Business Anaytics

M. Tech QROR – 2^{nd} yr (2024)

Classification

(Predictive Analytics)

Dr. Prasun Das

SQC & OR Unit

Indian Statistical Institute

e-mail: prasun@isical.ac.in

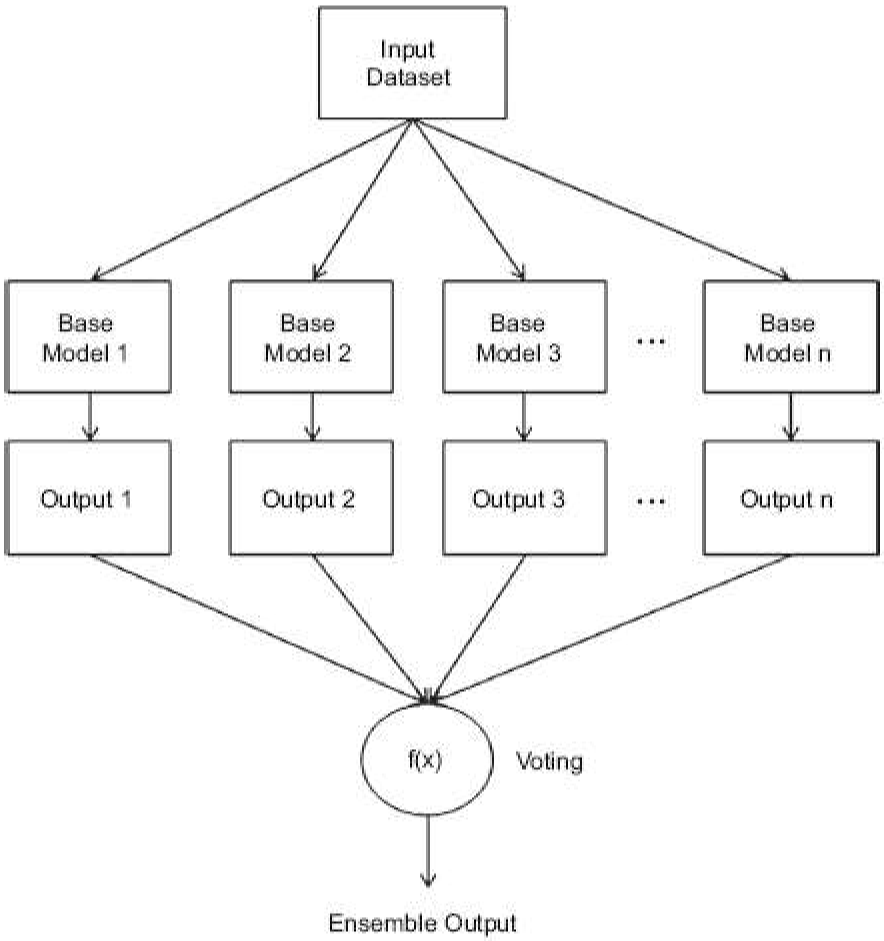
Ensemble Learners

Ensemble is a ML concept in which the idea is to train **multiple models** (*or*, **Classifiers**) using the same learning algorithm with a common objective and fuse together to solve the problem.

Ensemble helps to minimize the main causes of error in learning (i.e., **noise, bias and variance**).

These methods are designed to improve the **stability** and the **accuracy** of ML algorithms, especially in the case of unstable classifiers (like Decision tree, ANN etc.), and may produce a more reliable/accurate classification than a single classifier with much less generalization error.

Ensemble Model - Framework



Bagging and **Boosting** are two similar kinds of **ensemble techniques**, where a set of weak learners are combined to create a strong learner that obtains better performance than a single one.

Ensemble Technique - Bagging

Definition: Bagging is a technique where base models are developed by changing the training set for every base model.

Given: training set {T} of *n* records

Develop: *m* training sets (T_1, T_2, T_3, …, T_m) each with *n* records by SRSWR, containing duplicate records - This is called *bootstrapping*.

Model: a set of *m* base models, the prediction of each is aggregated for an ensemble model. This combination of bootstrapping and aggregating is called ***bagging***.

Unique records: In general, SRSWR of *n* records contains $1 – (1 – 1/n)^n$ unique records. For *n* being sufficiently large, we get 1 – 1/e = 63.2% unique records for each base training set T_i , on an average. **[Prove it…]**

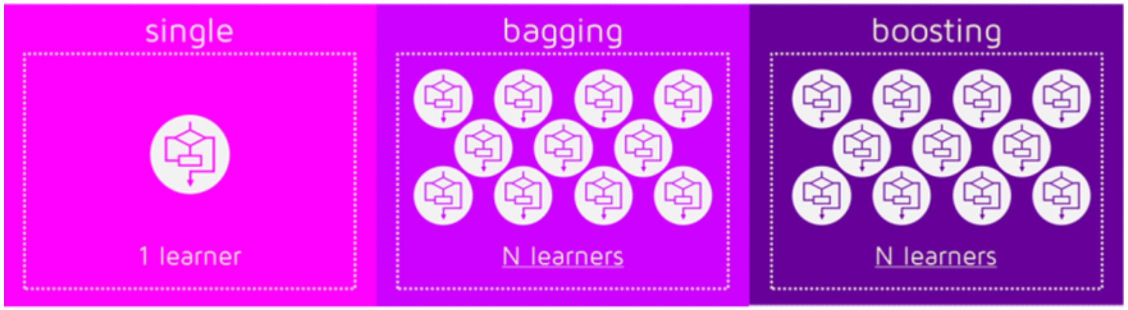
Ensemble Technique - Boosting

Definition: Boosting offers an iterative and sequential approach to building an ensemble model by minimizing bias or variance due to training records.

Process: To start with, assign equal weight to all training records. A training sample is selected based on the weights and used for model building. Then the model is used for testing with the whole training set. Incorrectly classified records are assigned a higher weight and correctly classified records are assigned a low weight, so hard-to-classify records have a higher propensity of selection as the training sample for the next round. Hence, the next model will focus on the hard-to-classify data space.

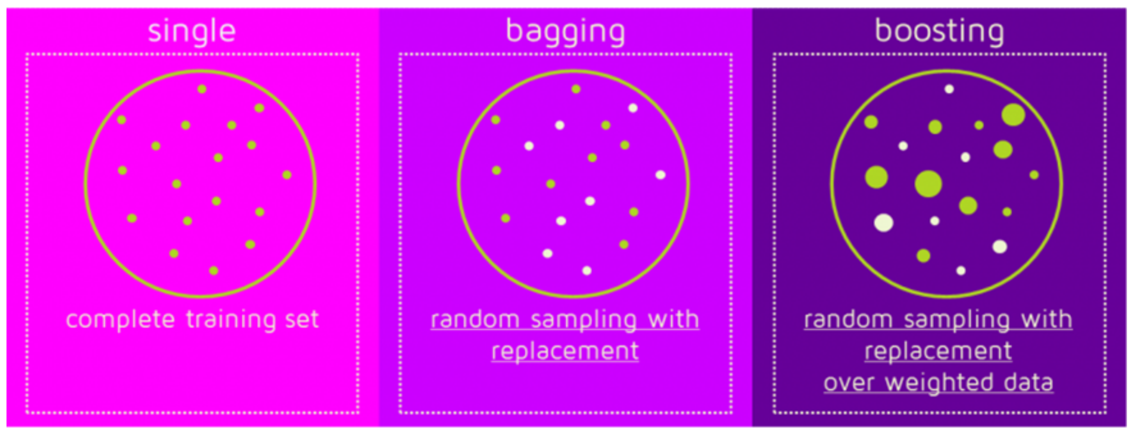
Specialty: This results in an ensemble of base learners specialized in classifying both easy-to-classify and hard-to-classify records. When applying the model, all base learners are combined through a simple voting aggregation.

Bagging vs. Boosting

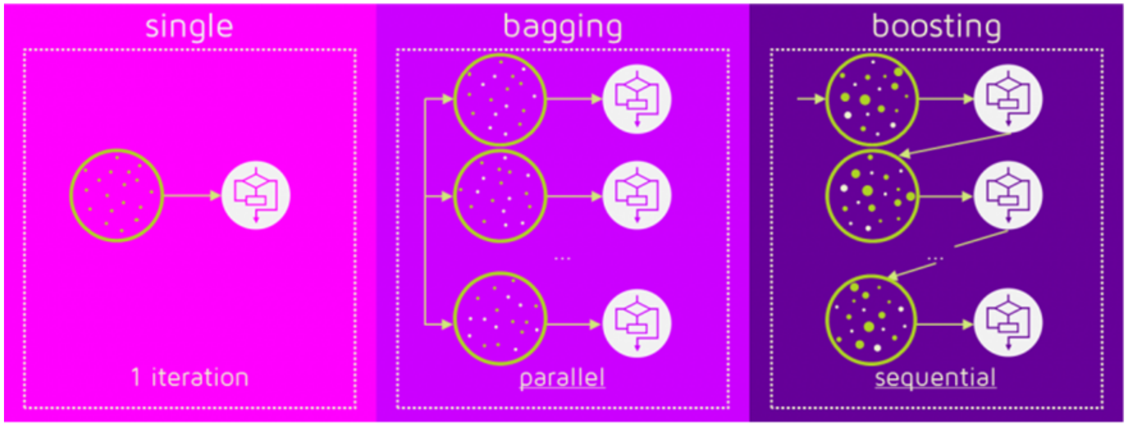


Stage-1: Pool of learners

Stage-2: SRSWR for unweighted (Bagging) and weighted (Boosting) training records

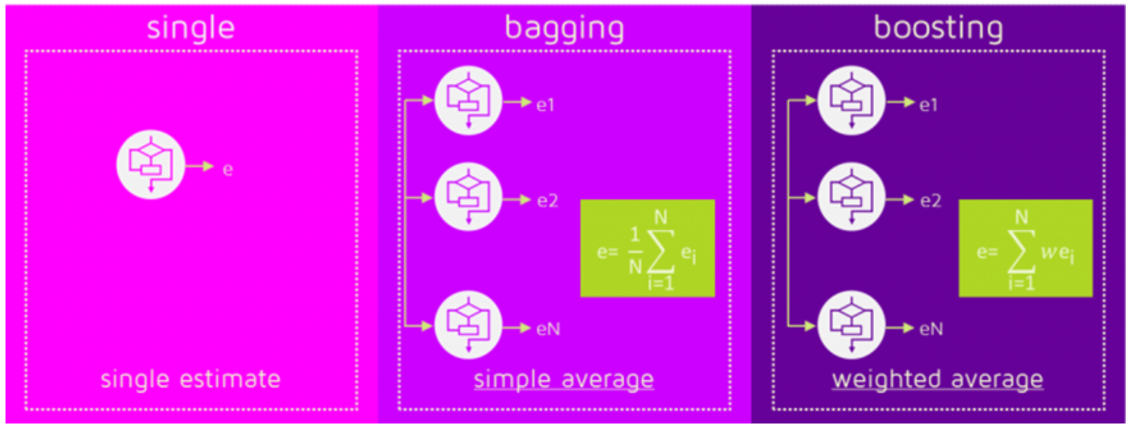


Bagging vs. Boosting



Stage-3: Parallel approach (Bagging) and Sequential approach (Boosting) of training/learning

Stage-4: Testing for classification of new record through simple average (Bagging) and weighted average (Boosting)



Bagging vs. Boosting

Stage-5: Updating training model (learner) through weights with less training error and good classification result for new record (Boosting); updation not applicable for Bagging.

Bagging vs. Boosting

Activity	Bagging	Boosting
Generation of several training data sets (by random sampling)	No weight for training data	determines weights for the data to tip the scales in favor of the most difficult cases.
Ensemble methods (get *N* learners from 1 learner)	built independently.	add new models that do well where previous models fail.
reducing variance and provide higher stability	Good (decrease the variance of single estimate by combining several estimates from different models)	Good (decrease the variance of single estimate by combining several estimates from different models)
Bias reduction	NA	Possible (generate a combined model with lower errors)
Solving overfitting problem	Possible	May increase
Final decision (taking the majority of N learners)	Good (equally weighted average on training data)	Good (more weight to those with better performance on training data)

Boosting - AdaBoost

AdaBoost is adaptive in nature because it assigns weights for base models (α) based on the accuracy of the model, and changes weights of the training records (*w*) based on the accuracy of the prediction. Consider the framework of the AdaBoost ensemble model with *m* base classifiers and *n* training records $((x_1,y_1), (x_2,y_2), \ldots, (x_n,y_n))$. The steps involved in AdaBoost are as follows:

1. Each training record is assigned an uniform weight $w_i = 1/n$.
2. Training records are sampled and the first base classifier $b_k(x)$ is built.
3. The error rate for the base classifier can be calculated as follows:

$$e_k = \sum_{k=1}^{n} w_i * I\left(b_k(x_i) \neq y_i\right)$$

where I(x) = 1 when the prediction is right, and 0 when the prediction is incorrect.

Boosting - AdaBoost

4. The weight of the classifier can be calculated as $\alpha_k = \ln (1- e_k)/e_k$. If the model has a low error rate, then the weight of the classifier is high and vice versa.

5. Next, the weights of all training records are updated by

$$w_{k+1}(i+1) = w_k(i) * e^{(\alpha_k F(bk(xi) \neq yi))}$$

where $F(x) = -1$ if the prediction is right and $F(x) = 1$ if the prediction is wrong.

Hence, the AdaBoost model updates the weights based on the prediction and the error rate of the base classifier. If the error rate is more than 50%, the record weight is not updated and reverted back to the next round.

Note: several alternatives exist with different boosting algorithms (AdaBoost, LPBoost, XGBoost, GradientBoost, BrownBoost) ways to determine the weights to use in the next training step and in the classification stage.

Ensemble Learner – Random Forest

Breiman, L. (2001). Random Forests. *Machine Learning, 45, 5–32*

Random Forest technique uses a similar concept to the one used in bagging. When deciding on splitting each node in a decision tree, the random forest only considers a random subset of all the attributes in the training set. To reduce the generalization error, the algorithm is randomized in two levels, training record selection and attribute selection, in the inner working of each base classifier.

Steps: If there are *n* training records with *m* attributes, and let *k* be the number of trees in the forest; then for each tree:

1. A random sample of size *n* is selected with replacement (similar to bagging).
2. Select a random number *D* (<<< *m*) which determines the number of attributes to be considered for node splitting.
3. A decision tree is started. For each node, Step-2 is repeated for every node and continued till all *k* tress are built.
4. As in any ensemble, the greater the diversity of the base trees, the lower the error of the ensemble.

Once all the trees in the forest are built, for every new record, all the trees predict a class and vote for the class with equal weights. The most predicted class by the base trees is the prediction of the forest.