21/04

Random Forest Algorithm

Input:
- Dataset D with features X and labels Y
- Number of trees T
- Number of features to consider at each split m

Output:
- A forest (ensemble) of decision trees for prediction.

Algorithm:

1. Initialize:
- Choose the number of trees T
- Set the number of features m to consider for splitting ($\sqrt{n}$ for classification, $n/3$ for regression, where n is total features)

2. For each tree $t = 1$ to T:
- Draw a bootstrap sample from the training data (sample with replacement)
- Train a Decision Tree on this sample
  - At each node
  - Randomly select m features from the total n features
  - Find the best split among the selected features.
  - Split the node and repeat recursively

PAGE NO :
DATE :

Tree is grown to the maximum depth or until minimum node size is reached

3. End for

4. Prediction :
For prediction
- Each tree votes for a class
- Final prediction = majority vote.

For regression :
- Each tree gives a value
- Final prediction = average of all tree outputs.

AdaBoost Classifier Algorithm

Goal : combining multiple weak classifiers to build a strong classifier

Input :
Training data
$D = \{(x_1, y_1), (x_2, y_2), \ldots (x_n, y_n)\}$
where $y_i \in \{-1, +1\}$
Number of boosting rounds : T

Output :
Final strong classifier :
$H(x)$

PAGE NO :
DATE :

Algorithm:

1. Initialize sample weights
Assign equal weights to all training examples.

$$w_1(i) = \frac{1}{n}, \text{ for } i = 1, 2, \ldots n.$$

2. For each boosting round $t = 1$ to T:

a. Train a weak classifier $h_t(x)$:
- Train using the current weights

b. Compute the weighted classification error:

c. Compute the importance (weight) of the weak classifier.

d. Update sample weights

$$w_{t+1}(i) = w_t(i) \cdot \exp(-\alpha_t y_i h_t(x_i))$$

- Misclassified samples will have their weights increased
- Normalize the weights.

3. Final strong classifier:

$$H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

- It is a weighted majority vote of all weak classifiers