Voting with Random Classifiers (VORACE): Theoretical and Experimental Analysis

JAAMAS Track

Cristina Cornelio
Samsung AI
Cambridge, United Kingdom
c.cornelio@samsung.com

Michele Donini*
Amazon
Berlin, Germany
donini@amazon.com

Andrea Loreggia
University of Brescia
Brescia, Italy
andrea.loreggia@gmail.com

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

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

ABSTRACT

Ensemble methods are built by training many different models and aggregating their outputs to output the prediction of the whole system. In this work, we study the behavior of an ensemble method where voting rules are used to aggregate the output of a set of randomly-generated classifiers. We provide both a theoretical and an empirical analysis of this method, showing that it performs comparably with other state-of-the-art ensemble methods, while not requiring any domain expertise to fine-tune the individual classifiers.

KEYWORDS

Multi-agent Learning; Machine Learning; Social Choice Theory; Ensemble Methods

ACM Reference Format:
Cristina Cornelio, Michele Donini, Andrea Loreggia, Maria Silvia Pini, and Francesca Rossi. 2022. Voting with Random Classifiers (VORACE): Theoretical and Experimental Analysis: JAAMAS Track. In *Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), Online, May 9–13, 2022*, IFAAMAS, 3 pages.

1 INTRODUCTION

In machine learning, an ensemble classifier consists of a set of classifiers whose outputs are aggregated to form the prediction of the system [9, 12]. This approach is justified by the observation that it is not easy to identify the best classifier for a certain complex task and that different classifiers may learn differently on different regions of the domain [1, 8, 10]. In this work, we propose an ensemble classifier (called VORACE) that considers each classifier as a voter in an election, expressing its preference on a set of possible alternatives (that is, the classes). Such preferences are then aggregated using a voting rule to compute the output of the ensemble classifier. A voting rule [13] is a function that chooses one out of a set of candidates, starting from a set of rankings over the candidates.

This use of voting rules is within the framework of maximum likelihood estimators, where each vote 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 [3, 4]. We 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. We also provide a closed formula to compute the probability that our ensemble method chooses the correct class when the voting rule used is Plurality, assuming that all the classifiers are independent and have the same accuracy. We also define the probability of choosing the right class when the classifiers have different accuracy and they are not independent. The proposed work has been published in the Journal of Autonomous Agents and Multi-Agent Systems [6], that is a revised and extended version of [5, 7]. All the code is available at https://github.com/aloreggia/vorace/.

2 VORACE

VORACE (VOting with RAndom ClassifiErs) is an ensemble method that uses a profile of n random classifiers, where n is an input parameter. The type of each classifier is chosen at random from a set of predefined ones, (some of) whose hyper-parameters are chosen at random. Classifiers in the ensemble are trained using the same set of training samples. The output of each classifier is an m-dimensional vector, with m the number of classes, representing the probability distribution that the input sample belongs to a class. This can be interpreted as a ranking over the classes, where the class with the highest probability is the first in the ranking. VORACE aggregates the rankings from the random classifiers by using a voting rule. In case of ties VORACE chooses the candidate that is most preferred by the classifier with the highest validation accuracy in the profile. This winner is the output of the ensemble classifier.

3 EXPERIMENTAL RESULTS

We considered 23 datasets from the UCI repository [11]. Individual classifiers are generated choosing among three classification algorithms: Decision Trees (DT), Neural Networks (NN), and Support Vector Machines (SVM). For each dataset, we train and test the ensemble method with a 10-fold cross validation process. Additionally, for each dataset, experiments are performed 10 times, leading to a

*This work was mainly conducted prior joining Amazon.

Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), P. Faliszewski, V. Mascardi, C. Pelachaud, M.E. Taylor (eds.), May 9–13, 2022, Online.

Table 1: Average F1-scores (and standard deviation), varying the number of voters, averaged over all datasets.

	Avg Profile	Borda	Plurality	Copeland	Kemeny	Sum	Best C.
Avg	0.8626 (0.0981)	0.8983 (0.0987)	**0.9006 (0.0998)**	0.9002 (0.0998)	0.9002 (0.1001)	0.8964 (0.1070)	0.8673 (0.1192)

Table 2: Performance on multiclass and binary datasets: Average F1-scores (and standard deviation). Best performance in bold. On binary datasets, all the voting rules behave as majority voting rule.

	Borda	Plurality	Copeland	Kemeny	Sum	RF	XGBoost
Avg Multi	0.9365 (0.0421)	0.9413 (0.0388)	0.9396 (0.0380)	0.9402 (0.0382)	**0.9416 (0.0399)**	0.8720 (0.0410)	0.9177 (0.0409)
Avg Binary	-	**0.8724 (0.0493)**	-	-	0.8574 (0.0658)	0.8666 (0.0409)	0.8636 (0.0493)

total of 100 runs for each method over each dataset. This is done to ensure greater stability. The voting rules considered in the experiments are Plurality, Borda, Copeland and Kemeny [13]. We compare the performance of VORACE to 1) the average performance of the individual classifiers in the profile, 2) the performance of the best classifier in the profile, 3) the performance of two state-of-the-art methods (Random Forest and XGBoost), and 4) the performance of the *Sum* method (also called *weighted averaging*). The *Sum* method 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. The winner is the one with the maximum value in the sum vector: $\arg\max x_j^{\text{Sum}}$. 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, 2) fixing the number of individual classifiers and analyzing the performance on each dataset and 3) considering the introduction of more complex classifiers as base classifiers for VORACE. Since the first experiment shows that the best accuracy of the ensemble occurs when $n = 50$, we use only this size for the second and third experiments. Table 1 and Table 2 report the aggregated results of the experiments. It is easy to see that using voting rules with random classifiers gives results that are comparable to using state of the art methods like RF and XGBoost, while not requiring domain expertise or time consuming parameter adjustment.

4 THEORETICAL ANALYSIS

Independent classifiers with same accuracy. 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 defined as the coefficient of the monomial x^{n-i} in the expansion of the following generating function: $\mathcal{G}_i^m(x) = \left(\sum_{j=0}^{i-1} \frac{x^j}{j!}\right)^{m-1}$ and K is a normalization constant defined as: $K = \sum_{j=0}^{n} \binom{n}{j} p^j (m-1)^{n-j}(1-p)^{n-j}$.

Independent classifiers with different accuracy. Considering the same accuracy p for all classifiers is not realistic. Thus we also study the general case where each classifier in the profile can have a different accuracy p_i, while still assuming they are independent. In this scenario, the probability of choosing the correct class c^* is:

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

where K is the normalization function, S is the set of all classifiers $S = \{1, 2, \ldots, n\}$; 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:

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

Comparison with the Condorcet Jury Theorem. We prove that, for $m = 2$, Formula 1 in Theorem 4.1 enforces the results stated in the Condorcet Jury Theorem [2]. However, since the assumptions in Theorem 4.1 do not always hold in practice, we prove the following broader statement:

Theorem 4.2. *The probability of electing the correct class c^*, among 2 classes, with a profile of an infinite number of classifiers, each one with accuracy $p \in [0,1]$, using Plurality, is given by:*

$$\lim_{n\to\infty} \mathcal{T}(p) = \begin{cases} 0 & p < 0.5 \\ 0.5 & p = 0.5 \\ 1 & p > 0.5 \end{cases} \tag{2}$$

Dependent classifiers. We also 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 4.3. *The probability of choosing the correct class c^* in a profile of n classifiers with accuracy $p \in [0,1[$, m classes and with an overlapping value ϱ, using Plurality to compute the winner, is larger than:*

$$(1-\varrho)\mathcal{T}(\widetilde{p}) + \varrho \, . \tag{3}$$

REFERENCES

[1] 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

[2] J.-A.-N. Condorcet and Marquis de Caritat. 1785. Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix. *Fac-simile reprint of original published in Paris, 1972, by the Imprimerie Royale* (1785).

[3] 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.

[4] 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* (Edinburgh, Scotland) *(UAI'05)*. AUAI Press, Arlington, Virginia, United States, 145–152. http://dl.acm.org/citation.cfm?id=3020336.3020354

[5] Cristina Cornelio, Michele Donini, Andrea Loreggia, Maria Silvia Pini, and Francesca Rossi. 2020. Voting with Random Classifiers (VORACE). In *Proceedings of the 19th International Conference On Autonomous Agents and Multi-Agent Systems (AAMAS)*. 1822–1824.

[6] Cristina Cornelio, Michele Donini, Andrea Loreggia, Maria Silvia Pini, and Francesca Rossi. 2021. Voting with random classifiers (VORACE): theoretical and experimental analysis. *Autonomous Agents and Multi-Agent Systems* 35, 2 (2021), 1–31.

[7] Michele Donini, Andrea Loreggia, Maria Silvia Pini, and Francesca Rossi. 2018. Voting with Random Neural Networks: a Democratic Ensemble Classifier.. In *RiCeRcA@AI*IA*.

[8] 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

[9] 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.

[10] 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

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

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

[13] 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