Jimmy Harvin & Charles Wallis

CS4375 Intro to Machine Learning

October 23, 2022

Portfolio Component 5: Kernel and Ensemble Methods

SVM (Support Vector Machines) creates a hyperplane in which data can be classified in a three-dimensional space using a kernel of characteristics in the data. The goal of SVM is to find the maximum margin hyperplane. The hyperplane or a set of hyperplanes can be then used for classification or regression analysis. Intuitively, if a hyperplane has a large margin from the nearest learning data point, the classifier error will be small and it will perform classification with a high accuracy.

In SVM, **kernels** are used to transform the dimension of the input data into a higher dimensional feature space. In each kernel, parameters are used to optimize. There is no automatic way to tell which kernel parameter is best, so we need to find conditions that show the optimal prediction rate by repeating the learning and predicting process. Some strengths of SVM include being able to classify and predict at the same time, and that it is less prone to overfitting (and outliers do not have a big effect on the model) than neural network techniques. SVM also has a high accuracy in prediction, and the algorithms are easy to use. Weaknesses include having to test several parameters in models and kernels to make an optimized model, and models take a very long time to build. The model also does not have enough explanation by itself as well.

Decision trees refer to a model that divides training data into tree structure by certain criteria. This tree is easily interpretable, not only the prediction results, but also the explanation of why it got there. In addition, decision trees have an advantage of being easy and fast to calculate, but it takes some training time. Decision trees are simple and accurate. However, since the decision tree follows a greedy algorithm of “dividing and conquering”, and it divides partitions repeatedly from top to bottom. Once a tree makes its separation, there is no turning back. Another limitation of the decision tree is that the decision boundary (line that separates the prediction outcome) is parallel to the axis. Overfitting may occur when the training data is small, or if there is noise, or when the tree is too large. Overfitting would mean the model is accurate on training data, but may have poor performance in the

test data. To prevent this, Pruning can be done. Stopping the algorithm before it grows past a certain size, or set an impurity level to stop the model when it reaches a certain GINI index or Entropy.

Random Forest algorithm uses several decision trees to predict based on soft voting. It randomly selects a feature to process bagging. Decision trees are often adopted by other ensemble algorithms due to their easy and intuitive nature. Random Forest has relatively fast speeds among ensemble algorithms and is a very advantageous algorithm in that it performs well in various fields. Data also doesn't require scaling. The disadvantage is that tree-based ensemble algorithms have many hyperparameters, which require a lot of time for tuning. (The speed itself is faster than other algorithms.) It also has the disadvantage of relatively poor performance improvement through hyperparameter adjustment. Weaknesses of RF comes from the tree becoming overly complex, and when the dimension of data is too large the performance may drop (e.g., text data). Additional data usually does not improve the random forest model as well.

Boosting algorithms are a method of learning by sequentially learning-predicting several weak learners and improving errors by weighting incorrectly predicted data. It is called a boosting method because it continues to perform learning by boosting weights to the classifier. Conventional Boosting methods require sequential operation, so parallel operation is not possible. Therefore, large datasets can require a lot of learning time. Typical algorithms for boosting include AdaBoost and GradientBoost, and XGBBoost and LightGBM are the most popular boosting algorithms that have recently been recognized in terms of performance.

AdaBoost stands for Adaptive Boosting, which is a representative algorithm that performs boosting by weighting error data. For classification, the main idea is to weight misclassified data, making the next classifier better classified. Finally, we combine these classifiers to create a final classifier. Because Adaboost weights error data, it is sensitive to outliers. Adaboost performs better than bagging, but it is slow and is prone to overfitting.

The **Gradient Boost Machine** algorithm is almost similar to AdaBoost. However, the difference is that weight updates are made gradient descent. On average, it has better predictive

performance than random forests, but also has the disadvantage of requiring hyperparameter tuning efforts and taking longer to run. The disadvantage is that parallel performance is not possible due to sequential progress. Many GBM-based algorithms have been studied with a focus on performance, and the two most commonly used algorithms are below.

XGBoost (eXtra Gradient Boost) is an algorithm based on GBM and has several advantages. It is faster than GBM and has advantages such as overfitting regulation. In addition, both classification and regression have excellent predictive performance, and have advantages such as self-embedded cross-validation and loss handling. XGBoost performs faster than GBM using a parallel CPU. Conversely, a multicore CPU is required to expect speed. In addition, various functions such as tree pruning have improved the speed. XGBoost is faster than GBM, but still is a slower algorithm, because hyperparameter tuning using GridSearchCV takes too long.

Ensembles are a great way to improve results in machine learning models, but it also reduces interpretability. Ensemble methods are not preferred where the training process is more important, however it is still very important to have high accuracy when it comes to data prediction. Also for some fields, accuracy is more important than the process, like the medical field (we don’t want to misclassify someone to not have a heart disease, when they actually do.)