ITI104 - Machine Learning Algorithms

Assignment 1

Voting and Stacking Ensembles

Pamela Sin Hui (21A530M)

This project explores heterogenous ensembles, delving into the comparison, technical details, advantages and disadvantages between Voting and Stacking methods. A general overview of the techniques is first laid out, followed by examining the Scikit-Learn implementation of the ensembles. Finally, the pros and cons of the methods are discussed, going into factors to consider in implementing the methods and real-world considerations.

a. Comparison between the Voting and Stacking methods.

Voting and stacking often consider heterogeneous weak learners, where different learning algorithms are combined.

The fundamental difference between voting and stacking is how the final aggregation is done. In voting, user-specified weights are used to combine the base classifiers. In stacking, the aggregation is done using a meta classifier [1]. Further, in voting no learning takes place at the meta level given that the final classification is simply decided by votes casted by the base classifiers, while in stacking learning takes place at the meta level [2].

(i) Voting

In voting, the voting classifier works like an electoral system. A prediction on a new data point is based on a voting of the members of a group of machine learning models [3].

As all models have an equal contribution to the prediction, one limitation of voting ensembles is that it treats all base models the same. This can pose a problem if some models perform well in some circumstances and perform poorly in others.

(ii) Stacking

In stacking generalization, the predictions of each individual estimator are stacked together and used as input to a final estimator to compute the prediction on a new data point. The intention is to reduce the generalization error of different generalizers. Stacking is a technique for combining estimators to reduce their biases, using the strength of each individual estimator by using their output as the input of a final estimator.

Firstly, machine learning models called base estimators are fitted on the dataset. Next, the results from these base learners then serve as input into the meta classifier. The stacking classifier learns when our base estimators can be relied upon or not.

b. Some technical details on how the algorithms work.

Note that terminology and methods under the Scikit-Learn library are italicized and underlined.

(i) Voting

The below table describes the key parameters for *sklearn.ensemble.VotingClassifier [4].*

Parameter	**Input**	**Description**
Estimators	list of (str, estimator)	Base estimators.
voting	{'hard', 'soft'}, default='hard'	• 'hard' uses predicted class labels for majority rule voting. • 'soft' predicts the class label based on the argmax of the sums of the predicted probabilities.
weights	array-like of shape (n_classifiers,), default=None	• In hard voting, weight the occurrences of predicted class labels. • In soft voting, weight the class probabilities before averaging. • 'None' uses uniform weights.

A hard voting ensemble involves summing the votes for class labels from base models and predicting the class with the most votes (majority rule voting). Hard voting is appropriate when the models used in the voting ensemble predict crisp class labels.

A soft voting ensemble involves summing the predicted probabilities of the individual base learners for class labels, and predicting the class label with the largest sum probability. Soft voting is appropriate when the base learners are able to predict the probability of class membership. For models that do not inherently predict a class membership probability, this may require configuration of their probability-like scores before being used in the ensemble (examples being decision trees, support vector machine and k-nearest neighbors) [5].

(ii) Stacking

The below table describes the key parameters for *sklearn.ensemble.StackingClassifier [6].*

Parameter	**Input**	**Description**
estimators	list of (str, estimator)	Base estimators to be stacked together in parallel on the input data.
final_estimator	estimator, default=None	A classifier which uses the predictions of the base estimators as input to combine the base estimators. The default classifier is a *LogisticRegression*.
cv	int, default=None	Determines the cross-validation splitting strategy used in *cross_val_predict* to train final_estimator. • For integer inputs or ‘None’, *StratifiedKFold* is used. • Otherwise, *KFold* is used.
passthrough	bool, default=False	• If ‘False’, the final_estimator is trained on the predictions of estimators. • If ‘True, the final_estimator is trained on the predictions of estimators and original training data.

Stacking is commonly composed of 2 training stages. The first stage (level 0) generates the new training data for the meta-model. Each independent weak learner is trained on the full training data. The predictions of each weak learner are stacked to build a new training set for the meta-model. The second stage (level 1) trains the meta-model on the new training set [9].

In reference to the Scikit-Learn implementation of stacking, base estimators are fitted on the full input data, while final_estimator is trained using cross-validated predictions of the base estimators using *cross_val_predict*.

During training, the base estimators are fitted on the whole input data. They will be used when calling *predict* or *predict_proba*.

To prevent the problem of over-fitting, internally the final_estimator is trained on out-samples using *sklearn.model_selection.cross_val_predict*.

c. Advantages and benefits of the ensemble methods over single classifier/regressor.

Factors that determine if an ensemble (voting or stacking) classifier performs better than a single classifier are firstly, the correlation of base classifiers with each other and secondly, the performance of the base classifier as opposed to random guessing. If the base classifiers are independent or have a low correlation to each other, and does better than a classifier that performs random guessing, we might expect the ensemble classifier to outperform its standalone base classifiers.

The goal of ensemble methods in combining the predictions of base estimators is to improve the generalization and robustness over a single estimator. An ensemble can attain better prediction performance over any of its constituent single contributing models. Weak learners are used as building blocks for constructing more complex models, which are formed through the amalgamation of multiple weak learners. The weak learners may not perform well on a standalone basis, which can be attributed to a high bias or high variance problem. Ensemble methods combine several weak learners to reduce the bias and/or variance of the weak learners, creating a strong learner that achieves better performance. For ensembles, the minimization of the variance element of prediction errors made by the contributing weak models is frequently the mechanism through which improved performance is achieved. Another advantage of an ensemble is that it can improve robustness, reducing the spread of predictions and model performance [7].

d. Any other facts and figures about Voting and Stacking methods that you find interesting and worth sharing.

(i) Wide-spread adoption of ensembles

The popularity of ensembles increased in the late 2000s due, in part, to their large success in machine learning competitions such as Kaggle competitions and the Netflix prize competition.

(ii) Blending classifier – Variation to stacking

Blending is similar to stacking, and the concept was made popular by the winning team of the Netflix prize competition through a 10-fold performance improvement on the movie recommendation algorithm using a blended solution [8].

In blending, a small holdout set is created out of the train set, and the meta model trains on the holdout set only. Blending can be thought of as simpler than stacking, and blending reduces the risk of an information leak as the base models and meta model use different datasets. However, blending uses less data and may lead to overfitting [9].

(iii) Multi-level stacking

Multi-Layer stacking is where layers of base learners are built before a final estimator is built. Each meta-model of the different levels of a multi-level stacking ensemble model can be any appropriate learning algorithm, even if the algorithm was used at lower levels. Adding levels can be data expensive, for example, if k-folds cross-validation is not used then more data may be required. Adding levels can also be computationally expensive, for example, if k-folds cross-validation is used then models are fitted multiple times [10]

(iv) Performance of ensembles

It is possible for the performance of an ensemble to be no better, or even worse, than the best-performing member of base classifiers. In such a scenario, the ensemble has a top-performing model and the ensemble is not able to harness the contribution of other base classifiers effectively. The top-performing model's predictions are "made worse" when collectively viewed with that of other poor-performing models. Thus, the use of ensembles does not guarantee an improvement in performance, and a suite of ensemble methods should be tested and tuned.

Ensembles may not be employed in real-world settings due to relatively small improvement over the best individual model, time-consuming assembly of base models and training, expensive deployment and maintenance, difficulty in interpreting results and limitation in the amount of available data [11].

References

[1] https://www.linkedin.com/pulse/comparing-voting-stacking-optimal-pipelines-python-subrata-chatterjee

[2] https://silo.tips/download/chapter-6-the-stacking-ensemble-approach

[3] https://towardsdatascience.com/ensemble-methods-comparing-scikit-learns-voting-classifier-to-the-stacking-classifier-f5ab1ed1a29d

[4] https://scikit-learn.org/stable/modules/ensemble.html#stacking

[5] https://machinelearningmastery.com/voting-ensembles-with-python/

[6] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.StackingClassifier.html

[7] https://machinelearningmastery.com/why-use-ensemble-learning

[8] https://neptune.ai/blog/ensemble-learning-guide

[9] https://towardsdatascience.com/ensemble-learning-stacking-blending-voting-b37737c4f483

[10] https://towardsdatascience.com/ensemble-methods-bagging-boosting-and-stacking-c9214a10a205

[11] https://www.quora.com/What-are-the-limitations-of-linear-stacking-ensemble