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

JAAMAS Track

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

Michele Donini
Amazon
Berlin, Germany
donini@amazon.com

Andrea Loreggia
University of Brescia
Brescia, Italy
andreloreggia@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:

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 work was mainly conducted prior joining Amazon.

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
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 \( x_{ij}^{\text{Sum}} = \sum_n x_{ij} \) for each individual classifier \( j \) and for each class \( i \), where \( x_{ij} \) is the probability that the sample belongs to class \( i \) predicted by classifier \( j \). The winner is the one with the maximum value in the sum vector: arg max \( x_{ij}^{\text{Sum}} \). 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:

\[
T(p) = \frac{1}{K}(1-p)^n \sum_{i=1}^{n} \varphi_i(n-i) \left( \binom{n}{i} \left( \frac{p}{1-p} \right)^i \right)
\]

where \( \varphi_i \) is defined as the coefficient of the monomial \( x^{n-i} \) in the expansion of the following generating function: \( G_{\Omega}(x) = \left( \sum_{i=1}^{n} \frac{x_i}{p} \right)^{m-1} \) and \( K \) is a normalization constant defined as: \( K = \sum_{j=0}^{n} \left( \binom{n}{j} \right)^p (m-1)^{n-j} (1-p)^{n-j} \).

**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:

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

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 \( c_i \); \( S^c \) is the set of classifiers that elect \( c^* \); \( S^c \) is the complement of \( S^c \) in \( S \); and \( \Omega_{c^*} \) is the set of all possible partitions of \( S \) in which \( c^* \) is chosen:

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

**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:

\[
\lim_{n \to \infty} T(p) = \begin{cases} 
0 & p < 0.5 \\
0.5 & p = 0.5 \\
1 & p > 0.5 
\end{cases}
\]

**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 \( \varrho \) the ratio of the examples that are in the easy-to-classify part of the domain. \( \varrho \) 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 \( \overline{p} = (p - \varrho)/(1 - \varrho) \) < \( p \) 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 \( \varrho \), using Plurality to compute the winner, is larger than:

\[
(1-\varrho)T(\overline{p}) + \varrho.
\]
REFERENCES


