IITM-CS5691 : Pattern Recognition and Machine Learning | Release Date: 29th Apr, 2021
New Method Writeup | Due Date: 28th May, 2021

Roll No: CH18B022 | Name: Shashank M Patil

- Dear Student, You may have tried or thought of trying different methods for your data contest. Choose one of the methods that was not taught in class, and submit a writeup of this "new method" in the template provided below. This is an individual submission - i.e., while you would've done your kaggle submission as a team of two members or while you may've discussed this method with your teammate, **you will have to write about the new method in your own words independently and submit it individually**.
- **Template:** Fill in whatever fields are applicable for your algorithm (overall 1-2 page writeup; since some fields may not be applicable for certain methods, we haven't shown points below).

1. (points) [Name of Method, and its ML Problem and Paradigm: problem could be regression/classification/clustering/etc., and paradigm could be supervised/unsupervised/..., generative/discriminative/direct, linear/non-linear models, etc.)]:

Solution:
Method Used: Gradient Boosted Decision Trees (GBDT) or Gradient Boosting Algorithm
ML Problem : Regression
Paradigm: Supervised Learning

The above algorithm was implemented using the LightGBM (Light Gradient Boosted Machine) library/framework by Microsoft which provides efficient algorithm implementation and in many cases, also results in a more effective model.

2. (points) [Brief introduction/motivation: One paragraph to describe briefly the new method (its name, what it does, its main application, etc.)]

Solution:
GBDT is an ensemble ML algorithm that fits boosted decision trees by minimizing an error gradient. The objective is to minimise the loss of the model by adding weak learners (decision trees) using a gradient descent procedure. Therefore the name, Gradient Boosted Decision Trees.

It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.This enables the technique to be applied beyond binary classification problems to support regression, multi-class classification etc.

3. (points) [Closely related method seen in class, and relation of your selected new method to method you eventually used for the data contest]:

Solution:
Closely related method seen in class is the Adaboost or Adaptive Boosting Algorithm. The weak learners in AdaBoost are decision trees with a single split, called decision stumps for their shortness but in the case of GBDT we usually have deeper trees. AdaBoost works by weighting the observations, putting more weight on difficult to classify instances and less on those already handled well. New weak learners are added sequentially that focus their training on the more difficult patterns.

Finally the method used in the Data Contest was also GBDT. The resulting models (strong learners) via GBDT were bagged to make the final predictions to take into account the variance.

4. (points) [Training Input and Output: (e.g., $\{x_i, y_i\}_{\{i=1...N\}}, x_i \in R^d, y_i \in \{-1, +1\}$, etc.]:

Solution:
Input: $\{x_i, y_i\}_{\{i=1...N\}}, x_i \in R^d, y_i \in \{1, 2, 3, 4, 5\}$,
Output: $y_i \in \{1, 2, 3, 4, 5\}$

5. (points) [Training Objective function (e.g., loss function that is to be optimized) or probabilistic model (over which MLE or Bayesian inference done):]

Solution:
The basic objective function in GBDT used for the problem of regression is L2 loss or the mean squared error.

$$\frac{\sum_i^n (\hat{y_i} - y_i)^2}{n}$$

$\hat{y}_i$ = the predicted value $F(x)$
y_i = the observed value
n is the number of samples

6. (points) [Training Algorithm: Brief description of key aspects of the algorithm]

Solution: Gradient boosting involves three elements:

Note: 'm' refers to the index of the loop which runs till M (total number of trees/weak learners to be learned)
1) **loss function to be optimized**: mean-squared error loss as given in the previous answer.

2)**A weak learner(decision tree) to make predictions**: LightGBM grows trees leaf-wise (best-first) where it will choose the leaf with max delta loss to grow. Holding leaf fixed, leaf-wise algorithms tend to achieve lower loss than level-wise algorithms. It should be noted that these trees are trained to fit the residuals (r_{m-1}) of the previous iteration which is basically obtained by the negative of the gradient of the loss function.

$$\mathbf{r}_{m-1} = \mathbf{y} - F_{m-1}(X)$$

3) **An additive model to add weak learners to minimize the loss function**: Trees are added one at a time, and existing trees in the model are not changed. It should be noted that the $\Delta_m(X)$ are predicted residuals from the learnt decision trees. The effect the weak learner has on the predicted output can be controlled by η,

$$F_m(X) = F_{m-1}(X) + \eta\Delta_m(X)$$

There are hyperparameters which have to be tuned to prevent overfitting mainly the maximum depth, learning rate, number of weak learners, number of leaves etc.

7. (points) [Testing Input and Output: (e.g., $x \in R^d$, $y \in \{-1, +1\}$)]

Solution: Same as that of training input and output

8. (points) [Testing Algorithm: Brief description of key aspects of the algorithm]

Solution: Same as that of training algorithm

9. (points) [Critique of the method: (1-2 paragraphs discussing its strengths and weaknesses in your own words)]

Solution:
It can be applied to a wide variety of ML tasks such as regression, classification etc as we are basically optimizing the loss function using gradient descent method, so as long as we have differentiable loss function, new algorithm need not be derived if we use a different loss function for a certain task. It also accounts for the categorical data and missing values which reduces the burden of extensive data-preprocessing. A wide range of hyperparameter tuning options provided which ensures good fit of the model.
Gradient boosting is a greedy algorithm and can overfit a training dataset quickly. Since it has fit the regression tree to the residuals it can be affected drastically by the presence of outliers in the dataset. But these can be avoided by the use hyperparameters such as maximum depth, regularization etc but the task of finding optimal hyperparameters is time intensive and computationally expensive.