MCSC201: Machine Learning

Dr. Ankit Rajpal
Department of Computer Science
University of Delhi

Ensemble Methods

- Aggregating the predictions of multiple classifiers.
- Also known as classifier combination methods.
- Set of base classifiers from training data and performs classification by taking a vote on the predictions made by each base classifier.

Example: Ensemble of 25 binary classifiers each of which has an error rate of $\in= 0.35$.

- If the base classifiers are identical, then the ensemble will misclassify the same examples predicted incorrectly by the base classifiers. Thus, the error rate of the ensemble remains 0.35.
- If the base classifiers are independent -i.e., their errors are uncorrelated-then the ensemble makes a wrong prediction only if more than half of the base classifiers predict incorrectly.

Rationale of Ensemble Methods

- Error rate of the ensemble classifier is:

$$e_{ensemble} = \sum_{i=13}^{25} 25_{C_i} \in^{i} (1 - \in)^{25-i} = 0.06$$

which is considerably lower than the error rate of the base classifiers.

- Necessary Conditions for an ensemble classifier to perform better than a single classifier :
  - Base classifiers should be independent of each other, and
  - Base classifiers should do better than a classifier that performs random guessing

Ensemble Methods

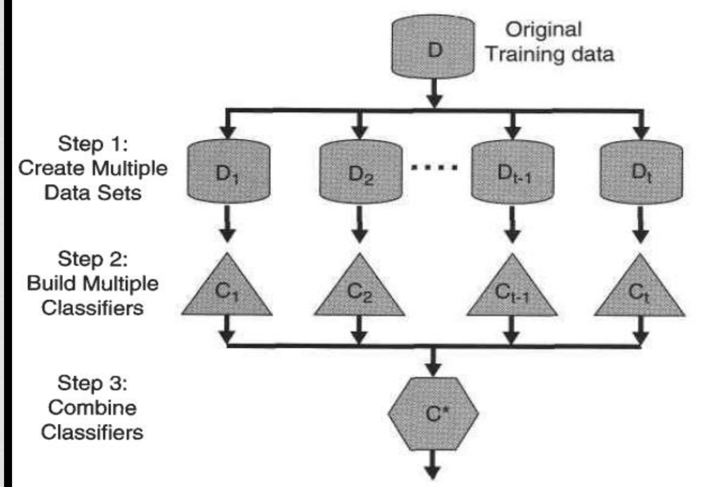


Figure 5.31. A logical view of the ensemble learning method.

Algorithm 5.5 General procedure for ensemble method.

Let D denote the original training data, k denote the number of base classifiers, and T be the test data.
for $i = 1$ to k **do**
    Create training set, D_i from D.
    Build a base classifier C_i from D_i.
end for
for each test record $x \in T$ **do**
    $C^*(x) = Vote(C_1(\mathbf{x}), C_2(\mathbf{x}), \ldots, C_k(\mathbf{x}))$
end for

Ensemble Methods

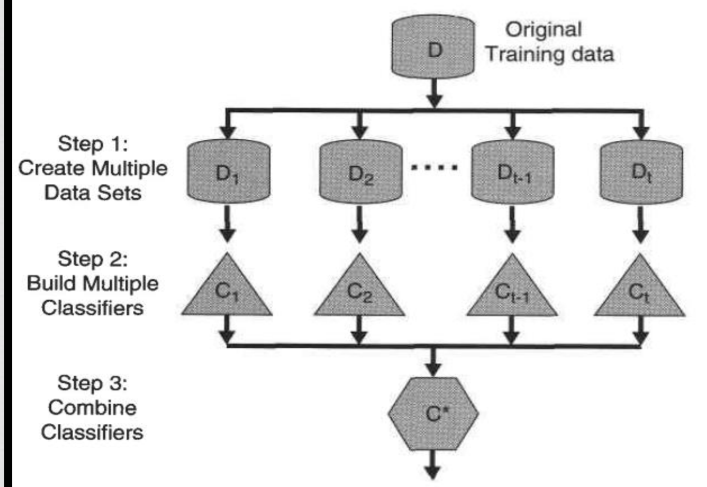


Figure 5.31. A logical view of the ensemble learning method.

- Unweighted Voting
- Weighted Voting

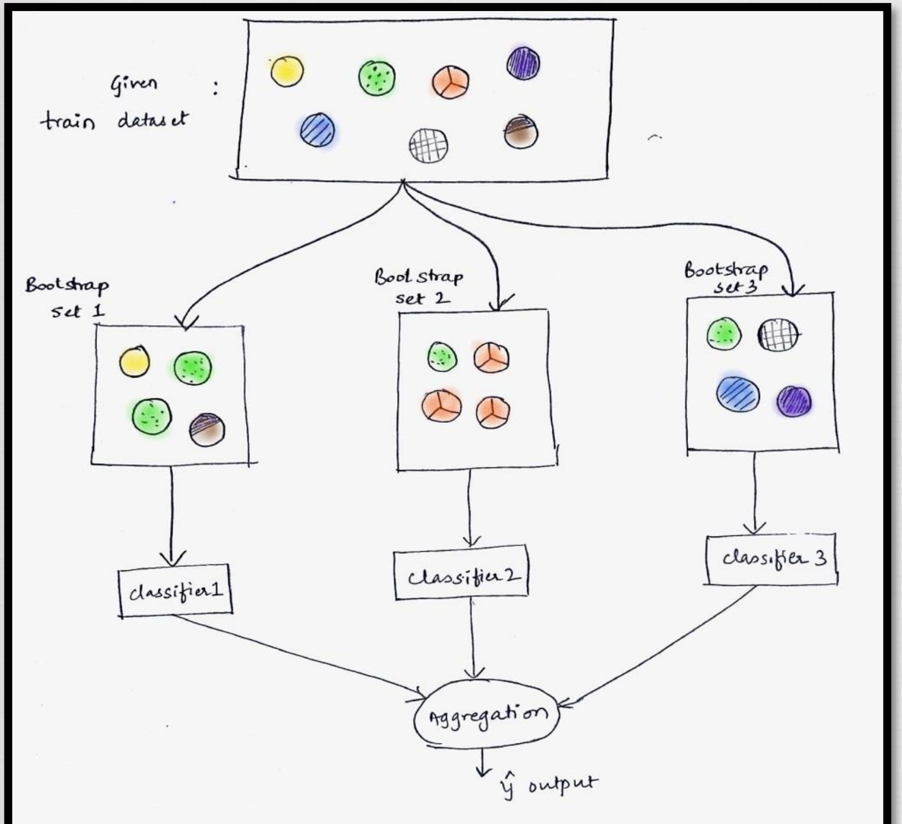
Algorithm 5.5 General procedure for ensemble method.

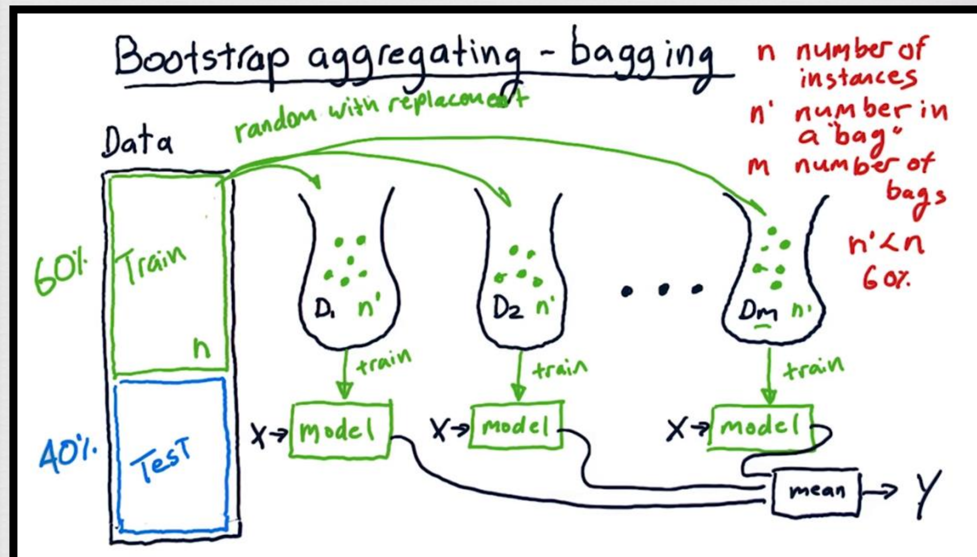
Let D denote the original training data, k denote the number of base classifiers, and T be the test data.
for $i = 1$ to k **do**
  Create training set, D_i from D.
  Build a base classifier C_i from D_i.
end for
for each test record $x \in T$ **do**
  $C^*(x) = Vote(C_1(\mathbf{x}), C_2(\mathbf{x}), \ldots, C_k(\mathbf{x}))$
end for

Bagging

- Also known as bootstrap aggregating
  - Repeatedly samples (with replacement) from a dataset according to a uniform probability distribution.
- For i = 1 .. M
  - Draw $n^*<n$ samples from D with replacement
  - Learn classifier C_i
- Final classifier is a vote of C_1 .. C_M

Bagging


Given train dataset :
Bootstrap set 1
Bootstrap set 2
Bootstrap set 3
classifier1
classifier2
classifier 3
Aggregation
ŷ output


Bootstrap aggregating - bagging
random with replacement
Data
60%
Train
n
40%
Test
D1 n'
D2 n'
Dm n'
train
train
train
X→ model
X→ model
X→ model
mean → Y
n number of instances
n' number in a "bag"
m number of bags
n'<n
60%

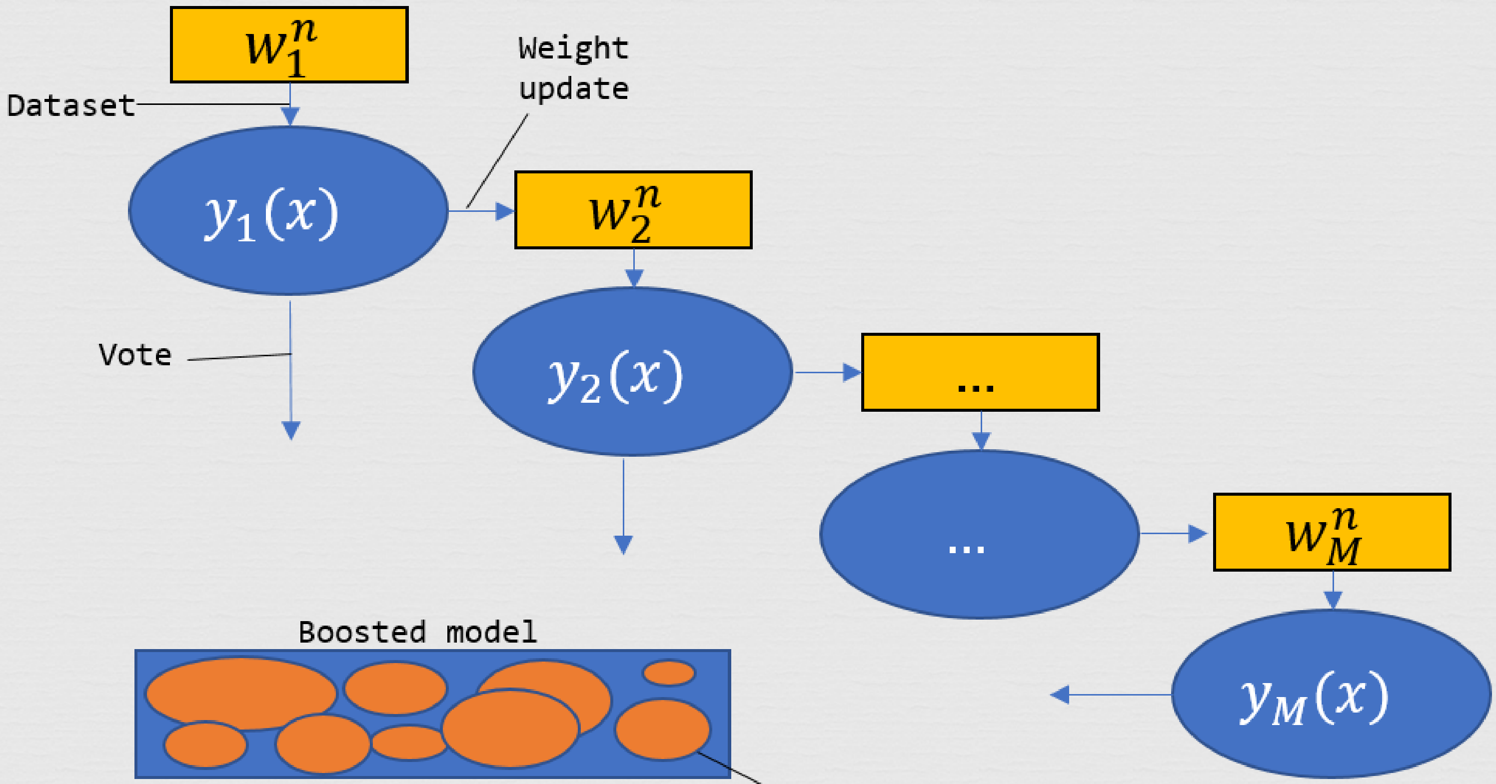
Random Forest Classifier

- A random forest is a meta estimator that fits a number of decision tree classifiers on various sub samples of the dataset and uses averaging to improve the predictive accuracy and control over fitting.
- The sub-sample size is controlled with the max_samples parameter if bootstrap = True (default), otherwise the whole dataset is used to build each tree.

Boosting

- Incremental
- Iterative Procedure used to adaptively change the distribution of training examples so that the base classifiers will focus on examples that are hard to classify.
- Build new models that try to do better on previous model's misclassifications
  - Can get better accuracy

Boosting



$$Y_M(x) = sign(\sum_m^M \alpha_m y_m(x))$$

Adaboost

- An AdaBoost [1] classifier is a meta-estimator that begins by fitting a classifier on the original dataset and then fits additional copies of the classifier on the same dataset but where the weights of incorrectly classified instances are adjusted such that subsequent classifiers focus more on difficult cases.

[1] Y. Freund, R. Schapire, “A Decision-Theoretic Generalization of on-Line Learning and an Application to Boosting”, 1995.