Contents

Chapter 1

Decision Trees for Classification

A decision tree is a popular machine learning algorithm used for both regression and classification problems. In this discussion, we will focus on its application in classification tasks.

1.1 Decision Trees for Classification

A decision tree for classification uses a tree structure to predict the class of an object based on its features. The tree is made up of nodes that split the data based on a feature value, and leaves that represent a class label. The idea is to create a tree that has minimum impurity, i.e., at the end of the tree, we would like each leaf to contain data points that belong to a single class.

1.2 Gini Impurity

Gini impurity is a measure of misclassification, which applies in a multiclass classifier context. It gives an idea of how often a randomly chosen element from the set would be incorrectly labeled if it was randomly labeled according to the distribution of labels in the subset.

The Gini impurity for a node of the tree is calculated as:

$$\mathrm{Gini}(p) = 1 - \sum_{i=1}^{J} (p_i)^2 \tag{1.1}$$

where p_i is the fraction of items classified to label i at a node and J is the total number of classes.

A Gini impurity of 0 is the best score, where all elements in a partition fall into a single category.

1.3 How Gini Impurity is Used

During the construction of a decision tree, the best feature to split on at each node is chosen by minimizing the Gini impurity of the child nodes. The algorithm will consider all features and all possible split points for each feature to find the split that yields the lowest weighted average Gini impurity.

CHAPTER 2

Bagging and Random Forests

Bagging (Bootstrap Aggregating) and Random Forests are ensemble machine learning methods that are primarily used to improve the stability and accuracy of prediction models.

2.1 Bagging

Bagging, an abbreviation for Bootstrap Aggregating, is a method for generating multiple versions of a predictor and using these to get an aggregated predictor. The aggregation averages the output (for regression) or performs a vote (for classification).

Given a standard training set D of size n, bagging generates m new training sets D_i, each of size n', by sampling from D uniformly and with replacement. By sampling with replacement, some observations may be repeated in each D_i. If $n' = n$, then for large n the set D_i is expected to have the fraction $(1 - 1/e) \approx 63.2\%$ of the unique examples of D, the rest being duplicates.

2.2 Random Forests

Random Forests is a substantial modification of Bagging that builds a large collection of de-correlated trees, and then averages them. When building these decision trees, each time a split in a tree is considered, a random sample of k features is chosen as split candidates from the full set of features. The split is allowed to use only one of those k features. A fresh sample of k features is taken at each node, and the best feature/split-point among the k is chosen.

For classification problems, $k = \sqrt{p}$ is typically taken, where p is the number of features in the model. For regression problems, the inventors recommend $k = p/3$, with a minimum node size of 5 as the default.

In Random Forests, there is no need for cross-validation or a separate test set to get an unbiased estimate of the test set error. It is estimated internally, during the run, as follows:

1. Each tree is constructed using a different bootstrap sample from the original data.
2. About one-third of the cases are left out of the bootstrap sample and not used in the construction of the k-th tree.

3. Let $y_{\text{tree } k}(x)$ be the class prediction of the k-th Random Forest tree for x. Then the Random Forest classifier does a majority vote over all trees:

$$y_{RF}(x) = \text{majority}\{y_{\text{tree } k}(x), k = 1, \ldots\}$$

Chapter 3

Boosting

Boosting is a machine learning ensemble meta-algorithm primarily used to reduce bias, and to a lesser extent variance, in supervised learning. It works by iteratively learning weak classifiers and adding them to a final strong classifier in a way that the subsequent weak learners try to correct the mistakes of the previous ones.

3.1 AdaBoost

AdaBoost, short for Adaptive Boosting, is one of the first and simplest boosting algorithms. Given a set of n training examples $(x_1, y_1), \ldots, (x_n, y_n)$ where y_i are binary outputs, the algorithm works as follows:

1. Initialize weights $w_i = 1/n$ for $i = 1, \ldots, n$.
2. For $t = 1$ to T:
   - Train a weak learner h_t using the weighted examples.
   - Compute the weighted error $\epsilon_t = \sum_{i:h_t(x_i) \neq y_i} w_i$.
   - Set $\alpha_t = \frac{1}{2} \log\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$.
   - Update the weights: $w_i = w_i \exp(-\alpha_t y_i h_t(x_i))$ for $i = 1, \ldots, n$, and normalize them so that they sum to one.
3. The final model is $H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$.

3.2 Gradient Boosted Trees

Gradient Boosted Trees is a generalization of boosting to arbitrary differentiable loss functions. It works by sequentially adding predictors to an ensemble, each one correcting its predecessor by fitting the new predictor to the residual errors.

Chapter 4

Clustering using k-Means

K-means is a popular unsupervised learning algorithm used for data clustering. The goal of k-means is to group data points into distinct non-overlapping subgroups, or clusters, based on their features.

4.1 The K-Means Algorithm

Given a dataset $X = \{x_1, x_2, \ldots, x_N\}$, where each x_i is a d-dimensional vector, and an integer k, the k-means clustering algorithm seeks to find k cluster centroids $C = \{c_1, c_2, \ldots, c_k\}$ such that the distance from each data point to its nearest centroid is minimized.

The k-means algorithm works as follows:

1. Initialize k centroids randomly.
2. Assign each data point to the nearest centroid. This forms k clusters.
3. For each cluster, update its centroid by computing the mean of all points in the cluster.
4. Repeat steps 2 and 3 until the centroids do not change significantly or a maximum number of iterations is reached.

The measure of distance typically used in k-means is the Euclidean distance. For two d-dimensional vectors $x = (x_1, \ldots, x_d)$ and $y = (y_1, \ldots, y_d)$, the Euclidean distance is defined as:

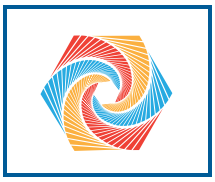
$$d(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \ldots + (x_d - y_d)^2} \tag{4.1}$$

4.2 Choosing K

Choosing an appropriate value for k is a significant aspect of the k-means algorithm. One common method is the Elbow Method, which involves plotting the explained variation as a function of the number of clusters, and picking the elbow of the curve as the number of clusters to use.

Appendix A

Answers to Exercises



Index