ENSEMBLE LEARNING TECHNIQUES AND ITS EFFICIENCY IN MACHINE LEARNING: A SURVEY

Thomas. Rincy. N
Department of Computer Science and Engineering
University Institute of Technology
Rajiv Gandhi Proudyogiki Vishwavidyalaya
Bhopal (M.P), India
rinc_thomas@rediffmail.com

Roopam Gupta
Department of Information Technology
University Institute of Technology
Rajiv Gandhi Proudyogiki Vishwavidyalaya
Bhopal (M.P), India
roopamgupta@rgtu.net.

Abstract- Ensemble learning is an imperative study in the domain of machine learning. Over the previous years, ensemble learning has drawn considerable attention in the field of artificial intelligence, pattern recognition, machine learning, neural network and data mining. Ensemble learning has shown to be efficient and functional in wide area of problem domain and substantial world application. Ensemble learning, it constructs several classifiers or the set of base learners and merge their output so that the overall variance should be reduced. By merging several classifiers or the set of base learners it significantly improves the accuracy in contrast to single classifier or single base learner. In this literature we survey the various ensemble learning techniques that is prevalent in machine learning.

Keywords- Ensemble learning, Boosting, Bagging, Stacking, Mixture of Experts.

I. INTRODUCTION

Ensemble learning begins its success from the year 1990. It was initially evolved to shorten the variance and by that increasing the accuracy of a decision automated making system; ensemble learning is able to strongly address the wide area of machine learning problems such as estimation, confidence, error correction, cumulative learning, missing features etc. Ensemble learning it constructs the several classifiers or number of base learners and combines them to train the dataset. The base learners are called the weak learners and obtained by base learning algorithm on training data. In most of the instances the ensemble learning constructs, the single base learning algorithms called homogenous ensembles, in some instances it constructs by applying different learning algorithms called heterogeneous ensembles [1]. The goodness of the ensemble learning is that they are capable of boosting the weak learners so that overall accuracy of the learning algorithm on training data can be increased. There are many applications that use the ensemble learning techniques. Huang et.al [2] proposes the ensemble learning framework on post face recognition invariant and concludes that ensemble method it performs better than traditional method (single learner). Giacinto et.al [3] introduced ensemble approaches for intrusion detection system and came to conclusion that the ensemble method has excellent detection ability for known attacks. Fig. 1 shows the basic architecture of ensemble learning. The rest of the literature is organised as follows: Section II, it discovers the various reasons behind the development of ensemble learning. Section III, introduces Boosting, a crucial paradigm in ensemble learning. Section IV, it reviews Bagging, an important concept in ensemble learning. Section V, it focuses on Stacking, an imperative learning area in ensemble learning. Section VI, analysis the Mixture of experts, an important study in ensemble learning and we conclude in section VII.

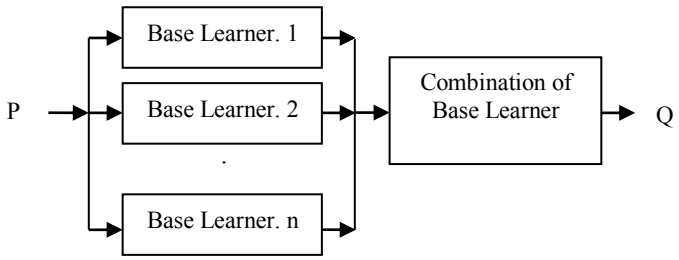

Fig: 1. A Basic architecture of Ensemble Learning.

II. LIMITATIONS OF SINGLE LEARNING ALGORITHM

To have the accurate ensemble of classifiers the mandatory condition is to have more authentic ensemble of classifiers than of its individual classifiers [4]. Accurate classifier is that which is having better error rate than a random guess on the new values. There are three fundamental reasons of failing of the single learning algorithm that leads to the development of ensemble classifier [5].

A. Statistical: Learning algorithm it searches the best hypotheses in the space. Due to insufficient number of training data or the training data is small, compared to hypothesis space the statistical problem arises. This leads to learning algorithm find different hypothesis in space, which gives the same accuracy. The ensemble algorithm helps out this situation by averaging their votes, hence reducing the possibility of choosing the incorrect classifier and thus predicting the accurate accuracy on training data.

B. Computational: The learning algorithm does it job by finding some foam of local search and sometimes they get stuck in the local optima, besides having the enough training data. In fact, the optimal training for decision tree and neural network is NP-Hard [6], so it is computationally difficult to obtain the best hypothesis. By constructing the ensemble learning and then running, local search from various origin points can lead to better resemblance to the accurate unexplored function as compared to single base learner.

III. BOOSTING

Boosting [7], it builds a strong classifier from number of base learners. Boosting works sequentially by training the set of a base learner to combine it for prediction. The boosting algorithm takes the base learning algorithms repeatedly, having the different distributions or weighting of training data on the base learning algorithms. On running the boosting algorithm, the base learning algorithms will generate a weak predicted rule, until various rounds of steps

TABLE I. TAXONOMY OF BOOSTING ALGORITHM

S.No.	Author	Algorithm	Paper	Strategy/ Methods
1	Freund et.al [8]	Adaboost	"A decision-theoretic generalization of on-line learning and an application to boosting".	Applies the weight on the training set. Base learner on hardest examples in order to enhance the learning algorithm.
2	Friedman et.al [9]	LogitBoost	"Additive logistic regression: A statistical view of boosting (with discussions)".	Reduces the loss function by fitting the additive logic regression.
3	Freund et.al [10]	AdaBoost.M1	"A decision-theoretic generalization of on-line learning and an application to boosting".	Implements multi class learners instead of binary class learners.
4	Freund et.al [11]	AdaBoost.M2	"A decision-theoretic generalization of on-line learning and an application to boosting".	Decreases pseudo-loss by applying the one-versus-one strategy.
5	Schapire et.al [12]	AdaBoost.MR	"Improved boosting algorithms using confidence-rated predictions".	Reduces the ranking loss. The highest ranked class is considered.
6	Bradley et.al [13]	FilterBoost	"FilterBoost: Regression and classification on large datasets".	Introduces the log loss function instead of exponential loss for improving the learning algorithm.
7	Freund et.al [14]	Boosting-By-Majority	"Boosting a weak learning algorithm by majority".	Applied the iterative boosting algorithm to improve the learning algorithm.
8	Freund et.al [15]	BrownBoost	"An adaptive version of the boost by majority algorithm".	Introduces noise tolerance property to improve accuracy.

C. Representational: In large cases of machine learning, the hypothesis space may not be expressed by a true function. By applying the various quantity of the hypothesis having weights, it is feasible to enlarge the space of the expressible function. The algorithms like neural network and decision trees it explores all the search space of the classifiers if the training data is large, so there must be an efficient space of hypothesis searched by learning algorithm.

the boosting algorithm merges the weak predicted rule in to single predicted rule that will be more accurate than the weak predicted rule. The fundamental question has to be answered; how the weights or distribution is selected at each round. Secondly, how the weak predicted rule is combined in to single predicted rule. The answer for former is that, the most weight should be put on the examples as it is misclassified by the previous classifier alternatively the base

learners then focus on the hardest examples. The answer for later is that, having their majority vote of their predictions. Adaboost Algorithm [8] focuses on maintaining the weights over the training data. At each round, the weights of examples that are incorrectly classified are expanded so that base learner can spotlight on the hardest examples in training dataset. The final classifier is obtained by having the majority vote of the base classifiers. LogitBoost [9] recommends, fitting the additive logistic regression model, to reduce the loss function. AdaBoost.M1 [10] is a variation of Adaboost algorithm in which, there is multiclass learners instead of binary classifiers. AdaBoost.M2 [11] applies the one-versus-one strategy, to decrease a pseudo-loss. One-versus-one strategy decomposes a multitasking class into binary class, where the purpose of particular task is to label the instances; either it belongs to j^{th} class or k^{th} class. AdaBoost.MR [12] it reduces the ranking loss. The appropriate class belongs to the superlative class. FilterBoost [13] introduces the log loss function instead of exponential loss function that is applied in AdaBoost. The Boosting-By-Majority [14] is an iterative boosting algorithm, but it requires applying unknown parameters in advance. BrownBoost [15] is another robust version of Boosting-By-Majority, which acquires Boosting-By-Majority noise resilience property [16].

the base learner. The aggregating methods such as voting for classification and averaging for regression are applied by bagging. Bagging uses a precedent to its base classifiers to obtain its outputs, votes its labels, and then obtains the label as a winner for the prediction. Bagging can be used with binary and multi-class classification. Bagging can scale down the deviation of larger-order items; still it does not change the linear factor. This signifies that bagging is highly tested exceptionally with non precise learners. Random Forest [19] is an extension of bagging, where the major contrast with the bagging, is the fusion of random feature selection. In the execution of a basic decision tree, at every stride of selecting the splits, random forest it promptly selects the feature subsets, and then accomplishes the selection of splits method within the preferred feature subset. Bagging applies determinist decision tree that demands to classify all features for selection of split method, while random forest applies random decision trees that commits to classify a feature subset. Random Forest develops the random decision trees by choosing the feature subset arbitrarily at every node, where as the selecting the splits inside the feature subset that is selected are closed determinate. Liu et.al [20] introduced the variable random tree (VR-Tree) ensemble approach, where the random decision trees are generated by arbitrating the selection of a

TABLE II. TAXONOMY OF BAGGING ALGORITHM

S.No.	Author	Algorithm	Paper	Strategy/ Methods
1	Breiman et.al [19]	Random Forest	"Random forests".	Random forest applies random decision trees to execute a feature subset instead of the deterministic decision tree that needs to evaluate all features for split selection in order to avoid large computations.
2	Liu et.al [20]	Variable Random tree (VR-Tree)	"Spectrum of variable-random trees".	Random decision trees are generated by arbitrating the selection of features and the selection of split processes to improve the accuracy.
3	Liu et.al [23]	Isolation Forest (iForest)	"A decision-theoretic generalization of on-line learning and an application to boosting".	The height limit is set to random trees or a setting the limit on the tree depth to avoid needless computation.
4	Liu et.al [24]	Split selection Criteria iForest (SCiForest)	"On detecting clustered anomalies using SCiForest".	SCiForest applies the hyper-plane obtained from the sequence of actual features to get the smoother decision boundaries.

IV. BAGGING

Bootstrap AGGregatING [17]. Bagging algorithms it combines the bootstrap and aggregation and it represents as parallel ensemble methods. Bootstrap sampling [18] is applied by bagging to acquire the subsets of data for training

features and the selection of split processes. In anomaly detection [21] the data points with very low densities are treated as anomalies. Liu et.al [22] reveals that, density may not be considered as a sufficient factor for anomaly, the reason is the small clustering faction of anomaly points

might have a large density and the normal points in the border that might have the low density. With this recommendation, ensemble random tree may work well for anomaly detection, and a random tree is efficient for calculating the adversity of isolating data points. Liu et.al [23] also mentioned the Isolation Forest (iForest) approach for anomaly detection. For every random tree, the total number of segregation required to isolate a data point can be calculated by the length of its path, from the root of the node to the leaf of the node consisting that data point. The fewer the segregation, it is easier to isolate the data points. The short path lengths of data points are of great importance. Thus, to reduce unnecessary computational time, the height limit is set to random trees or a setting the limit on the tree depth. Liu et.al [24] introduced the Split selection Criteria iForest (SCiForest), which is another version of the iForest. For the construction of random tree the iForest it applies the axis parallel methods by splitting the original features, whereas for smoother decision boundary the SCiForest applies the hyper plane obtained from the sequence of actual features. Buja et.al [25] analyzed bagging by applying the U-statistics, and came to the conclusion that the variance effect on bagging is least.

with labels of the contemporary dataset. The cardinal learner is obtained by applying different learning algorithms, often generating composite stacked ensembles, although the uniform stacked ensembles can be constructed. The contemporary dataset has to be obtained by the cardinal learner, otherwise if the dataset is same for the cardinal and a Meta learner there is a speculation of over fitting. Wolpert et.al [27] emphasized on various features for contemporary dataset, and the categories of learning algorithms for the Meta Learner. Merz et.al [28] introduces a stacking approach called SCANN that applies a correlation analysis to find correlations among the predictions of the base-level classifiers. The values of the class predictions are then reconstructed, in order to eliminate the dependencies. At the new feature space the nearest neighborhood is used as Meta classifier. Ting et.al [29] applies the base-level classifiers in which the predictions are the probability distribution whose predictions are probability distributions over a set of class values, relatively than single class values. Seewald et.al [30] introduces a method for combining classifiers called grading that learns a Meta level classifier for each base-level classifier. A stacked regression [31] designs the linear combinations of the distinct predictors to give the improved

TABLE III. TAXONOMY OF STACKING ALGORITHM

S.No.	Author	Algorithm	Paper	Strategy/ Methods
1	Wolpert et.al [27]	Stacked Generalization	"Stacked generalization".	Each classifier is initially trained on block of a training data. Classifier is assessed on the block which is not accessed in training. This strategy helps, as the combination rule for the primary classifiers.
2	Merz et.al [28]	SCANN	"Using correspondence analysis to combine classifiers".	The values of class predictions are reconstructed, in order to eliminate the dependencies.
3	Ting et. al [29]	Multi Response Linear Regression	"Issues in stacked generalization".	Rather than single class values, the probability distribution of those predictions are the probability distributions, upon the set of class values.
4	Seewald et.al [30]	Grading	"An evaluation of grading classifiers".	For each base-level classifier, it learns a Meta level classifier.

V. STACKING

In stacking [26] the independent learners are combined by the learner. The independent learners can be called as cardinal learner, while the combined learner is called Meta learner. The concept of stacking is to train the cardinal learner with the initial datasets to generate the contemporary dataset to be applied to the Meta learner. The output generated by the cardinal learner is the input features

prediction accuracy. The purpose is to apply cross-validation data and least squares, beneath non-negativity constraints to obtain the coefficients in the sequence. Stacked generalization [32], initially creates primary classifier that depends on cross validation of the partition of training data. The whole training dataset is split in blocks, then all the classifier is initially trained upon blocks of the training data. Then all the classifiers are assessed, upon the

block which is not accessed at the time of training. The classifiers output on the training blocks create the training data for the Meta classifier that finally provides a combination rule for the primary classifiers.

VI. Mixture of Experts

Mixture of experts [33] uses trainable combiner. It trains the ensemble of classifiers by applying a technique known as sampling. By a combination of weighted rule, classifiers are then coupled. The gating network determines the weighted combination rule of the classifiers. This gating network is then trained on the training datasets by the expectation-maximization (EM) [34] algorithm. The weights obtained from gating network are assigned dynamically, that are designated to the input of Mixture of experts. The part of feature space learned by individual ensemble representative is efficiently learned by the mixture of experts. Sometimes the mixture of experts, likewise called classifier selection algorithm, in which each classifiers are trained to develop as experts. The relevant classifier is selected with the help of combination rules. The combining system chooses an individual classifier by its maximum weight or by calculating the sum of the class is selected having the highest weight.

VII. Conclusion

In this study we discussed an important concept called ensemble learning that is prevalent methodology in machine learning. The advantage of ensemble learning and its approaches such as, Boosting that builds a strong classifier from the number of base learners. Bagging, which combines the bootstrap and aggregation and it is represented as a parallel ensemble method. Stacking, in which the independent learners are combined by the learner and a mixture of experts that trains an ensemble of classifiers by applying a technique called sampling is the important contributions of our study. In future we propose to introduce a machine learning based model that applies an ensemble learning techniques.

References

[1] Zhi–Hua Zhou, "Ensemble Methods Foundation and Algorithm", CRC press: Taylor and Francis Group, (2012).

[2] J.Huang, Z.-H. Zhou, H.-J. Zhang, and T. Chen, "Pose invariant face recognition". In Proceedings of the 4th IEEE International Conference on Automatic Face and Gesture Recognition, pp: 245–250, Grenoble, France, (2000).

[3] Giacinto, F. Roli, and L. Didaci, "Fusion of multiple classifiers for intrusion detection in computer networks". In: Pattern Recognition Letters, pp: 1795–1803, (2003).

[4] Hansen L, Solomon P, "Neural Network Ensembles". In: IEEE transactions on pattern analysis, pp: 993-1001, (1990).

[5] Thomas. G. Dietterich, "Ensembles methods in machine learning." In: International workshop on multiple classifier systems. pp: 1-15, Springer, (2000).

[6] Blum A, Rivest R.L, Training a 3-node neural network is NP-Complete. In: Proceedings of the 1988 workshop on computational learning theory, pp: 9-18, San Francisco, CA. Morgan Kaufmann, (1988).

[7] R. E. Schapire, "The strength of weak learnability". In: Machine Learning, pp: 197–227, doi: 10.1007/BF00116037, (1990).

[8] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting". Journal of Computer and System Sciences, pp: 119–139, (1997).

[9] Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: A statistical view of boosting (with discussions)". In: Annals of Statistics, pp: 337– 407, (2000).

[10] Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting". Journal of Computer and System Sciences, pp: 119–139, (2000).

[11] Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting". In: Journal of Computer and System Sciences, pp: 119–139, (2000).

[12] E. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions". Machine Learning, 37(3): pp: 297–336, (1999).

[13] K. Bradley and R. E. Schapire, "FilterBoost: Regression and classification on large datasets". Advances in Neural Information Processing Systems pp: 185–192, MIT Press, Cambridge, MA, (2008).

[14] Freund, "Boosting a weak learning algorithm by majority". Information and Computation, pp: 256–285, (1995).

[15] Freund, "An adaptive version of the boost by majority algorithm". Machine Learning, pp: 293–318, (2001).

[16] A. Aslam and S. E. Decatur, "General bounds on statistical query learning and PAC learning with noise via hypothesis boosting". In: Proceedings of the 35th IEEE Annual Symposium on Foundations of Computer Science, pp: 282–291, Palo Alto, CA, (1993).

[17] L. Breiman, "Bagging predictors". Machine Learning, pp: 123–140, (1996).

[18] B. Efron and R. Tibshirani, "An Introduction to the Bootstrap". Chapman & Hall, New York, NY, (1993).

[19] L. Breiman, "Random forests". Machine Learning, pp: 5–32, (2001).

[20] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Spectrum of variable –random trees". Journal of Artificial Intelligence Research, pp: 355- 384, (2008).

[21] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey". ACM Computing Surveys, 41(3):1–58, (2009).

[22] F. T. Liu, K. M. Ting, Y. Yu, and Z.-H. Zhou, "Spectrum of variable-random trees". Journal of Artificial Intelligence Research, pp: 355–384, (2008a).

[23] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest". In: Proceedings of the 8th IEEE International Conference on Data Mining, pp: 413–422, Pisa, Italy. (2008b).

[24] F. T. Liu, K.M. Ting, and Z.-H. Zhou, "On detecting clustered anomalies using SCiForest". In: Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, Barcelona, Spain. pp: 274–290, (2010).

[25] A. Buja and W. Stuetzle, "The effect of bagging on variance, bias, and mean squared error". Technical report, AT&T Labs-

Research, (2000a).
[26] P. Smyth and D. Wolpert, “Stacked density estimation”. Advances in Neural Information Processing Systems, pp: 668–674, MIT Press, Cambridge, MA, (1998).
[27] D.H.Wolpert, “Stacked generalization”. Neural Networks, pp: 241–260, (1999).
[28] Merz C. J, “Using correspondence analysis to combine classifiers”. Machine Learning, pp: 33-58, (1999).
[29] Ting, K. M., & Witten, I. H, “Issues in stacked generalization”. Journal of Artificial Intelligence Research, pp: 271-289, (1999).
[30] Seewald, A. K., & Furnkranz, J, “An evaluation of grading classifiers”. Proc. Fourth International Symposium on Intelligent Data Analysis, pp: 221-232. Berlin: Springer, (2001).
[31] L, Breiman, “Stacked Regression”. In: Machine Learning, pp: 49-64, (1996).
[32] D. H. Wolpert, “Stacked generalization”. Neural Networks, pp: 241–259, (1992).
[33] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, “Adaptive mixtures of local experts,” Neural Computation, vol. 3, no. 1, pp: 79–87, (1991).
[34] A. Dempster, N. Laird, and D. Rubin, “Maximum likelihood from incomplete data via the EM algorithm.” Journal of the Royal Statistical Society, pp: 1–38, (1977).