Introducción al Aprendizaje Automático

Ferdinand Pineda

Machine Learning
Ingeniería de Sistemas

November 9, 2021

Overview

Ensemble Methods

Ensemble methods are techniques that create multiple models and then combine them to produce improved results.

Ensemble methods usually produces more accurate solutions than a single model would. Methods of ensemble: voting, stacking, bagging and boosting.

Ensemble Methods

Ensemble methods are techniques that create multiple models and then combine them to produce improved results.

Ensemble methods usually produces more accurate solutions than a single model would. Methods of ensemble: voting, stacking, bagging and boosting.

Logistic regression models the probabilities for classification problems with two possible outcomes. It's an extension of the linear regression model for classification problems.

Voting and averaging are two of the easiest ensemble methods. Voting is used for classification and averaging is used for regression.

In both methods, the first step is to create multiple classification/regression models using some training dataset. Each base model can be created using different splits of the same training dataset and same algorithm, or using the same dataset with different algorithms, or any other method.

Voting and Averaging Based Ensemble Methods

Logistic regression models the probabilities for classification problems with two possible outcomes. It's an extension of the linear regression model for classification problems.

Voting and averaging are two of the easiest ensemble methods. Voting is used for classification and averaging is used for regression.

In both methods, the first step is to create multiple classification/regression models using some training dataset. Each base model can be created using different splits of the same training dataset and same algorithm, or using the same dataset with different algorithms, or any other method.

Voting and Averaging Based Ensemble Methods

Logistic regression models the probabilities for classification problems with two possible outcomes. It's an extension of the linear regression model for classification problems.

Voting and averaging are two of the easiest ensemble methods. Voting is used for classification and averaging is used for regression.

In both methods, the first step is to create multiple classification/regression models using some training dataset. Each base model can be created using different splits of the same training dataset and same algorithm, or using the same dataset with different algorithms, or any other method.

Voting and Averaging Based Ensemble Methods

- Majority Voting
  Every model makes a prediction (votes) for each test instance and the final output prediction is the one that receives more than half of the votes. If none of the predictions get more than half of the votes, we may say that the ensemble method could not make a stable prediction for this instance.

Voting and Averaging Based Ensemble Methods

- Majority Voting

Colleague 1	Colleague 2	Colleague 3	Colleague 4	Colleague 5	Final rating
5	4	5	4	4	4

Figure: Majority Voting

Voting and Averaging Based Ensemble Methods

- Weighted Voting Unlike majority voting, where each model has the same rights, we can increase the importance of one or more models. In weighted voting you count the prediction of the better models multiple times. Finding a reasonable set of weights is up to you.

Voting and Averaging Based Ensemble Methods

- Simple Averaging In simple averaging method, for every instance of test dataset, the average predictions are calculated. This method often reduces overfit.

- Simple Averaging

For example, in the below case, the averaging method would take the average of all the values.

i.e. (5+4+5+4+4)/5 = 4.4

Colleague 1	Colleague 2	Colleague 3	Colleague 4	Colleague 5	Final rating
5	4	5	4	4	4.4

Figure: Simple Averaging

Voting and Averaging Based Ensemble Methods

- Weighted Averaging Weighted averaging is a slightly modified version of simple averaging, where the prediction of each model is multiplied by the weight and then their average is calculated.

- Weighted Averaging

The result is calculated as [(5*0.23) + (4*0.23) + (5*0.18) + (4*0.18) + (4*0.18)] = 4.41.

	Colleague 1	Colleague 2	Colleague 3	Colleague 4	Colleague 5	Final rating
weight	0.23	0.23	0.18	0.18	0.18	
rating	5	4	5	4	4	4.41

Figure: Weighted Averaging

Stacking Multiple Machine Learning Models

Stacking, also known as stacked generalization, is an ensemble method where the models are combined using another machine learning algorithm. The basic idea is to train machine learning algorithms with training dataset and then generate a new dataset with these models. Then this new dataset is used as input for the combiner machine learning algorithm.

Stacking Multiple Machine Learning Models

The training dataset for combiner algorithm is generated using the outputs of the base algorithms. The base algorithm is generated using training dataset and then the same dataset is used again to make predictions. But as we know, in the real world we do not use the same training dataset for prediction, so to overcome this problem you may see some implementations of stacking where training dataset is splitted.

Stacking Multiple Machine Learning Models

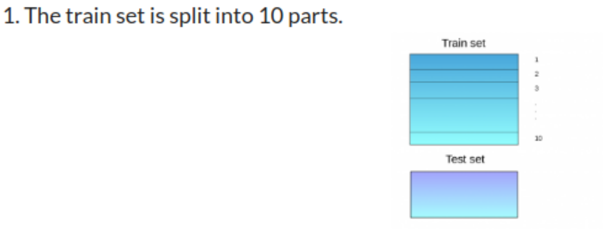

Figure: Stacking

Stacking Multiple Machine Learning Models

2. A base model (suppose a decision tree) is fitted on 9 parts and predictions are made for the 10th part. This is done for each part of the train set.

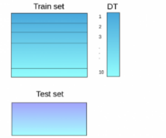

Figure: Stacking

3. The base model (in this case, decision tree) is then fitted on the whole train dataset.

4. Using this model, predictions are made on the test set.

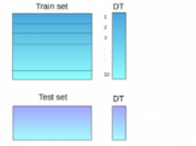

Figure: Stacking

5. Steps 2 to 4 are repeated for another base model (say knn) resulting in another set of predictions for the train set and test set.

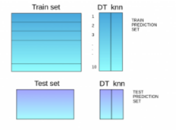

Figure: Stacking

Stacking Multiple Machine Learning Models

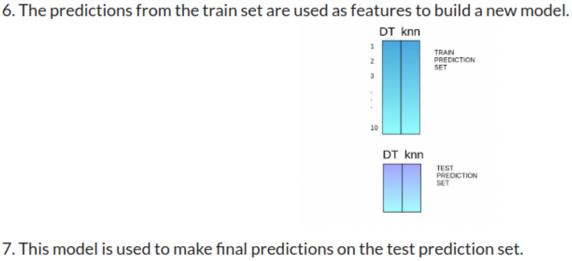

Figure: Stacking

Bootstrap Aggregating

The name Bootstrap Aggregating, also known as “Bagging”, summarizes the key elements of this strategy.

In the bagging algorithm, the first step involves creating multiple models. These models are generated using the same algorithm with random sub-samples of the dataset which are drawn from the original dataset randomly with bootstrap sampling method.

In bootstrap sampling, some original examples appear more than once and some original examples are not present in the sample. If you want to create a sub-dataset with m elements, you should select a random element from the original dataset m times. And if the goal is generating n dataset, you follow this step n times.

Bootstrap Aggregating

The name Bootstrap Aggregating, also known as “Bagging”, summarizes the key elements of this strategy.

In the bagging algorithm, the first step involves creating multiple models. These models are generated using the same algorithm with random sub-samples of the dataset which are drawn from the original dataset randomly with bootstrap sampling method.

In bootstrap sampling, some original examples appear more than once and some original examples are not present in the sample. If you want to create a sub-dataset with m elements, you should select a random element from the original dataset m times. And if the goal is generating n dataset, you follow this step n times.

Bootstrap Aggregating

The name Bootstrap Aggregating, also known as “Bagging”, summarizes the key elements of this strategy.

In the bagging algorithm, the first step involves creating multiple models. These models are generated using the same algorithm with random sub-samples of the dataset which are drawn from the original dataset randomly with bootstrap sampling method.

In bootstrap sampling, some original examples appear more than once and some original examples are not present in the sample. If you want to create a sub-dataset with m elements, you should select a random element from the original dataset m times. And if the goal is generating n dataset, you follow this step n times.

Bootstrap Aggregating

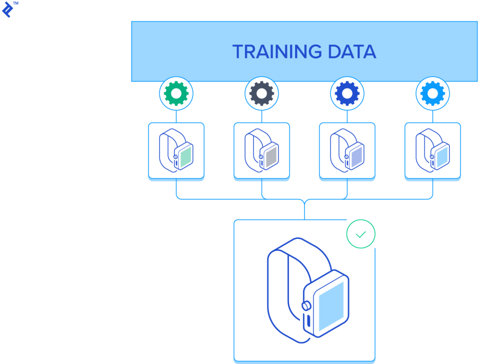

Figure: Bootstrap Aggregating

The second step in bagging is aggregating the generated models. Well known methods, such as voting and averaging, are used for this purpose.

In bagging, each sub-samples can be generated independently from each other. So generation and training can be done in parallel.

Bootstrap Aggregating

The second step in bagging is aggregating the generated models. Well known methods, such as voting and averaging, are used for this purpose.

In bagging, each sub-samples can be generated independently from each other. So generation and training can be done in parallel.

Bootstrap Aggregating

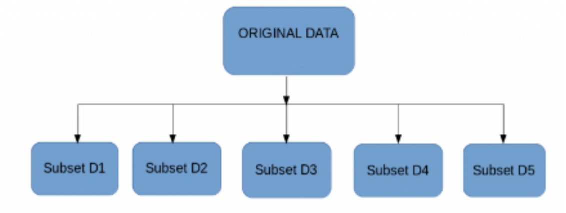

Figure: Bagging

Bootstrap Aggregating

1. Multiple subsets are created from the original dataset, selecting observations with replacement.
2. A base model (weak model) is created on each of these subsets.
3. The models run in parallel and are independent of each other.
4. The final predictions are determined by combining the predictions from all the models.

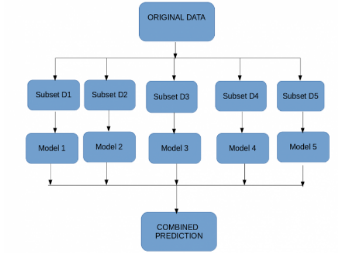

Figure: Bagging

Boosting: Converting Weak Models to Strong Ones

The term “boosting” is used to describe a family of algorithms which are able to convert weak models to strong models.

The model is weak if it has a substantial error rate, but the performance is not random (resulting in an error rate of 0.5 for binary classification).

Boosting incrementally builds an ensemble by training each model with the same dataset but where the weights of instances are adjusted according to the error of the last prediction. The main idea is forcing the models to focus on the instances which are hard. Unlike bagging, boosting is a sequential method, and so you can not use parallel operations here.

Boosting: Converting Weak Models to Strong Ones

The term “boosting” is used to describe a family of algorithms which are able to convert weak models to strong models.

The model is weak if it has a substantial error rate, but the performance is not random (resulting in an error rate of 0.5 for binary classification).

Boosting incrementally builds an ensemble by training each model with the same dataset but where the weights of instances are adjusted according to the error of the last prediction. The main idea is forcing the models to focus on the instances which are hard. Unlike bagging, boosting is a sequential method, and so you can not use parallel operations here.

Boosting: Converting Weak Models to Strong Ones

The term "boosting" is used to describe a family of algorithms which are able to convert weak models to strong models.

The model is weak if it has a substantial error rate, but the performance is not random (resulting in an error rate of 0.5 for binary classification).

Boosting incrementally builds an ensemble by training each model with the same dataset but where the weights of instances are adjusted according to the error of the last prediction. The main idea is forcing the models to focus on the instances which are hard. Unlike bagging, boosting is a sequential method, and so you can not use parallel operations here.

Boosting: Converting Weak Models to Strong Ones

Adaboost is a widely known algorithm which is a boosting method. The founders of Adaboost won the Gödel Prize for their work. Mostly, decision tree algorithm is preferred as a base algorithm for Adaboost and in sklearn library the default base algorithm for Adaboost is decision tree.

The model is weak if it has a substantial error rate, but the performance is not random (resulting in an error rate of 0.5 for binary classification).

Boosting incrementally builds an ensemble by training each model with the same dataset but where the weights of instances are adjusted according to the error of the last prediction. The main idea is forcing the models to focus on the instances which are hard. Unlike bagging, boosting is a sequential method, and so you can not use parallel operations here.

Boosting: Converting Weak Models to Strong Ones

Adaboost is a widely known algorithm which is a boosting method. The founders of Adaboost won the Gödel Prize for their work. Mostly, decision tree algorithm is preferred as a base algorithm for Adaboost and in sklearn library the default base algorithm for Adaboost is decision tree.

The model is weak if it has a substantial error rate, but the performance is not random (resulting in an error rate of 0.5 for binary classification).

Boosting incrementally builds an ensemble by training each model with the same dataset but where the weights of instances are adjusted according to the error of the last prediction. The main idea is forcing the models to focus on the instances which are hard. Unlike bagging, boosting is a sequential method, and so you can not use parallel operations here.

Boosting: Converting Weak Models to Strong Ones

Adaboost is a widely known algorithm which is a boosting method. The founders of Adaboost won the Gödel Prize for their work. Mostly, decision tree algorithm is preferred as a base algorithm for Adaboost and in sklearn library the default base algorithm for Adaboost is decision tree.

The model is weak if it has a substantial error rate, but the performance is not random (resulting in an error rate of 0.5 for binary classification).

Boosting incrementally builds an ensemble by training each model with the same dataset but where the weights of instances are adjusted according to the error of the last prediction. The main idea is forcing the models to focus on the instances which are hard. Unlike bagging, boosting is a sequential method, and so you can not use parallel operations here.

Boosting: Converting Weak Models to Strong Ones

1. A subset is created from the original dataset.
2. Initially, all data points are given equal weights.
3. A base model is created on this subset.
4. This model is used to make predictions on the whole dataset.

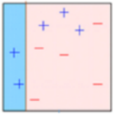

Figure: Boosting

Boosting: Converting Weak Models to Strong Ones

5. Errors are calculated using the actual values and predicted values.
6. The observations which are incorrectly predicted, are given higher weights.
   (Here, the three misclassified blue-plus points will be given higher weights)
7. Another model is created and predictions are made on the dataset.
   (This model tries to correct the errors from the previous model)

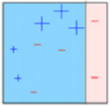

Figure: Boosting

Boosting: Converting Weak Models to Strong Ones

8. Similarly, multiple models are created, each correcting the errors of the previous model.
9. The final model (strong learner) is the weighted mean of all the models (weak learners).

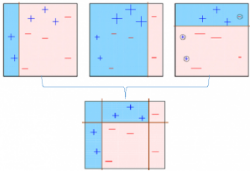

Figure: Boosting

Introducción al Aprendizaje Automático

Ferdinand Pineda

Machine Learning
Ingeniería de Sistemas

November 9, 2021