International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 10 | Oct 2021 www.irjet.net p-ISSN: 2395-0072

Introduction to Ensemble Methods for Machine Learning Applications

Maheswari Kannapureddy[1]

[1]*School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, India*

Abstract - *Ensemble methods are successful in machine learning applications. Applications of ensemble methods have been used in a broad range of industries, such as air traffic controllers which uses ensembles to minimize airplanes arrival delay time. This paper describes mainly about three ensemble algorithms. They are stacking, bagging and boosting. These three ensemble methods which are also known as meta-algorithms approaches which mainly focuses on combining various machine learning algorithms into a single model through which predictions can be done to decrease the variance, which is also called bagging and boosting to decrease bias and improving predictive force using stacking.*

Key Words: **Ensemble methods, stacking, bagging, boosting, Machine learning**

1.INTRODUCTION

Numerous weather forecast agencies implement ensemble learning to improve weather forecasting accuracy. Why Ensemble methods? In machine learning, statistical problem arises as we often have only limited number of datasets in practice. Therefore, we can find various hypotheses which fit reasonably better and we cannot predict which one of them has the best generalization performance. Eventually, this makes difficult to select one among them. Therefore, the use of ensemble methods which can help to avoid the issue by taking the average over several models available to get a better approximation of the unknown true hypothesis. The second reason for choosing ensemble methods is that it is computational. For example an ensemble constructed by starting the local search from many different available points may provide a good approximation to the true unknown function. Depending on the application and the approach, the ensemble methods are broken down to various stages.

1.1 Literature Survey

An algorithmic extension to the stacked regression can be made which prunes size of a homogenous ensemble set by considering the diversity of contents or members of the set and accuracy.[1] By using this stacked ensemble method the ensemble set as accurate as that of non pruned version by taking an average of the datasets that are tested in non prune version, which in turn provides benefits in ensemble set such as its application efficiency and reduced complexity of the set.[2] Using stacking, ensemble size can be reduced using the base models generated using two different learning algorithms. Stacking performs a cross validation history of its base models to form the meta-data it would require little algorithmic overhead to determine accuracy and diversity of each of the base model during the training phase.[3] These measures are then used as a means for pruning the base members which are considered as not sufficiently accurate or diverse. As a result, the ensemble set could be reduced in size which leads to the increase in efficiency of stacking.[4] For constructing a heterogeneous classifiers with stacking and to show that they perform the best when compared to selecting the best classifier from the ensemble by cross validation. Stacking with probability distributions and multi response linear regression performs the best. When compared to the stacking approaches, several other extensions such as using an extended set of meta level features and the other is using multi response model trees to learn at the meta level proves to be better than existing stacking approaches.[5] The process of stacked generalization is successfully used in many methods such as protein prediction, bankruptcy prediction and many others.[6] In the previous methods, some of the issues are solved for third party testing and all.[7] Here in this paper, using stacked generalization, it is more focused to propose more applicable and accurate defect predicting model for testing.[8] When people make complicated decisions, generally opinions of the server are taken into account instead of depending on their own judgment or that of an advisor, thereby combining various prediction models is the most recommended approach. The number of detected defects depends on various factors other than only code and its complexity.[9] Defect number prediction is essential in order to make a key decision on the time to stop testing . [10]For more accurate prediction an ensemble prediction model based on stacked generalization can be used, and the number of defects can be detected by thee third party black box testing.[11] Stacked generalization is a way of combining various models. It is mainly applied to models which are built by different learning algorithms and obtain the final prediction results according to the various prediction models.[12] Unlike bagging and boosting, stacked generalization is used to combine models of the different type. [13]In stacked generalization, multiple models are divided into level 0 and level 1 models. Intially level 0 models are trained and predicted the original data with multiple level 0 prediction models separately. [14] Level 1 models are trained and predicted, using the output results that are obtained while predicting level 0 models. After the base classifiers are obtained, the next stage would be process of generating ensembles is basically an application of proper combination scheme.[15]

1.2 Application of Ensemble methods

Depending on the application and the approach, the ensemble methods are broken down to various stages. In the application of Ensemble methods, the first method would be ensemble generation. It mainly focuses on obtaining a set of calibrated models that have their analyzed outcome and individual prediction. In this stage base models are obtained. In the next stage called ensemble pruning ,it describes the process of choosing subset of models from thee available ones with better accuracy .The last step in any ensemble method is ensemble integration. It describes how the remaining calibrated models are combined into a single composite set. Ensemble integration methods differ in approaches and classifications. An ensemble method is said to be successful when it has accurate predictors and commits error in different parts of the input space.

1.3 Steps involved in Algorithm

Every algorithm comprises of two steps

1) Producing a distribution of sample machine learning models on subsets of the available original data

2) Integrating the distribution into one aggregate or composite model

Both bagging and boosting use a single learning algorithm for all the steps, but they use different methods on handling the training samples. Both thee bagging and boosting are ensemble methods that combines decision from multiple models in order to improve accuracy.

2. Bagging Algorithm

2.1 Introduction to Bagging Algorithm

Basically stands for Bootstrap Aggregating is a way to decrease the variance of the prediction by producing additional or extra data for training from the present original data set using repetitions with combinations in order to produce multisets of the same size of cardinality as that of the present original data. By increasing the cardinality or size of the training set it is not possible to increase the predictive force, but there is possibility to decrease the variance, narrowly tuning thee prediction to that of expected outcome.

- Re-samples training data to get M number of subsets (bootstrapping)
- Trains M classifiers (same algorithm) based on the M datasets- different samples
- Final classifier combines M outputs by voting
- Samples weight equally
- Classifiers weight equally
- Decreases errors by decreasing variance

When does bagging work: Bagging especially works in two cases. They are

1) Under-fitting

2) Over-fitting

Under-fitting: A model is said to be under-fitting when t is not accurate or when the model has small variance (smaller influence of examples in the training set)

Over-fitting: Small bias (Models that are flexible enough to fit well to the training data) Large variance (dependence on the training set is very large)

2.2 How Bagging Algorithm works

Bagging algorithm:

Step 1: Information given Training set of some particular number of examples say N A class containing learning models, say for example neural networks, decision tree

Step 2: Method to be followed: Multiple models with different samples are to be trained and the average of their predictions is calculated. Predicting or testing the average of the results obtained in the k models and

Step 3: Main goal: Multiple copies are used in order to increase the accuracy of one model Misclassification errors on various data splits average is taken which in turn gives a better estimate of the predictive ability of a learning method

Step 4: Training In every iteration for example n=1,2,3….N N samples from the training set are replaced with randomly samples A base model is chosen and is trained **Step 5**: Test For each and every test example, the process is started with trained base models Results are predicted by combining the results of all the N trained models

2.3 Architecture

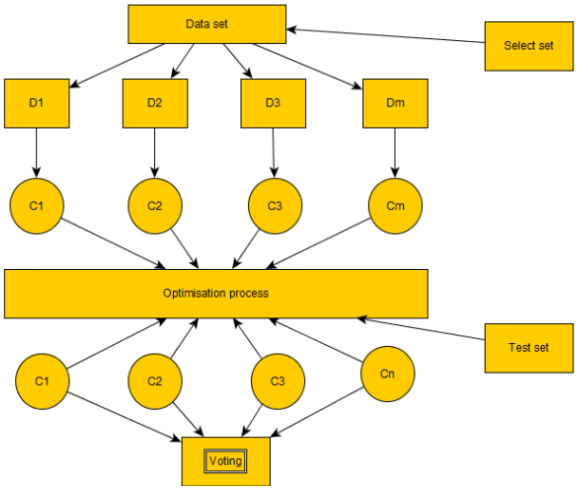

Fig -1: Architecture of Bagging algorithm

3. Boosting Algorithm

3.1 Introduction to Bagging Algorithm

There are two steps involved in boosting. In the first step the method uses subsets of the original data that is provided which in turn produces a series of models which are averagely performing. In the second step, a particular cost

function is used for combining together which leads to boosting up of performance. The subset creation is not random in the case of classical boosting unlike bagging and it mainly depends upon the performance of the existing or previous models. Every new subset that is created contains the elements that were misclassified in the existing or previous models.

Boosting algorithm mainly focuses on:

Begin with equal weight for all the samples in the first round

•In the upcoming following M-1 rounds the samples which were misclassified in the existing or previous models that is in the last round, weights are increased

•By using the weighted voting, the final classifier which is obtained in the last round combines various multiple classifiers obtained in the previous rounds, and assign larger weights to the classifiers with less misclassification

•Step wise re-weights samples, weights for each round based on the results obtained from the previous rounds, re-weight samples (boosting) instead of re sampling (bagging)

3.2 How Boosting Algorithm works

Step 1: Given information A set of N examples are taken as a training set A base training model is chosen be it either neural network, decision tree

Step 2: Stage of training A sequence of N base models are trained on N different sampling distributions which are defined upon the training set A sample distribution M(n), for building the model n is constructed by modifying the sampling distribution M(n-1) from the (n-1)th step. The classification of example which are done incorrectly in the previous step receive comparatively higher weights in the new data (increase in weights is the attempt to cover misclassified examples

Step 3: Classification stage: According to the weighted majority of the classifiers thee examples are classified accordingly.

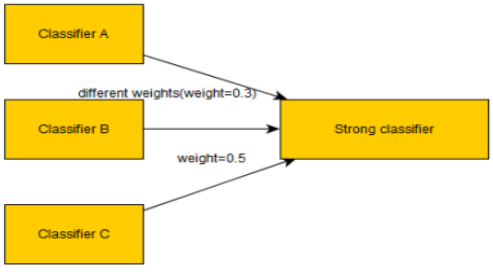

Fig -2: Architecture of Boosting algorithm

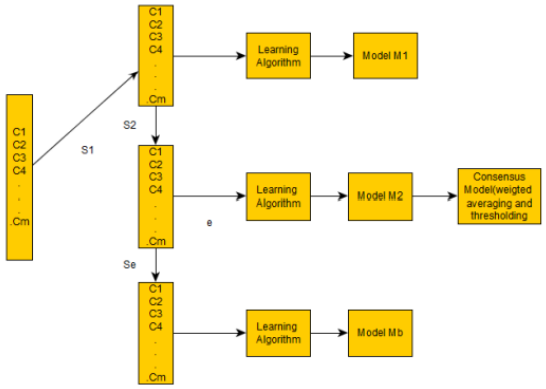

Fig -3: Working of boosting algorithm

4. Stacking Algorithm

4.1 Introduction to Stacking Algorithm

Stacking is basically increasing the predictive force of the classifier. Stacking is similar to boosting; we also apply several models to the original data. The difference in stacking is that however that we don't just have an empirical formula for the weight function, rather a meta-level is introduced and another model or approach is used to estimate the input together with outputs of every model present to estimate the weights. In other words, to determine what models perform well and to determine how badly the input is given for a particular model. The basic underlying idea of stacked generalization or stacking is to learn whether training data has been properly learned. Stacking is concerned with multiple classifiers generated by different learning algorithms. The stacking process is primarily broken down into two processes, the first process involves generation of particular set of base-level classifiers and the second process involves training of a meta-classifier to combine all the outputs of the base-level classifiers obtained in the first process.

4.2 How Stacking Algorithm works

Step 1:Input or the information given Data sets , for example D={(x1,y1),(x2,y2),....(xm,ym)} First level learning algorithms L1....L(t) Second level learning algorithm L

Step 2: Process Train the first level individual learner by applying the first level learning algorithm L(t) to the original dataset D Generate a new dataset Use individual learner h(t) to classify the training examples x(i) Train the second level learner by applying the second-level learning algorithm L to the new dataset d

Step 3: Output H(x)=h'(h1(x),....ht(x))

4.3 Architecture of stacking algorithm

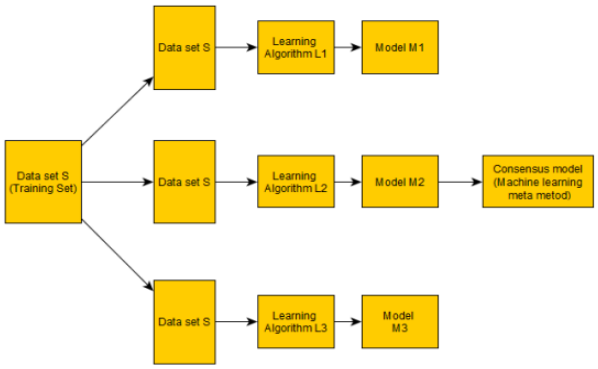

Fig -4: Architecture of stacking algorithm

5. CONCLUSIONS

Numerous weather forecast agencies implement ensemble learning to improve weather forecasting accuracy. Why Ensemble methods? In machine learning, statistical problem arises as we often have only limited number of datasets in practice. Therefore, we can find various hypotheses which fit reasonably better and we cannot predict which one of them has the best generalization performance. Eventually, this makes difficult to select one among them. In general, the field of data mining is improving at a rapid pace and the performance of these algorithms would slowly evolve.

REFERENCES

[1] Gupta, A., & Thakkar, A. R. (2014). Optimization of stacking ensemble Configuration based on various metahueristic algorithms. 2014 IEEE International Advance Computing Conference (IACC). doi:10.1109/iadcc.2014.6779365

[2] Jurek, A., Bi, Y., Wu, S., & Nugent, C. D. (2014). Clustering-Based Ensembles as an Alternative to Stacking. IEEE Transactions on Knowledge and Data Engineering,26(9), 2120-2137. doi:10.1109/tkde.2013.49

[3] Kadkhodaei, H., & Moghadam, A. M. (2016). An entropy based approach to find the best combination of the base classifiers in ensemble classifiers based on stack generalization. 2016 4th International Conference on Control, Instrumentation, and Automation (ICCIA). doi:10.1109/icciautom.2016.7483200

[4] Korzh, O., Cook, G., Andersen, T., & Serra, E. (2017). Stacking approach for CNN transfer learning ensemble for remote sensing imagery. 2017 Intelligent Systems Conference(IntelliSys).doi:10.1109/intellisys.2017.8324356

[5] Li, N., Li, Z., Nie, Y., Sun, X., & Li, X. (2011). Predicting software black-box defects using stacked generalization. 2011 Sixth International Conference on Digital Information Management. doi:10.1109/icdim.2011.6093330

[6] Liao, S., Zhang, H., Shu, G., & Li, J. (2017). Adaptive Resource Prediction in the Cloud Using Linear Stacking Model. 2017 Fifth International Conference on Advanced Cloud and Big Data (CBD). doi:10.1109/cbd.2017.14

[7] Mohammed, M., Mwambi, H., Omolo, B., & Elbashir, M. K. (2018). Using stacking ensemble for microarray-based cancer classification. 2018 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE). doi:10.1109/iccceee.2018.8515872

[8] Nakano, F. K., Mastelini, S. M., Barbon, S., & Cerri, R. (2017). Stacking Methods for Hierarchical Classification. 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA). doi:10.1109/icmla.2017.0-145

[9] Niranjan, A., Akshobhya, K. M., Shenoy, P. D., & Venugopal, K. R. (2018). EKNIS: Ensemble of KNN, Naïve Bayes Kernel and ID3 for Efficient Botnet Classification Using Stacking. 2018 International Conference on Data Science and Engineering (ICDSE). doi:10.1109/icdse.2018.8527791

[10] Pavlyshenko, B. (2018). Using Stacking Approaches for Machine Learning Models. 2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP). doi:10.1109/dsmp.2018.8478522

[11] Rooney, N., Patterson, D., & Nugent, C. (n.d.). Reduced ensemble size stacking [ensemble learning]. 16th IEEE International Conference on Tools with Artificial Intelligence. doi:10.1109/ictai.2004.105

[12] Wang, J., Sun, C., Zhao, Z., & Chen, X. (2017). Feature ensemble learning using stacked denoising autoencoders for induction motor fault diagnosis. 2017 Prognostics and System Health Management Conference (PHM-Harbin). doi:10.1109/phm.2017.8079196

[13] Zhang, Y., Liu, G., Luan, W., Yan, C., & Jiang, C. (2018). An approach to class imbalance problem based on stacking and inverse random under sampling methods. 2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC). doi:10.1109/icnsc.2018.8361344

[14] Zhou, H., Zhang, Y., Yang, L., & Liu, Q. (2018). Short-Term Photovoltaic Power Forecasting Based on Stacking-SVM. 2018 9th International Conference on Information Technology in Medicine and Education (ITME). doi:10.1109/itme.2018.00221

[15] Zirpe, S., & Joglekar, B. (2017). Negation Handling using Stacking Ensemble Method. 2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA)doi:10.1109/iccubea.2017.8463946