Machine Learning

Model Improvement (Ensemble Learning)

Satishkumar L. Varma

Department of Information Technology
SVKM's Dwarkadas J. Sanghvi College of Engineering, Vile Parle, Mumbai.
ORCID | Scopus | Google Scholar | Google Site | Website

Ensemble Learning: Outline

- Model Improvement
- Ensemble Learning
  - Ensemble methods
  - Bagging to avoid overfitting
  - Boosting to avoid underfitting
  - Stacking to avoid underfitting

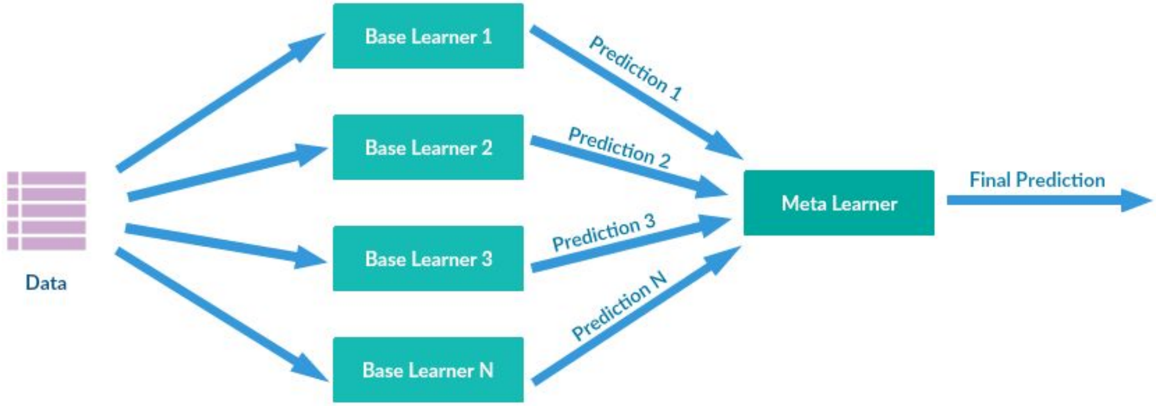
Ensemble Learning: Why Ensemble Learning

- Need of Model Improvement (Ensemble Learning)
  - ML models are nonlinear and have a high variance,
    - which can be difficult when preparing a final model for making predictions.
  - Ensemble learning combines the predictions from multiple neural network models
    - to reduce the variance of predictions and reduce generalization error.
  - Techniques for ensemble learning can be grouped by the element that is varied,
    - such as training data, the model, and how predictions are combined.
- Objectives
  - To introduce the concept of ensemble learning and understand the algorithms which use this technique.
  - To know ensemble learning types, methods – Bagging, Boosting, Stacking and Blending.
  - To Distinguish between bagging, boosting and stacking.
  - To explain process of ensemble learning techniques; as well as their advantages and disadvantages.
  - To understand how each technique works and which technique to use and when.
  - To choose the best Ensemble learning method for improving your model's accuracy.
  - To understand algorithms in Python using a hands-on case study on a real-life problem.

Ensemble Learning: Introduction

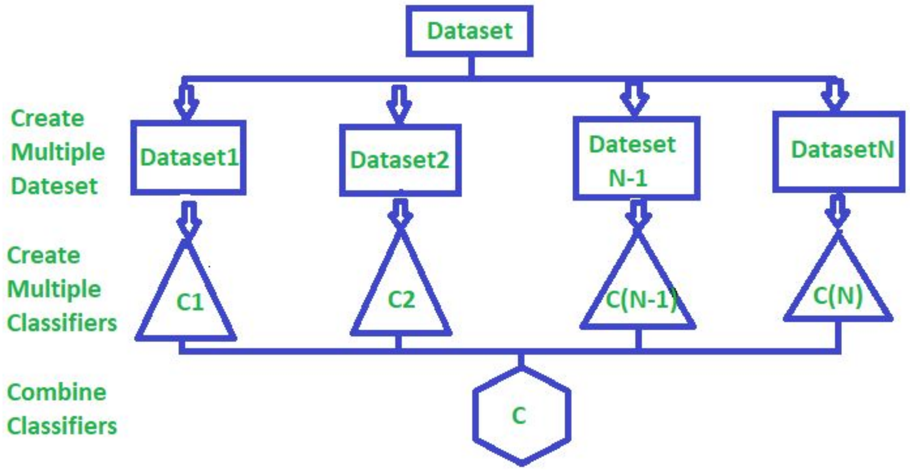
- Ensemble Learning
- Review and compare different mobile models, checking for their features and prices.
- Ensemble models combine the decisions from multiple models to improve the overall performance.

  Ensembling is the process of combining multiple learning algorithms
  - to obtain their collective performance i.e.,
  - to improve the performance of existing models
  - Models are stacked together to improve their performance and get one final prediction (see figure).



Ensemble Learning: Introduction

- Ensemble learning helps improve machine learning results by combining several models.
- This approach allows the production of better predictive performance compared to a single model.
- Basic idea is to learn a set of classifiers (experts) and to allow them to vote.
- Advantage : Improvement in predictive accuracy.
- Disadvantage : It is difficult to understand an ensemble of classifiers.



Ensemble Learning: Ensemble Learning Types

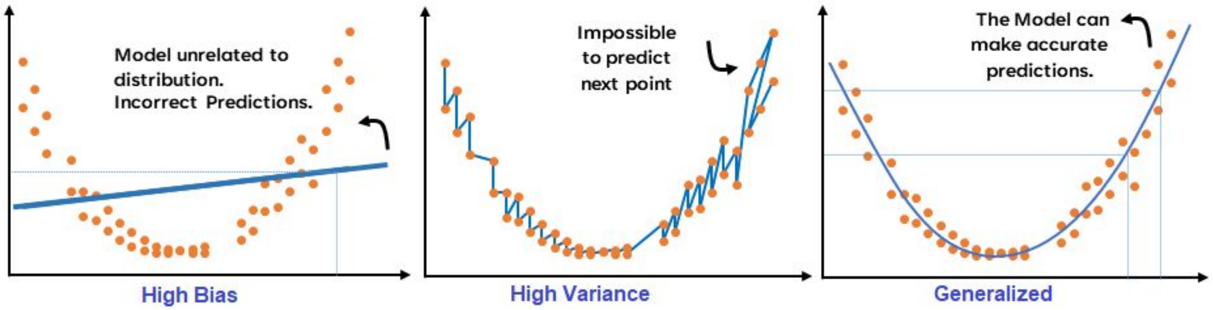
- Ensemble Learning Types: Categorized into two groups
- 1. Sequential Ensemble Methods
    - In this method base learners are dependent on the results from previous base learners.
    - Every subsequent base model corrects the prediction made by its predecessor fixing the errors in it.
    - Hence the overall performance can be increased via improving the weight of previous labels.
- 2. Parallel Ensemble Methods
    - In this method there is no dependency between the base learners and
    - all base learners execute in parallel and
    - the results of all base models are combined in the end (using averaging for regression and voting for classification problems).
    - Parallel Ensemble methods are divided in two categories-
        - 1. Homogeneous Parallel Ensemble Methods
            - In this method a single machine learning algorithm is used as a base learner.
        - 2. Heterogeneous Parallel Ensemble Methods
            - In this method multiple machine learning algorithms are used as base learners.
- Advanced Ensemble Techniques: Bagging, Boosting and Stacking

Ensemble Learning: Introduction

- How Ensemble Learning Works?
- Ensemble learning is a learning method that consists of combining ML/DL learning models.
- A problem in ML/ML is that individual models tend to perform poorly.
- In other words, they tend to have low prediction accuracy.
- To mitigate this problem, we combine multiple models to get one with a better performance.
- The individual models that we combine are known as weak learners.
- We call them weak learners because they either have a high bias or high variance.
- Because they either have high bias or variance, weak learners cannot learn efficiently and perform poorly.
- A high-bias model results from not learning data well enough.
- It is not related to the distribution of the data.
- Hence future predictions will be unrelated to the data and thus incorrect.

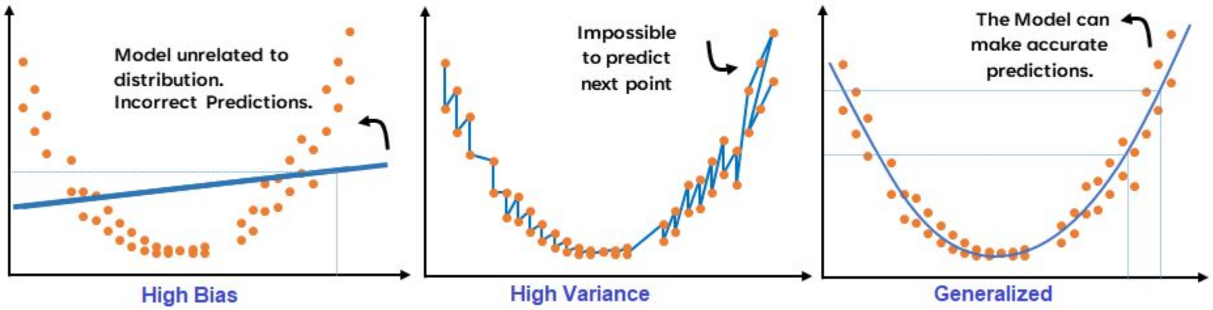
Ensemble Learning: Introduction

- Bias:
  - A high-bias model results from not learning data well enough.
  - It is not related to the distribution of the data.
  - Hence future predictions will be unrelated to the data and thus incorrect.
- Variance:
  - A high variance model results from learning the data too well.
  - It varies with each data point.
  - Hence it is impossible to predict the next point accurately.
- Both high bias and high variance models thus cannot generalize properly.
- Thus, weak learners will either make incorrect generalizations or fail to generalize altogether.
- Because of this, the predictions of weak learners cannot be relied on by themselves.



Ensemble Learning: Introduction

- Bias-variance trade-off,
  - an underfit model has high bias and low variance,
  - whereas an overfit model has high variance and low bias.
- In either case, there is no balance between bias and variance.
- For there to be a balance, both the bias and variance need to be low.
- Ensemble learning tries to balance this bias-variance trade-off by reducing either the bias or the variance.
- Ensemble learning will aim to reduce the bias if we have a weak model with high bias and low variance.
- Ensemble learning will aim to reduce the variance if we have a weak model with high variance and low bias.
- This way, the resulting model will be much more balanced, with low bias and variance.
- Thus, the resulting model will be known as a strong learner.
- This model will be more generalized than the weak learners.
- It will thus be able to make accurate predictions.



Ensemble Learning: Bagging and Boosting

- Bagging to avoid overfitting, Boosting to avoid underfitting
- Bagging attempts to reduce the chance of overfitting complex models.
  - It trains a large number of “strong” learners in parallel.
  - A strong learner is a model that’s relatively unconstrained.
  - Bagging then combines all the strong learners together in order to “smooth out” their predictions.
- Boosting attempts to improve the predictive flexibility of simple models.
  - It trains a large number of “weak” learners in sequence.
  - A weak learner is a constrained model (i.e. you could limit the max depth of each decision tree).
  - Each one in the sequence focuses on learning from the mistakes of the one before it.
  - Boosting then combines all the weak learners into a single strong learner.

Ensemble Learning: Introduction

- Ensemble learning improves a model's performance in mainly three ways:
    - By reducing the variance of weak learners
    - By reducing the bias of weak learners,
    - By improving the overall accuracy of strong learners.
- Bagging is used to reduce the variance of weak learners.
- Boosting is used to reduce the bias of weak learners.
- Stacking is used to improve the overall accuracy of strong learners.
- **1. Reducing Variance with Bagging**
- We use bagging for combining weak learners of high variance.
- Bagging aims to produce a model with lower variance than the individual weak models.
- These weak learners are homogenous, meaning they are of the same type.
- Bagging is also known as Bootstrap aggregating.
- It consists of two steps:
    - Bootstrapping and
    - Aggregation

Ensemble Learning: Introduction

- **Bootstrapping**
  - Involves resampling subsets of data with replacement from an initial dataset.
  - In other words, subsets of data are taken from the initial dataset.
  - These subsets of data are called bootstrapped datasets or, simply, bootstraps. Resampled 'with replacement' means an individual data point can be sampled multiple times.
  - Each bootstrap dataset is used to train a weak learner.
- **Aggregating**
  - Individual weak learners are trained independently from each other.
  - Each learner makes independent predictions.
  - The results of those predictions are aggregated at the end to get the overall prediction.
  - The predictions are aggregated using either **max voting or averaging**.

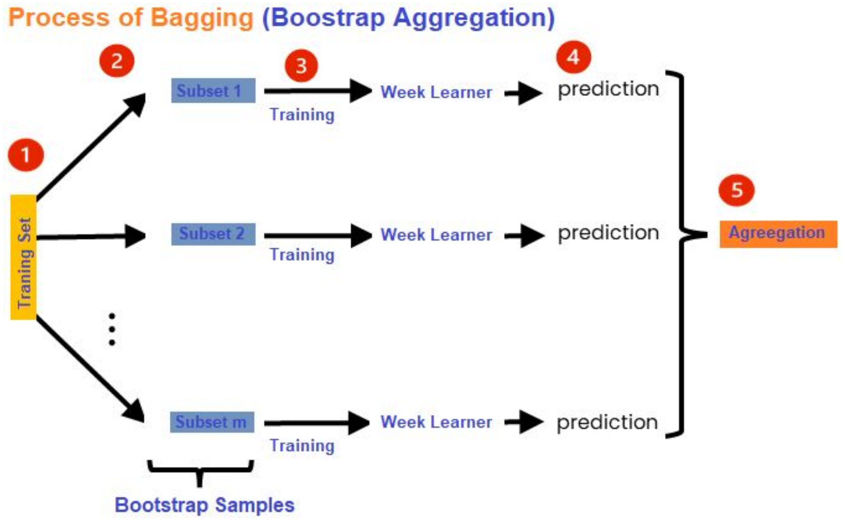
Ensemble Learning: Introduction

- **Max Voting**
  - It is commonly used for classification problems.
  - It consists of taking the mode of the predictions (the most occurring prediction).
  - It is called voting because like in election voting, the premise is that 'the majority rules'.
  - Each model makes a prediction.
  - A prediction from each model counts as a single 'vote'.
  - The most occurring 'vote' is chosen as the representative for the combined model.
- **Averaging**:
  - It is generally used for regression problems.
  - It involves taking the average of the predictions.
  - The resulting average is used as the overall prediction for the combined model.

Ensemble Learning: Introduction

- **Steps of Bagging**
- 1. Initiate training dataset containing n-number of instances.
- 2. Create a m-number of subsets of data from the training set.
- 2.1 We take a subset of N sample points from the initial dataset for each subset.
- 2.2 Each subset is taken with replacement. It means that a specific data point can be sampled more than once.
- 3. For each subset of data, we train the corresponding weak learners independently.
- 3.1 These models are homogeneous, meaning that they are of the same type.
- 4. Let each model makes a prediction.
- 4. The predictions are aggregated into a single prediction.
- 4.1 For this, use either max voting or averaging

  - m – number of subsets
  - n – number of instances in initial dataset
  - N – number of sample points in a subset, where n > N

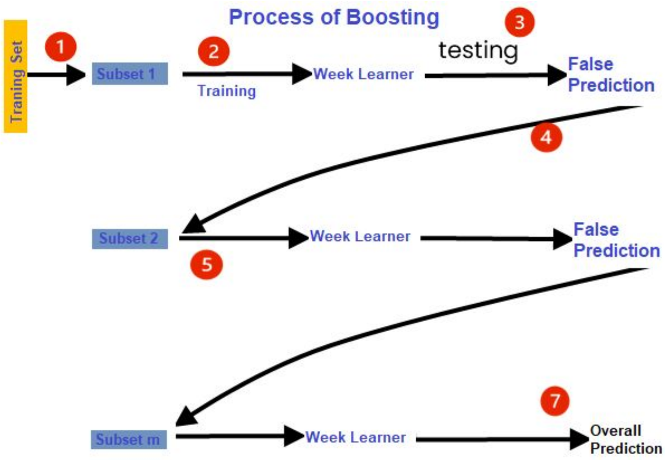


Ensemble Learning: Introduction

- Reducing Bias by Boosting
- We use boosting for combining weak learners with high bias.
- Boosting aims to produce a model with a lower bias than that of the individual models.
- Like in bagging, the weak learners are homogeneous.
- **Boosting**: Boosting involves sequentially training weak learners.
  - Here, each subsequent learner improves the errors of previous learners in the sequence.
  - A sample of data is first taken from the initial dataset.
  - This sample is used to train the first model, and the model makes its prediction.
  - The samples can either be correctly or incorrectly predicted.
  - The samples that are wrongly predicted are reused for training the next model.
  - In this way, subsequent models can improve on the errors of previous models.
- **Aggregates**: Aggregates prediction results at the end, boosting aggregates the results at each step.
  - They are aggregated using weighted averaging.
  - Weighted averaging involves giving all models different weights depending on their predictive power.
  - In other words, it gives more weight to the model with the highest predictive power.
  - This is because the learner with the highest predictive power is considered the most important.

Ensemble Learning: Steps of Boosting

- 1. We sample m-number of subsets from an initial training dataset.
- 2. Using the first subset, we train the first weak learner.
- 3. We test the trained weak learner using the training data. As a result of the testing, some data points will be incorrectly predicted.
- 4. Each data point with the wrong prediction is sent into the second subset of data, and this subset is updated.
- 5. Using this updated subset, we train and test the second weak learner.
- 6. We continue with the following subset until the total number of subsets is reached.
- 7. We now have the total prediction. The overall prediction has already been aggregated at each step, so there is no need to calculate it.

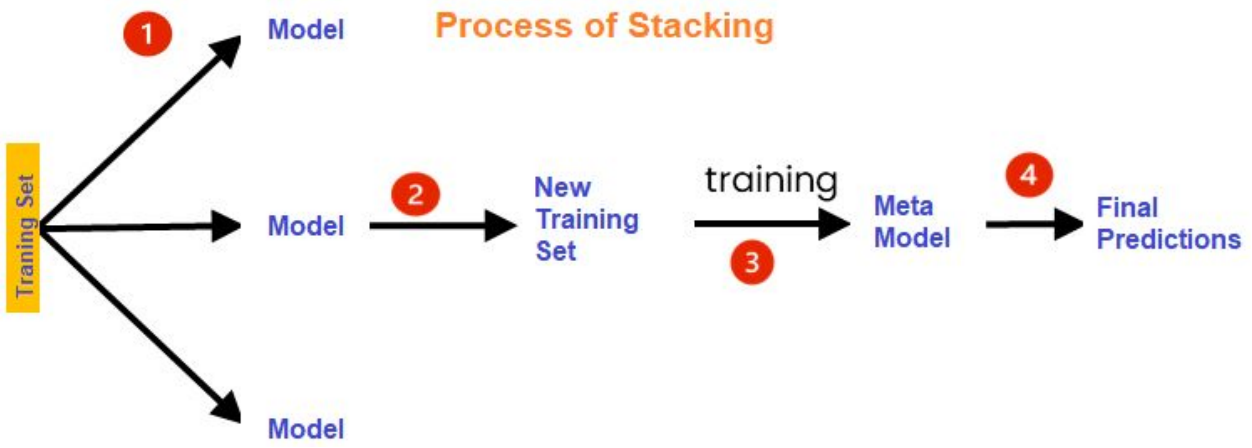


Ensemble Learning: Stacking

- **Improving Model Accuracy with Stacking**
- We use stacking to improve the prediction accuracy of strong learners. Stacking aims to create a single robust model from multiple heterogeneous strong learners.
- Stacking differs from bagging and boosting in that:
    - It combines strong learners
    - It combines heterogeneous models
    - It consists of creating a Metamodel. A metamodel is a model created using a new dataset.
- Individual heterogeneous models are trained using an initial dataset.
- These models make predictions and form a single new dataset using those predictions.
- This new data set is used to train the metamodel, which makes the final prediction.
- The prediction is combined using weighted averaging.
- Because stacking combines strong learners, it can combine bagged or boosted models.

Ensemble Learning: Stacking

- **Steps of Stacking**
- 1. We use initial training data to train m-number of algorithms.
- 2. Using the output of each algorithm, we create a new training set.
- 3. Using the new training set, we create a meta-model algorithm.
- 4. Using the results of the meta-model, we make the final prediction. The results are combined using weighted averaging.



Ensemble Learning: Bagging vs Boosting vs Stacking

- When to use Bagging vs Boosting vs Stacking?
- If you want to reduce the overfitting or variance of your model, you use bagging.
- If you are looking to reduce underfitting or bias, you use boosting.
- If you want to increase predictive accuracy, use stacking.
- Bagging and boosting both works with homogeneous weak learners.
- Stacking works using heterogeneous solid learners.
- All three of these methods can work with either classification or regression problems.
- One disadvantage of boosting is that it is prone to variance or overfitting.
- It is thus not advisable to use boosting for reducing variance.
- Boosting will do a worse job in reducing variance as compared to bagging.

	Bagging	Boosting	Stacking
Purpose	Reduce Variance	Reduce Bias	Improve Accuracy
Base Learner Types	Homogeneous	Homogeneous	Heterogeneous
Base Learner Training	Parallel	Sequential	Meta Model
Aggregation	Max Voting, Averaging	Weighted Averaging	Weighted Averaging

Ensemble Learning: Bagging vs Boosting vs Stacking

- When to use Bagging vs Boosting vs Stacking?
- On the other hand, the converse is true.
- It is not advisable to use bagging to reduce bias or underfitting.
- This is because bagging is more prone to bias and does not help reduce bias.
- Stacked models have the advantage of better prediction accuracy than bagging or boosting.
- But because they combine bagged or boosted models, they have the disadvantage of needing much more time and computational power.
- If you are looking for faster results, it's advisable not to use stacking.
- However, stacking is the way to go if you're looking for high accuracy.

	Bagging	Boosting	Stacking
Purpose	Reduce Variance	Reduce Bias	Improve Accuracy
Base Learner Types	Homogeneous	Homogeneous	Heterogeneous
Base Learner Training	Parallel	Sequential	Meta Model
Aggregation	Max Voting, Averaging	Weighted Averaging	Weighted Averaging

Ensemble Learning: Conclusion

- Ensemble learning combines multiple ML models into a single model to increase the performance.
- Bagging, boosting and stacking are important for ensuring the accuracy of models.
- Bagging Technique
  - Developed to overcome instability in decision trees.
  - Example of the bagging technique: Random forest algorithm
  - The random forest is an ensemble of multiple decision trees.
  - Decision trees tend to be prone to overfitting.
  - Because of this, a single decision tree can’t be relied on for making predictions.
  - Prediction accuracy of DTs is improved by employing bagging to form a random forest.
  - The resulting random forest has a lower variance compared to the individual trees.
- Bagging aims to decrease variance; and Boosting aims to decrease bias;
- Stacking aims to improve prediction accuracy.
- Bagging + Boosting combine homogenous weak learners.
- Stacking combines heterogeneous solid learners.
- Bagging trains models in parallel;
- Boosting trains the models sequentially.
- Stacking creates a meta-model.

Model Optimization

- Refer slide
  - vsat2k_ML_Ch1d Model Optimization [PDF]

References

Text books:

1. Ethem Alpaydin, “Introduction to Machine Learning , 4th Edition, The MIT Press, 2020.
2. Peter Harrington, “Machine Learning in Action”, 1st Edition, Dreamtech Press, 2012."
3. Tom Mitchell, “Machine Learning”, 1st Edition, McGraw Hill, 2017.
4. Andreas C, Müller and Sarah Guido, “Introduction to Machine Learning with Python: A Guide for Data Scientists”, 1ed, O'reilly, 2016.
5. Kevin P. Murphy, “Machine Learning: A Probabilistic Perspective”, 1st Edition, MIT Press, 2012."

Reference Books:

6. Aurélien Géron, “Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow”, 2nd Edition, Shroff/O'Reilly, 2019.
7. Witten Ian H., Eibe Frank, Mark A. Hall, and Christopher J. Pal., “Data Mining: Practical machine learning tools and techniques”, 1st Edition, Morgan Kaufmann, 2016.
8. Han, Kamber, “Data Mining Concepts and Techniques”, 3rd Edition, Morgan Kaufmann, 2012.
9. Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar, “Foundations of Machine Learning”, 1ed, MIT Press, 2012.
10. H. Dunham, “Data Mining: Introductory and Advanced Topics”, 1st Edition, Pearson Education, 2006.

Thank You.