Weekly report of lessons

Name: Suryam Arnav Kalra
Roll No: 19CS30050
The week: 25th October 2021 to 29th October 2021

The topics covered:

- No learner perfect!
- Issues on combining learners
- Techniques of diversification
- Diversity vs Accuracy
- Model combination schemes
- Classifier combination rules
- Bayesian combination rule
- Error correcting output codes
- Bagging and Boosting
- Boosting by three weak learners in tandem
- AdaBoost (Adaptive Boosting) and AdaBoost.M1 (The original algorithm)
- Mixture of experts, Stacked generalization and Cascading

Summary topic wise:

- No learner perfect: No single learning algorithm in any domain always induces the most accurate learner, each learning model dictates a certain model that comes with a set of assumptions. A suitable combination of learners should improve accuracy.
- Issues on combining learners: The space and time complexity increases and the model combination may not increase accuracy. If we maximize individual accuracies and diversity between learners, we may get good results.
- Techniques of diversification: We can use different algorithms and different hyperparameters for the same learning algorithm to achieve diversification. We can use different input representations (random sub-space, sensor fusion), different training sets (bagging and boosting).
- Diversity vs Accuracy: The base learner should be simple and should perform marginally better than random guess and they should be diverse providing high accuracies on different instances of the problem.
- Model combination schemes: We can use voting, stacking (global approach) and mixture of experts, gating model (local approach) for getting high accuracy.
- Classifier combination rules: We can use different types of decision fusion functions such as: Sum or average, weighted average, max, min, median, product. Median rule is more robust to outliers, minimum and maximum are pessimistic and optimistic approaches and the product rule empowers each learner veto power.
- Bayesian combination rule: We can use weights approximating prior probabilities of models. Instead of all models in the space, we choose only those who have high $P(m_j)$. For each classifier if $P(error) < 0.5$, with the increase of number of classifiers, accuracy increases by majority voting.
- ECOC: The ECOC method is a technique that allows a multi-class classification problem to be reframed as multiple binary classification problems, allowing the use of native binary classification models to be used directly. Let $y_i = \sum w_{ij} d_j$, then we choose the class with the highest y_j.

- Bagging: It is a voting method in which each base learner is trained over slightly different training sets. They are of similar structure but with different set of parameters and allow sampling with replacement. It can be used for both classification and regression problems but if a small change in data causes large variation in model then the learning is unstable.
- Boosting: In this method, we generate complementary base learners in which we train the next learner from the mistakes of the previous learner lowering the chance factor and instability of the training algorithm.
- Boosting by three weak learners: We randomly divide the training samples into 3 sets X_1, X_2, X_3 and train d_1 with X_1 and test d_1 with X_2. We then form a training set X_2' for training d_2 and train d_2 with X_2' and test X_3 with d_1 and d_2. Then train d_3 with instances disagreed by d_1 and d_2. For testing, if a sample X has same labels by d_1 and d_2, accept it, else accept the result from d_3.
- AdaBoost: We use the same training set over and over and combine an arbitrary number of base learners, not just three and each leaner should perform with error rate < 0.5 and they should not overfit the data.
- AdaBoost.M1: At each iteration i, train with the sample set and compute the training error e of classification. If $e > 0.5$, then stop otherwise include the model d_i in the list and update the sampling probability of each t^{th} training sample by decreasing the weight of correctly classified samples as $p = w.p$ where $w = e/1 - e$. During testing, calculate the outputs y_i and assign the class with maximum y.
- Mixture of experts: In voting weights are fixed for each classifier. In this method, depending upon inputs these weights may vary. The final classification score for each class is the weighted means of votes.
- Stacked generalization: In this method, instead of linear combination of weights it could be any general function with parameters ∅, which could also be learned. It could also be a multilayer perceptron.
- Cascading: The base learners are ordered in terms of complexity (cost) and each learner produces output (y) with a confidence (w). The next learner is used only if the previous learners' decision lacked confidence.

Concepts challenging to comprehend:

AdaBoost is a little bit challenging to comprehend.

Interesting and exciting concepts:

Bagging and boosting are quite interesting and exciting to learn.

Concepts not understood:

After going through the book and the video lectures the concepts are understood.

Any novel idea of yours out of the lessons:

I think that we can also use gradient descent technique for finding the optimal set of learners. If we imagine the data points to be in a N-dimensional space, we can represent a point with the true labels of the data points. Now, we can consider different learners. A strong learner will be very close to the true point and a weak learner will be far from the true point. So we can randomly choose some base learners and using a gradient descent algorithm we can minimize the error (get them close to the true point) by making a linear combination of the models. I think that it might perform better than AdaBoost.