voting with Random Classifiers (VORACE). In *Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), Auckland, New Zealand, May 9–13, 2020,* IFAAMAS, 3 pages.

1 INTRODUCTION

It is not easy to identify the best classifier for a certain complex task. Different classifiers may be able to exploit better the features of different regions of the domain at hand, and consequently their accuracy might be better only in that region [3, 11, 14]. Moreover, fine-tuning the classifier's hyper-parameters is a time-consuming task, which also requires a deep knowledge of the domain and a good expertise in tuning various kinds of classifiers. An alternative approach, useful to improve the performance of a classifier, consists of combining several different classifiers (that is, an *ensemble* of them) and taking the class proposed by their combination [1, 2, 8, 9, 12, 13, 15, 17, 18, 20]. In this paper we propose a new ensemble classifier method, called VORACE, which aggregates randomly generated classifiers using voting rules in order to provide an accurate prediction for a supervised classification task. We interpret each classifier as a voter, whose vote is its prediction over the classes, and a voting rule aggregates such votes to identify the "winning" class. This use of voting rules is within the framework of maximum likelihood estimators, where each vote (that is, a classifier's rank of all classes) is interpreted as a noisy perturbation of the correct ranking (that is not available), so a voting rule is a way to estimate this correct ranking [5, 6, 19]. We theoretically and experimentally show that the usage of generic classifiers in an ensemble environment can give results that are comparable with other state-of-the-art ensemble methods. To study the accuracy of our method, we performed three kinds of experiments over 25 datasets: varying the number of individual classifiers in the profile and averaging the performance over all datasets; fixing the number of individual classifiers and analyzing the performance on each dataset and considering the introduction of more complex classifiers as base classifiers for VORACE. We show that this approach generates ensemble classifiers that perform similarly to, or even better than, existing ensemble methods. This is especially true when VORACE employs Plurality or Copeland as voting rules. In particular, Plurality has also the added advantage to require very little information from the individual classifiers and to be tractable. Moreover, we provide a closed formula to compute the probability that our ensemble method chooses the correct class in the case of Plurality, assuming that all the classifiers are independent and have the same accuracy and we define the probability of choosing the right class when the classifiers have different accuracy and they are not independent. Besides the classical properties that the voting theory community has considered (like anonymity, monotonicity, IIA, etc.), there may be also other properties not yet considered, such as various forms of fairness, whose study is facilitated by the use of voting rules.

2 VORACE

VORACE generates a profile of n classifiers (where n is an input parameter) by randomly choosing the type of each classifier among a set of predefined ones. For instance, the classifier type can be drawn between a decision tree or a neural network. For each classifier, some of its hyper-parameters values are chosen at random, where the choice of which hyper-parameters and which values are

* A longer version of the paper can be found on ArXiv [7].
† This work was mainly conducted prior joining Amazon.

Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), B. An, N. Yorke-Smith, A. El Fallah Seghrouchni, G. Sukthankar (eds.), May 9–13, 2020, Auckland, New Zealand.

randomly chosen depends on the type of the classifier. When all classifiers are generated, they are trained using the same set of training samples. For each classifier, the output is a vector with as many elements as the classes, where the i-th element represents the probability that the classifier assigns the input sample to the i-th class. Output values are ordered from the highest to the smallest one, and the output of each classifier is interpreted as a ranking over the classes, where the class with higher value is the first in the ranking, then we have the class that has the second highest value in the output of the classifier, and so on. These rankings are then aggregated using a voting rule. The winner of the election is the class with higher score. This corresponds to the prediction of VORACE. In cases of ties, the winner is elected using a tie-breaking rule, which chooses the candidate that is most preferred by the classifier with the highest validation accuracy in the profile.

3 EXPERIMENTAL RESULTS

We considered 23 datasets from the UCI repository [16]. To generate the individual classifiers, we use three classification algorithms: Decision Trees (DT), Neural Networks (NN), and Support Vector Machines (SVM). To study the accuracy of our method, we performed three kinds of experiments:

1) Varying the number of individual classifiers in the profile and averaging the performance over all datasets. The first experiment shows that increasing the number of classifiers in the profile leads to an improvement of the performance.

2) Fixing the number of individual classifiers and analyzing the performance on each dataset. The second experiment shows that it is possible to reach performances very close or better to the state-of-the-art using a very simple method as VORACE is.

Aggregation	Multiclass	Binary	All
Borda	0.9382	**0.8726**	0.9068
Plurality	**0.9469**	**0.8726**	**0.9114**
Copeland	0.9451	**0.8726**	0.9104
HalfAppr	0.9056	**0.8726**	0.8898
Sum	0.9456	0.8550	0.9023
RF	0.8492	0.8661	0.8573
XGBoost	0.9264	0.8605	0.8949

Table 1: Average F1-score on binary and multiclass datasets. For binary datasets, all the voting rules collapse to Majority.

In Table 1 we can see the results for this experiment averaged on: the multiclass datasets; the binary datasets and on all the considered datasets. Here VORACE is used in combination with 4 different voting rules (Borda, Plurality, Copeland and Half Approval) and compared to 3 state-of-the-art methods: Sum method[1], Random Forest [10] and XGBoost [4].

3) Considering the introduction of more complex classifiers as base classifiers for VORACE. In the third experiment instead we study a more complex version of VORACE (allowing more complex classifiers as base classifiers). The results show that introducing this additional level of complexity does not provide any significant advantage in terms of performance.

[1]It computes $x_j^{\text{Sum}} = \sum_i^n x_{j,i}$ for each individual classifier i and for each class j, where $x_{j,i}$ is the probability that the sample belongs to class j predicted by classifier i.

4 THEORETICAL ANALYSIS

We are interested in computing the probability that VORACE chooses the correct class, using Plurality voting rule.

Independent classifiers with same accuracy. Initially, we consider a scenario with m classes (the candidates) and a profile of n independent classifiers (the voters), where each classifier has the same probability p of classifying a given instance correctly.

Theorem 4.1. *The probability of electing the correct class c^*, among m classes, with a profile of n classifiers, each one with accuracy $p \in [0, 1]$, using Plurality is given by:*

$$\mathcal{T}(p) = \frac{1}{K}(1-p)^n \sum_{i=\lceil \frac{n}{m} \rceil}^{n} \varphi_i (n-i)! \binom{n}{i} \left(\frac{p}{1-p}\right)^i \tag{1}$$

where φ_i is the coefficient of the monomial x^{n-i} in the expansion of the generating function $\mathcal{G}_i^m(x)$ and K is a normalization constant:

$$\mathcal{G}_i^m(x) = \left(\sum_{j=0}^{i-1} \frac{x^j}{j!}\right)^{m-1} \qquad K = \sum_{j=0}^{n} \binom{n}{j} p^j (m-1)^{n-j} (1-p)^{n-j}$$

Independent classifiers with different accuracy. Considering the same accuracy p for all classifiers is not realistic, thus we consider the general case where each classifier in the profile can have a different accuracy p_i, while still considering them independent. The probability of choosing the correct class c^* is:

$$\frac{1}{K} \sum_{(S_1,\dots,S_m) \in \Omega_{c^*}} \Big[\prod_{i \in \overline{S^*}} (1-p_i) \cdot \prod_{i \in S^*} p_i \Big]$$

where K is the normalization function, S is the set of all classifiers $S = \{1, 2, \dots, n\}$; S_i is the set of classifiers that elect candidate c_i; S^* is the set of classifiers that elect c^*; $\overline{S^*}$ is the complement of S^* in S ($\overline{S^*} = S \setminus S^*$); and Ω_{c^*} is the set of all possible partitions of S in which c^* is chosen:

$$\Omega_{c^*} = \{(S_1, \dots, S_{m-1}) | \text{ partitions of } \overline{S^*} \text{ s.t. } |S_i| < |S^*| \; \forall i : c_i \neq c^*\}.$$

Dependent classifiers. We now relax the independence assumption between classifiers by taking into account the presence of areas of the domain that are correctly classified by at least half of the classifiers simultaneously. We denote by ϱ the ratio of the examples that are in the *easy-to-classify* part of the domain. ϱ is bounded by the probability of the correct classification of an example by at least half of the classifiers (which are correctly classified by the ensemble). Removing the *easy-to-classify* examples from the training dataset, we obtain the accuracy $\widetilde{p} = ((p - \varrho)/(1 - \varrho)) < p$ for the other examples, leading to a generalization of Theorem 4.1:

Theorem 4.2. *The probability of choosing the correct class c^* in a profile of n classifiers with accuracy $p \in [0, 1[$, m classes, overlapping value ϱ and using Plurality to compute the winner, is larger than:*

$$(1-\varrho)\mathcal{T}(\widetilde{p}) + \varrho\ . \tag{2}$$

5 CONCLUSIONS

We have proposed the use of voting rules in the context of ensemble classifiers, in line with the MLE approach to voting. Via a theoretical and experimental analysis, we have shown that this approach generates ensemble classifiers that perform similarly to, or even better than, existing ensemble methods.

REFERENCES

[1] Tayyab Ateeq, Muhammad Nadeem Majeed, Syed Muhammad Anwar, Muazzam Maqsood, Zahoor-Ur Rehman, Jong Weon Lee, Khan Muhammad, Shuihua Wang, Sung Wook Baik, and Irfan Mehmood. 2018. Ensemble-classifiers-assisted detection of cerebral microbleeds in brain MRI. *Computers & Electrical Engineering* 69 (2018), 768–781. https://doi.org/10.1016/j.compeleceng.2018.02.021

[2] Mohsen Azadbakht, Clive S. Fraser, and Kourosh Khoshelham. 2018. Synergy of sampling techniques and ensemble classifiers for classification of urban environments using full-waveform LiDAR data. *Int. J. Applied Earth Observation and Geoinformation* 73 (2018), 277–291. https://doi.org/10.1016/j.jag.2018.06.009

[3] Eric Bauer and Ron Kohavi. 1999. An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants. *Machine Learning* 36, 1-2 (1999), 105–139. https://doi.org/10.1023/A:1007515423169

[4] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proc. of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. ACM, 785–794.

[5] Vincent Conitzer, Matthew Rognlie, and Lirong Xia. 2009. Preference Functions that Score Rankings and Maximum Likelihood Estimation. In *IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, July 11-17, 2009*. 109–115.

[6] Vincent Conitzer and Tuomas Sandholm. 2005. Common Voting Rules As Maximum Likelihood Estimators. In *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence (UAI'05)*. AUAI Press, Arlington, Virginia, United States, 145–152. http://dl.acm.org/citation.cfm?id=3020336.3020354

[7] Cristina Cornelio, Michele Donini, Andrea Loreggia, Maria Silvia Pini, and Francesca Rossi. 2019. Voting with Random Classifiers (VORACE). In *arXiv:1909.08996*.

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

[9] Isha Gandhi and Mrinal Pandey. 2015. Hybrid Ensemble of classifiers using voting. In *Green Computing and Internet of Things (ICGCIoT), 2015 International Conference on*. IEEE, 399–404.

[10] Tin Kam Ho. 1995. Random decision forests. In *Document analysis and recognition*, Vol. 1. IEEE, 278–282.

[11] Taghi M. Khoshgoftaar, Jason Van Hulse, and Amri Napolitano. 2011. Comparing Boosting and Bagging Techniques With Noisy and Imbalanced Data. *IEEE Trans. Systems, Man, and Cybernetics, Part A* 41, 3 (2011), 552–568. https://doi.org/10.1109/TSMCA.2010.2084081

[12] J. Kittler, M. Hatef, and R. P. W. Duin. 1996. Combining Classifiers. In *Proceedings of the Sixth International Conference on Pattern Recognition*. IEEE Computer Society Press, Silver Spring, MD, 897–901.

[13] Sotiris B. Kotsiantis, Ioannis D. Zaharakis, and Panayiotis E. Pintelas. 2006. Machine learning: a review of classification and combining techniques. *Artif. Intell. Rev.* 26, 3 (2006), 159–190. https://doi.org/10.1007/s10462-007-9052-3

[14] Prem Melville, Nishit Shah, Lilyana Mihalkova, and Raymond J. Mooney. 2004. Experiments on Ensembles with Missing and Noisy Data. In *Multiple Classifier Systems, 5th International Workshop, MCS 2004, Cagliari, Italy, June 9-11, 2004*. 293–302. https://doi.org/10.1007/978-3-540-25966-4_29

[15] Antonino Feitosa Neto and Anne M. P. Canuto. 2018. An exploratory study of mono and multi-objective metaheuristics to ensemble of classifiers. *Appl. Intell.* 48, 2 (2018), 416–431. https://doi.org/10.1007/s10489-017-0982-4

[16] C.L. Blake D.J. Newman and C.J. Merz. 1998. UCI Repository of machine learning databases. (1998). http://www.ics.uci.edu/$\sim$mlearn/MLRepository.html

[17] Lior Rokach. 2010. Ensemble-based classifiers. *Artificial Intelligence Review* 33, 1-2 (2010), 1–39.

[18] Emran Saleh, Jerzy Blaszczynski, Antonio Moreno, Aïda Valls, Pedro Romero-Aroca, Sofia de la Riva-Fernandez, and Roman Slowinski. 2018. Learning ensemble classifiers for diabetic retinopathy assessment. *Artificial Intelligence in Medicine* 85 (2018), 50–63. https://doi.org/10.1016/j.artmed.2017.09.006

[19] Roman Seidl. 2018. Handbook of Computational Social Choice *by Brandt Felix, Vincent Conitzer, Ulle Endriss, Jerome Lang, Ariel Procaccia. J. Artificial Societies and Social Simulation* 21, 2 (2018). http://jasss.soc.surrey.ac.uk/21/2/reviews/4.html

[20] Xiaofeng Sun, Xiangguo Lin, Shuhan Shen, and Zhanyi Hu. 2017. High-Resolution Remote Sensing Data Classification over Urban Areas Using Random Forest Ensemble and Fully Connected Conditional Random Field. *ISPRS Int. J. Geo-Information* 6, 8 (2017), 245. https://doi.org/10.3390/ijgi6080245