

18-Gradient_Boostring

October 20, 2024

1 Gradient Boosting

Gradient Boosting is an ensemble learning technique that builds models in a stage-wise fashion. It creates a strong predictive model by combining the predictions of several weaker models (often decision trees). Unlike AdaBoost, which focuses on misclassified instances, Gradient Boosting minimizes a loss function using the gradient descent optimization algorithm, hence the name “Gradient Boosting.”

1.0.1 When to Use Gradient Boosting?

Gradient Boosting is particularly useful when:

- You have structured or tabular data.
- You want to achieve high accuracy and predictive performance.
- Your data contains complex patterns that simpler models struggle to capture.
- You are interested in both regression and classification tasks.

Gradient Boosting can handle a variety of loss functions and is often employed in Kaggle competitions due to its robustness.

1.0.2 How Does Gradient Boosting Work?

Gradient Boosting works by sequentially adding weak learners (typically shallow decision trees) to the ensemble, each one correcting the errors made by the previous learners. Here’s a step-by-step breakdown:

1. Initialize the Model:

- Start with a base prediction (often the mean of the target values for regression tasks).

2. Calculate Residuals:

- Calculate the residuals (the difference between actual values and predicted values) of the current model.

3. Fit a Weak Learner:

- Train a new weak learner (e.g., decision tree) on the residuals to predict these errors. The idea is to learn the error of the previous model.

4. Update the Model:

- Update the model by adding the predictions of the new weak learner to the current model predictions, scaled by a learning rate (denoted as η).

5. Repeat:

- Repeat steps 2 to 4 for a specified number of iterations or until a stopping criterion is met.

6. Final Prediction:

- The final prediction is a weighted sum of the weak learners' predictions.

1.0.3 Who Should Use Gradient Boosting?

- **Data scientists and machine learning practitioners:** Who want to build predictive models that leverage the power of ensemble learning.
- **Kaggle competitors:** Gradient Boosting is popular in data science competitions for its high predictive accuracy and flexibility.
- **Industries that require predictive analytics:** Applicable in finance, marketing, healthcare, and other fields that deal with regression or classification problems.

Advantages of Gradient Boosting:

- **High predictive performance:** Often achieves state-of-the-art results on various tasks.
- **Flexibility:** Can optimize various loss functions and handle different types of data (e.g., continuous, categorical). **Robustness to overfitting:** When used with techniques like early stopping or regularization (e.g., limiting tree depth).
- **Feature importance:** Provides insight into the importance of features in making predictions.

Disadvantages of Gradient Boosting:

- **Sensitive to noise and outliers:** Can overfit if not properly regularized.
- **Longer training times:** Sequential training can be slower compared to parallelized models like Random Forest.
- **Complexity:** Requires careful tuning of hyperparameters (e.g., number of trees, learning rate, tree depth) for optimal performance.

1.0.4 Real-World Applications of Gradient Boosting:

- **Finance:** Credit scoring and risk assessment, where high accuracy is crucial.
- **Customer segmentation:** In marketing analytics, to understand customer behavior.
- **Healthcare:** Predictive models for patient outcomes and disease risk.
- **Recommendation systems:** To personalize user experiences based on historical data.

```
[19]: import pandas as pd

from sklearn import datasets
from sklearn.model_selection import train_test_split, cross_val_score, GridSearchCV
```

```
from sklearn.ensemble import GradientBoostingClassifier
```

```
wine = datasets.load_wine(as_frame=True)
wine
```

```
[19]: {'data':      alcohol  malic_acid  ash  alcalinity_of_ash  magnesium
total_phenols  \
0      14.23      1.71  2.43      15.6      127.0      2.80
1      13.20      1.78  2.14      11.2      100.0      2.65
2      13.16      2.36  2.67      18.6      101.0      2.80
3      14.37      1.95  2.50      16.8      113.0      3.85
4      13.24      2.59  2.87      21.0      118.0      2.80
..      ...      ...      ...      ...      ...      ...
173    13.71      5.65  2.45      20.5      95.0      1.68
174    13.40      3.91  2.48      23.0      102.0      1.80
175    13.27      4.28  2.26      20.0      120.0      1.59
176    13.17      2.59  2.37      20.0      120.0      1.65
177    14.13      4.10  2.74      24.5      96.0      2.05

      flavanoids  nonflavanoid_phenols  proanthocyanins  color_intensity  hue
\
0      3.06      0.28      2.29      5.64  1.04
1      2.76      0.26      1.28      4.38  1.05
2      3.24      0.30      2.81      5.68  1.03
3      3.49      0.24      2.18      7.80  0.86
4      2.69      0.39      1.82      4.32  1.04
..      ...      ...      ...      ...      ...
173    0.61      0.52      1.06      7.70  0.64
174    0.75      0.43      1.41      7.30  0.70
175    0.69      0.43      1.35     10.20  0.59
176    0.68      0.53      1.46      9.30  0.60
177    0.76      0.56      1.35      9.20  0.61

      od280/od315_of_diluted_wines  proline
0      3.92     1065.0
1      3.40     1050.0
2      3.17     1185.0
3      3.45     1480.0
4      2.93      735.0
..      ...      ...
173    1.74      740.0
174    1.56      750.0
175    1.56      835.0
176    1.62      840.0
177    1.60      560.0
```

```

[178 rows x 13 columns],
'target': 0      0
1      0
2      0
3      0
4      0
..
173    2
174    2
175    2
176    2
177    2
Name: target, Length: 178, dtype: int32,
'frame':      alcohol  malic_acid  ash  alkalinity_of_ash  magnesium
total_phenols \
0      14.23      1.71  2.43      15.6      127.0      2.80
1      13.20      1.78  2.14      11.2      100.0      2.65
2      13.16      2.36  2.67      18.6      101.0      2.80
3      14.37      1.95  2.50      16.8      113.0      3.85
4      13.24      2.59  2.87      21.0      118.0      2.80
..      ...      ...  ...      ...      ...      ...
173    13.71      5.65  2.45      20.5      95.0      1.68
174    13.40      3.91  2.48      23.0      102.0      1.80
175    13.27      4.28  2.26      20.0      120.0      1.59
176    13.17      2.59  2.37      20.0      120.0      1.65
177    14.13      4.10  2.74      24.5      96.0      2.05

      flavanoids  nonflavanoid_phenols  proanthocyanins  color_intensity  hue
\
0      3.06      0.28      2.29      5.64  1.04
1      2.76      0.26      1.28      4.38  1.05
2      3.24      0.30      2.81      5.68  1.03
3      3.49      0.24      2.18      7.80  0.86
4      2.69      0.39      1.82      4.32  1.04
..      ...      ...      ...      ...  ...
173    0.61      0.52      1.06      7.70  0.64
174    0.75      0.43      1.41      7.30  0.70
175    0.69      0.43      1.35     10.20  0.59
176    0.68      0.53      1.46      9.30  0.60
177    0.76      0.56      1.35      9.20  0.61

      od280/od315_of_diluted_wines  proline  target
0      3.92     1065.0      0
1      3.40     1050.0      0
2      3.17     1185.0      0
3      3.45     1480.0      0
4      2.93      735.0      0

```

..
173	1.74	740.0	2
174	1.56	750.0	2
175	1.56	835.0	2
176	1.62	840.0	2
177	1.60	560.0	2

```
[178 rows x 14 columns],
'target_names': array(['class_0', 'class_1', 'class_2'], dtype='<U7'),
'DESCR': '.. _wine_dataset:\n\nWine recognition
dataset\n-----\n\n**Data Set Characteristics:**\n\nNumber of
Instances: 178\nNumber of Attributes: 13 numeric, predictive attributes and the
class\nAttribute Information:\n    - Alcohol\n    - Malic acid\n    - Ash\n
- Alcalinity of ash\n    - Magnesium\n    - Total phenols\n    - Flavanoids\n
- Nonflavanoid phenols\n    - Proanthocyanins\n    - Color intensity\n    -
Hue\n    - OD280/OD315 of diluted wines\n    - Proline\n    - class:\n    -
class_0\n    - class_1\n    - class_2\n\nSummary
Statistics:\n\n===== \n
Min  Max  Mean  SD\n===== \n
===== \nAlcohol:          11.0  14.8   13.0   0.8\nMalic Acid:
0.74  5.80   2.34  1.12\nAsh:          1.36  3.23   2.36
0.27\nAlcalinity of Ash:          10.6  30.0   19.5   3.3\nMagnesium:
70.0 162.0   99.7  14.3\nTotal Phenols:          0.98  3.88   2.29
0.63\nFlavanoids:          0.34  5.08   2.03  1.00\nNonflavanoid
Phenols:          0.13  0.66   0.36  0.12\nProanthocyanins:          0.41
3.58   1.59  0.57\nColour Intensity:          1.3  13.0   5.1   2.3\nHue:
0.48  1.71   0.96  0.23\nOD280/OD315 of diluted wines: 1.27  4.00   2.61
0.71\nProline:          278  1680   746
315\n===== \n\nMissing
Attribute Values: None\nClass Distribution: class_0 (59), class_1 (71), class_2
(48)\nCreator: R.A. Fisher\nDonor: Michael Marshall
(MARSHALL%PLU@io.arc.nasa.gov)\nDate: July, 1988\n\nThis is a copy of UCI ML
Wine recognition datasets.\nhttps://archive.ics.uci.edu/ml/machine-learning-
databases/wine/wine.data\n\nThe data is the results of a chemical analysis of
wines grown in the same\nregion in Italy by three different cultivators. There
are thirteen different\nmeasurements taken for different constituents found in
the three types of\nwine.\n\nOriginal Owners:\nForina, M. et al, PARVUS -\nAn
Extendible Package for Data Exploration, Classification and
Correlation.\nInstitute of Pharmaceutical and Food Analysis and
Technologies,\nVia Brigata Salerno, 16147 Genoa, Italy.\n\nCitation:\n\nLichman,
M. (2013). UCI Machine Learning Repository\n[https://archive.ics.uci.edu/ml].
Irvine, CA: University of California,\nSchool of Information and Computer
Science.\n\n|details-start|\n**References**\n|details-split|\n\n(1) S.
Aeberhard, D. Coomans and O. de Vel,\nComparison of Classifiers in High
Dimensional Settings,\nTech. Rep. no. 92-02, (1992), Dept. of Computer Science
and Dept. of\nMathematics and Statistics, James Cook University of North
Queensland.\n(Also submitted to Technometrics).\n\nThe data was used with many
```

others for comparing various classifiers. The classes are separable, though only RDA has achieved 100% correct classification. (RDA : 100%, QDA 99.4%, LDA 98.9%, 1NN 96.1% (z-transformed data)) (All results using the leave-one-out technique)

(2) S. Aeberhard, D. Coomans and O. de Vel, "THE CLASSIFICATION PERFORMANCE OF RDA" Tech. Rep. no. 92-01, (1992), Dept. of Computer Science and Dept. of Mathematics and Statistics, James Cook University of North Queensland. (Also submitted to Journal of Chemometrics).

```
{
  'feature_names': ['alcohol',
    'malic_acid',
    'ash',
    'alcalinity_of_ash',
    'magnesium',
    'total_phenols',
    'flavanoids',
    'nonflavanoid_phenols',
    'proanthocyanins',
    'color_intensity',
    'hue',
    'od280/od315_of_diluted_wines',
    'proline']}]}
```

```
[20]: X = wine['data']
      y = wine['target']
```

```
[21]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
      random_state=17)

gbr = GradientBoostingClassifier()
gbr.fit(X_train, y_train)
```

```
[21]: GradientBoostingClassifier()
```

```
[22]: cross_val_score(gbr, X_train, y_train, cv=3, n_jobs=-1).mean()
```

```
[22]: 0.9221335697399526
```

```
[23]: param_grid = {
      'n_estimators': [10,50,100, 500],
      'learning_rate': [0.0001, 0.001, 0.01, 0.1, 1],
      'max_depth': [3,5,7,9]
    }

gbr2 = GridSearchCV(gbr, param_grid, cv=3, n_jobs=-1)
gbr2.fit(X_train, y_train)
```

```
[23]: GridSearchCV(cv=3, estimator=GradientBoostingClassifier(), n_jobs=-1,
      param_grid={'learning_rate': [0.0001, 0.001, 0.01, 0.1, 1],
```

```
'max_depth': [3, 5, 7, 9],  
'n_estimators': [10, 50, 100, 500]})
```

```
[24]: gbr2.best_params_
```

```
[24]: {'learning_rate': 0.1, 'max_depth': 3, 'n_estimators': 500}
```

```
[25]: gbr2.best_score_
```

```
[25]: 0.9434101654846335
```