Machine Learning
COSC 6342
Fall 2019
Final Project

A COMPREHENSIVE ANALYSIS OF ENSEMBLE LEARNING WITH A NEOTERIC APPROACH TO CASCADE GENERALISATION

by

DINESH REDDY MARRI - 50559779

Introduction:

Ensemble Learning:

Ensemble learning is a part of machine learning where more than one learning algorithms are combined to increase the predictivity of a solution. Ensemble learning combines different learning algorithms in a manner to increase the efficiency of the combined model. Ensemble learning can be classified based on two factors: Change of datasets, Change of algorithms.

Change of datasets:

Bagging (Bootstrap aggregation):

Bagging is an ensemble learning algorithm where we divide the datasets into small bags randomly and train with an algorithm. The same algorithm is trained several times with different datasets. The bootstrap data may contain overlapped data i.e. data may be repeated in different bags and trained with the algorithm. Due to the bags being filled with random datasets the algorithm may not be trained with some parts of the datasets i.e. some data may not be added to any bag and may not be trained by the algorithm. Bagging is mainly used to reduce the variance of the classifier algorithm. When we test bagging with a random dataset the results from each bag classified is put into a vote. The result with the majority vote wins. Bagging usually decreases the variance of the predictions. Bagging takes place in parallel as shown the below graph:

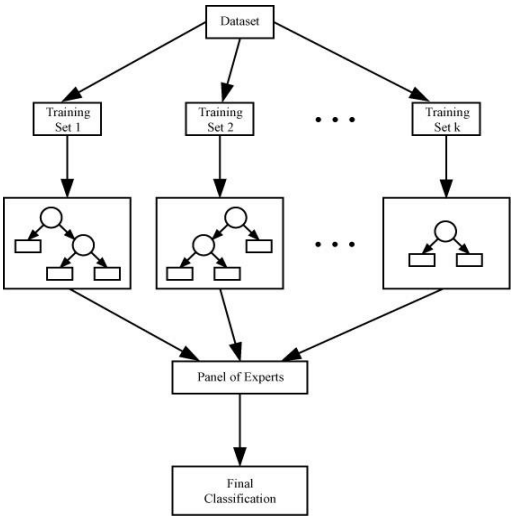

Image Source: https://upload.wikimedia.org/wikipedia/commons/a/ac/DTE_Bagging.png

Boosting:

Boosting is where a set of weak classifiers are combined to form a strong classifier. In boosting a weak classifier is improved by increasing the weights of a wrongly classified one in multiple iterations. Once the dataset is trained with a weak classifier the weights of wrongly classified examples are increased and in the next iterations, the wrongly classified data is selected by the classifier and trained by the classifier. A process of weighted voting is used in boosting.
Boosting reduces the bias of the model.

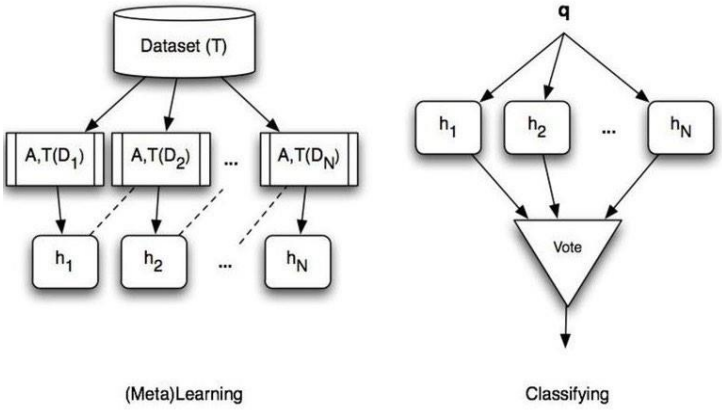

Image Source: https://slideplayer.com/slide/16133675/

Change of algorithms:

Mixture of Experts:

Mixture of experts algorithm defines which algorithm to use for a feature space in a dataset. In this model, we will have a gating network that activates the suitable algorithm for a feature space. Mixture of experts trains individual models to become experts in different regions of the feature space [5].

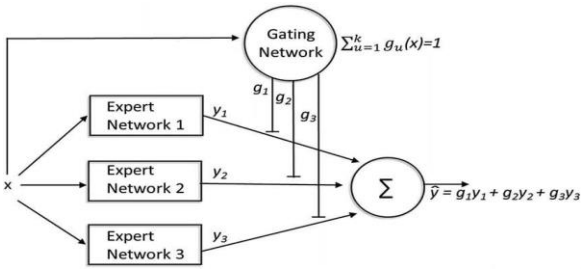

Image Source: *[5]*

Stacking:

Stacking is where we test the same dataset with different algorithms and form different predictions which are eventually combined to form a meta dataset that is finally trained with a meta-learning algorithm for the final prediction. When an example is given it should first be converted into a meta-example with values of different classifier algorithms. Then it can be fed into a meta-learner for the final prediction. The feature space of the meta-dataset depends on the number of algorithms while the size of the meta-dataset remains the same.

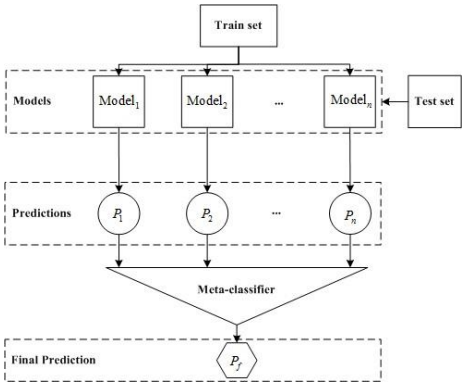

Image Source:
https://www.researchgate.net/figure/The-architecture-of-the-stacking-ensemble-learning-In-the-base-classifiers-the-training_fig1_335156833

Cascade generalization:

Cascade generalization is a technique like stacking but done in a sequential way rather than a parallel way like in stacking. In cascade generalization, we take the prediction of an algorithm and add it as a new column to the dataset which is used for the training of the next algorithm.

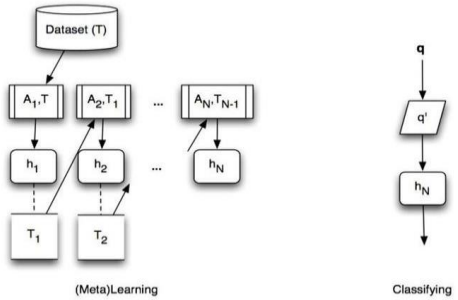

Image Source: https://slideplayer.com/slide/16133675/

Cascading:

Cascading is like boosting where the classification is based on confidence in the prediction of an algorithm. The next algorithm is trained only if the previous model is below a certain threshold. Cascading stops when confidence is high. First, we train a model with the dataset to get a prediction and when the confidence is low on the model then we increase the weight of that and train with a new dataset like boosting. In testing, we pass our example through the first model and if confidence is high then that will be our prediction else, we will pass it through our next model and so on until the confidence is high.

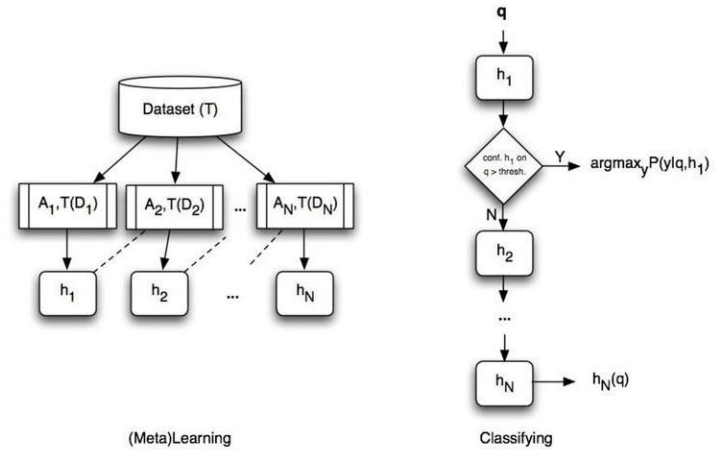

Image Source: https://slideplayer.com/slide/16133675/

Error-correcting output codes (ECOC):

ECOC is an ensemble learning algorithm that combines many base learners. It creates a matrix where the number of rows is the number of classes and the number of columns is always the number of learning algorithms. A code in the ECOC matrix is a sequence of predictions from the given set of models. Although the more models we have, the better it is but the cost increases with the number of models. The optimal number of models to be used must be decided. +1 in the matrix indicates a class that is positive. For a new test example, the class would be the one with minimum hamming distance from the test code. If multiple classes are closest to the current test code then we expand the matrix by considering pairwise combinations of classes. We also expand the matrix by considering all the combinations of classes.

Binary Classification Problems

Classes	BP1	BP2	BP3	BP4	BP5	BP6	BP7
y1	+1	+1	+1	+1	+1	+1	+1
y2	+1	+1	+1	-1	-1	-1	-1
y3	+1	-1	-1	+1	+1	-1	-1
y4	-1	+1	-1	+1	-1	+1	-1

Proposed Methodology:

In ensemble learning, while we use different algorithms in a sequential manner such as in cascade generalization where we update the dataset with the prediction from the previous model. Our proposed method deals with how to find the best sequence the algorithms should be trained in order to obtain an optimal result. A sequence in this paper is a cascade generalization algorithm with the order in which the algorithms are trained. If there are "n" algorithms to be trained, then "n!" sequences are possible.

Implementation:

Cascade generalization is a variation of the stacking algorithm. It is an ensemble learning mechanism where a combination of various classifiers is considered to improve the accuracy of the prediction. All classifiers are trained with original feature space. Attributes built by extending the dataset are also considered as original features by the higher-level models of the framework. Cascade generalization is considered computationally efficient compared to other combinational ensemble models.

The sequence in which the set of algorithms are applied, greatly influences the final prediction capability of the framework. As the extended dataset depends on the previously trained model and its predictions, the higher-level models are to be placed optimally to get the best predictions.

In cascade generalization, it is recommended that algorithms with low variance are preferred at a low level, whereas, algorithms with low bias at higher levels. Under equal performances, algorithms with a smaller number of components are placed at lower levels.

Algorithm:

1. Initially, the dataset is trained with algorithm-1 and predictions are obtained using this trained model.
2. The predictions from this model are attached to the original dataset as a new attribute to form an extended dataset.
3. This extended dataset is now used to train algorithm-2.
4. Repeat steps 2, 3 for the given number of algorithms N.
5. We use the final model thus obtained to predict test data.
6. For a new test example, to make the predictions we first need to enlarge its feature space with the predictions from the models and feed that to the last model.
7. Repeat steps 1, 6 for N! sequences of N algorithms. Output the maximum accuracy obtained and its sequence of algorithms used.

We implemented a code that outputs the best possible sequence of the given algorithms with which we can obtain maximum accuracy.

Experimental Results:

Below are the train and test accuracies from cascade generalization using K-Nearest neighbor, Stochastic gradient descent and Random forest classifiers run with different sequences. From the below data we can observe a significant change in test accuracies with different sequences.

On the left, we used the "Diabetes" dataset, here the test accuracy was maximum (Around 78%) when the sequence of algorithms applied in the cascade are SGD, KNN, RFC.

On the right, we applied cascade generalization on the "Wine Classification" dataset, with same configurations, the test accuracy, in this case, is observed to be maximum (Around 97.3%) when the sequence is KNN, SGD, RFC. Also, we can observe that the difference between obtained training and testing accuracies is low and that model is not Overfitting the dataset.

Experimental accuracies:

KNN -> SGD -> RFC
Train accuracy: 0.7980456026058632
Test accuracy: 0.7727272727272727

-> SGD
Train accuracy: 0.501628664495114
Test accuracy: 0.4935064935064935

SGD -> KNN -> RFC
Train accuracy: 0.8061889250814332
Test accuracy: 0.7857142857142857

KNN -> SGD -> RFC
Train accuracy: 0.9831233122222222
Test accuracy: 0.9732222222222222

KNN -> RFC -> SGD
Train accuracy: 0.6267605633802817
Test accuracy: 0.6077777777777778

SGD -> KNN -> RFC
Train accuracy: 0.9841333122222222
Test accuracy: 0.9722222222222222

------------------------------ ------------------------------
SGD -> RFC -> KNN SGD -> RFC -> KNN
Train accuracy: 0.7850162866449512 Train accuracy: 0.8087323943661971
Test accuracy: 0.7532467532467533 Test accuracy: 0.7833333333333334
------------------------------ ------------------------------ RFC -> KNN -> SGD
RFC -> KNN -> SGD
Train accuracy: 0.6237785016286646 Train accuracy: 0.704225352112676
Test accuracy: 0.6168831168831169 Test accuracy: 0.6855555555555556
------------------------------ ------------------------------ RFC -> SGD -> KNN
RFC -> SGD -> KNN
Train accuracy: 0.7850162866449512 Train accuracy: 0.7987323943661971
Test accuracy: 0.7532467532467533 Test accuracy: 0.7633333333333334 Each classifier makes errors in different regions. So, by combining different classifiers we tend to reduce the errors caused by classifiers when used individually.

Confusion Matrices of Sequences:

In Fig 3.1 we can see the confusion matrices for the "Diabetes" dataset with different sequences. We can infer from the following confusion matrices that with varying sequences the predictions also vary. And the optimal sequence for the given problem is SGD, KNN, RFC.

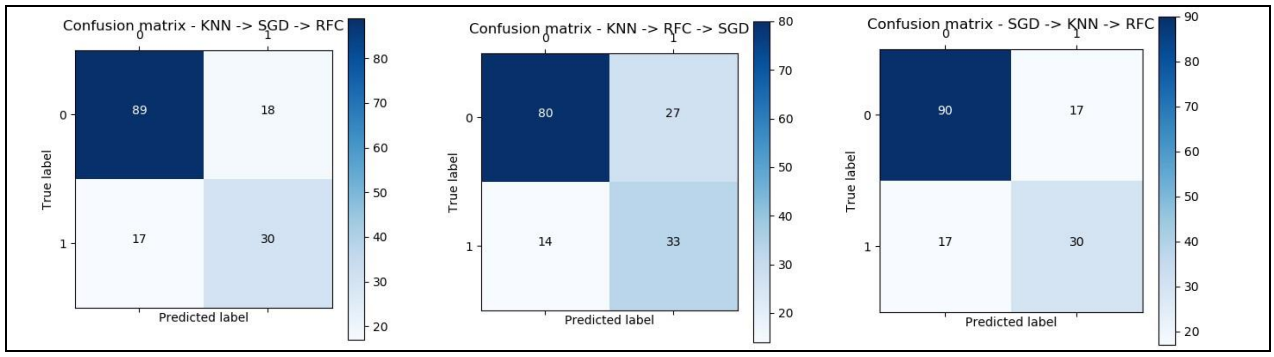

Fig-3.1

Fig-3.2 are the curves for confusion matrices for the "Wine Classification" dataset. We can observe that the sequence is changed with the new dataset. We can infer from the Fig-3.2 that the sequence of KNN, SGD, RFC gives high accuracy.

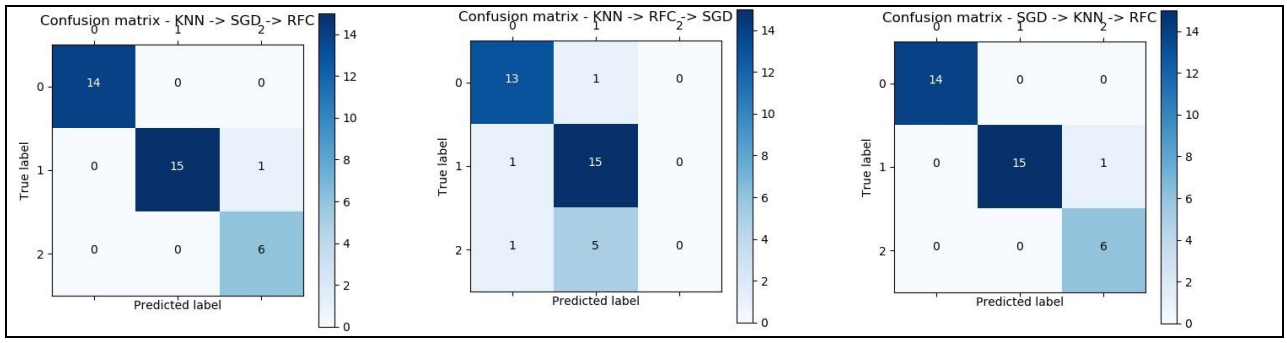

Fig-3.2

Receiver Characteristic Curves:

The ROC Curves in Fig-3.3 is for the "Diabetes" dataset shows that the curve is closer to the left hand and top border for cascade generalization predictions if the sequence of algorithms used is SGD, KNN, RFC.

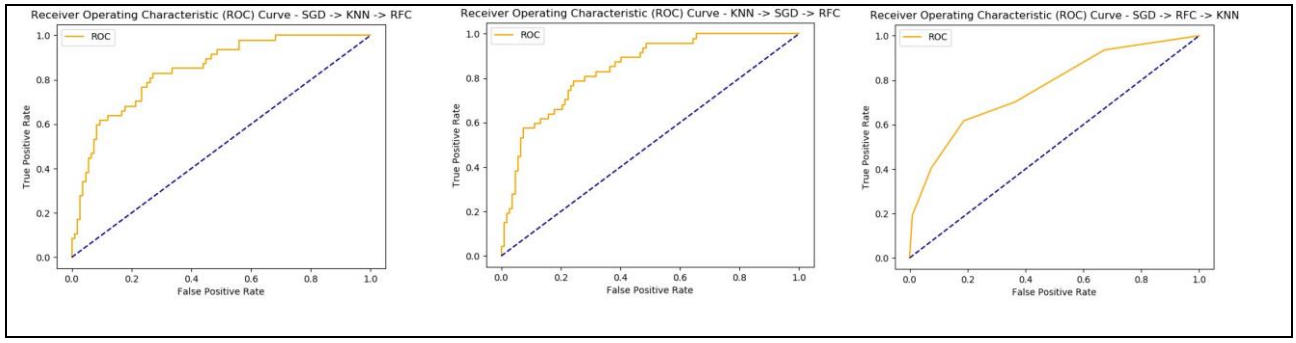

Fig-3.3

Comparison of Stacking and Cascade Generalization:

Stacking employs the predictions of the base-level classifiers to form a meta dataset. However, cascade generalization uses the predictions from low-level classifiers to extend the size of the feature space. Stacking is parallel in nature whereas cascading is sequential. Cascade generalization allows intermediate classifiers to have access to original attributes plus the predictions from the low-level models.

The below confusion matrix is obtained from stacking with XGB boosting as a meta-learning algorithm on Diabetes dataset. The final prediction score in this scenario is around 75%, while cascade generalization with the best sequence produced a prediction accuracy of 78%.

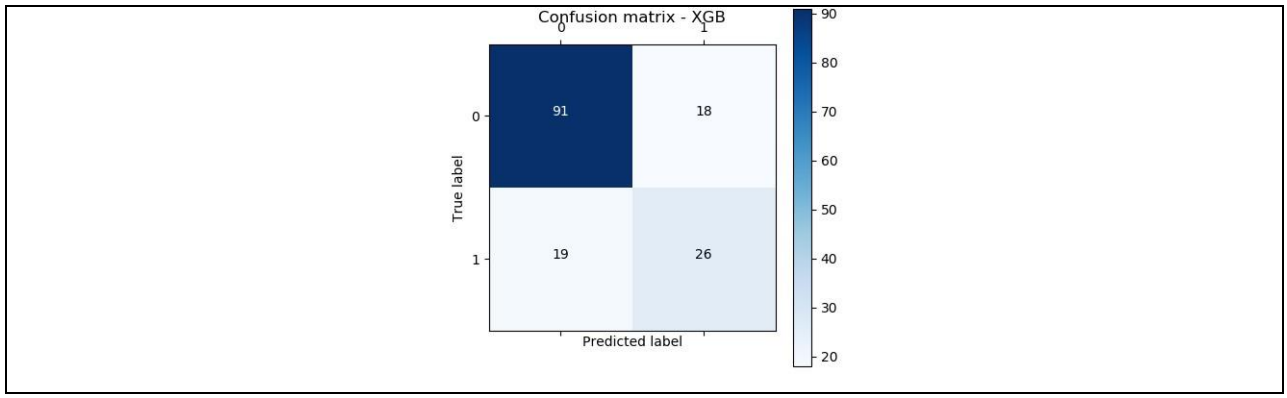

Fig-3.4

Final prediction score - Stacking: **0.7597402597402597**

Conclusion:

Ensemble learning is a powerful tool in machine learning to boost accuracies by combining different classifiers. The basic idea of this method is to combine different classifiers for training the model and get better accuracies. To achieve this we explored two different ensemble methodologies, Stacking and Cascade generalization. With various experimental results, we observed that there is a change in accuracy scores with varying sequences. We came up with a project to train the dataset with a defined sequence of classifiers in Cascade generalization, such the accuracy is maximum. It is recommended to use algorithms with low variance at a low level and algorithms with low bias at a high level. Cascading with such sequences train better models than other sequences. We used two datasets, namely, Diabetes and Wine classification for testing and validation. Diabetes is a binary classification dataset whereas Wine classification is a multiclassification problem. Obtained experimental results from Stacking and Cascade generalization show that the model trained with the recommended sequence of classifiers gives better accuracy than the model obtained from stacking.

References:

[1] http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.46.635&rep=rep1&type=pdf

[2] https://pdfs.semanticscholar.org/3fb5/fe6cb4c32d3c9077e2d8adf08248deec25a1.pdf

[3] https://towardsdatascience.com/simple-guide-for-ensemble-learning-methods-d87cc6870 5a2

[4] https://pdfs.semanticscholar.org/3fb5/fe6cb4c32d3c9077e2d8adf08248deec25a1.pdf

[5] https://www.commonlounge.com/discussion/1697ade39ac142988861daff4da7f27d

[6]https://medium.com/@saugata.paul1010/ensemble-learning-bagging-boosting-stacking-a nd-cascading-classifiers-in-machine-learning-9c66cb271674

[7] https://www.researchgate.net/publication/230867318_Ensemble_methods_A_review