[1]

In this lecture we will consider several improvements to address the issue of accuracy and overfitting in decision trees. We will consider the idea of model ensembles where, rather than trying to build a single accurate decision tree model, we instead choose to build multiple competing trees and using the aggregate predictions from these in the hope of getting better accuracy and reduce the risk of overfitting

[2]

A “model ensemble” is a model build from a set of cooperating models. For our purposes, we will assume that means a collection of decision trees. Each model is built separately and independently of the others and makes its own predictions independently. Importantly, the models are built on different variations of the training data, usually involving a random sampling approach. The aggregated output of the model ensemble is the consideration of each of the ensemble’s member predictions and choosing the best result across these. What “best” means here will depend on whether the model ensemble is trying to predict a categorical value (i.e. a classification) or a continuous value (i.e. a regression). If categorical, then a weighted majority vote could suffice. If continuous, then a measure of the central tendency of the feature variable could be used.

[3]

As previously noted, decision tree algorithms are very sensitive to changes in the training dataset. A small change in the dataset can have a big effect on the execution of the algorithm such as causing a a different feature to chosen to be the root of a subtree. If this change impacts a subtree high up in the tree, this can have a ripple down effect on the overall tree and impact the performance and accuracy of its predictive modelling ability. There are two possible mitigations we can consider when building our information-based models as part of model ensembles. These are “bootstrapping” and “subspace sampling”.

[4]

The idea behind bootstrapping is to train the model over a set of fixed-sized, randomly chosen samples from the dataset. If the dataset is not particularly large, then this random sampling is usually performed as sampling-with-replacement, meaning that the same instances can appear in the same sample or multiple other samples and be considered by multiple training builds. The objective is that every model will be trained on sufficiently different examples but, in aggregate, will not consider all the dataset instances during the training phase. This technique promotes greater diversity in the models being built which helps to reduce the risk of overfitting.

[5]

Whereas bootstrapping has to do with randomly sampling instances from the dataset (i.e. rows), subspace sampling has to do with randomly sampling features from the dataset (i.e. columns). This column-based subsampling also promotes greater diversity in the resultant models. By considering fewer columns, this makes the tree faster to build and ultimately shallower and faster to evaluate. The combined use of bootstrapping and subspace sampling when building decision tree models is called a “random forest”. Subsampling continues to play a role when building each subtree of the decision tree in that a different subsample of features can be used at each level. The choice of the size of the subsample is initially randomly chosen but maintained. For each decision tree to be built within a forest, we use the same subsample size. Later we will see that this can be adjusted up or down in size based on the accuracy of the generated model.

[6]

A random forest is an example of a model ensemble. The idea is to generate multiple random decision trees, ideally hundreds or even thousands depending on the dataset size, using random bootstrapping and random subsampling. Combining and comparing the outputs of aggregated bootstrapped models is called bagging. To use a random forest to make a prediction for a target feature variable, we feed the unseen descriptive feature values through each of the decision trees in our forest and note their individual results. For example, returning to our heart disease diagnosis dataset from the previous lecture, suppose we provide a yet unseen sample to each of our trees, we would then use the simple majority vote of our individual forest predictions to make the final, aggregate prediction of a positive or negative diagnosis for heart disease.

[7]

Measuring the accuracy of a random forest follows a similar process to testing any other machine learning model. We collect the out-of-bag samples, that is those sample instances not used in the training step of the random forest and submit these to the model for comparison with the expected predictions. This means that each out-of-bag sample is submitted to each tree in the forest for prediction and its result noted. Again, using the example from the heart disease data set, a simple majority vote is used to make the aggregate prediction. The accuracy of the random forest model is the proportion of out-of-bag sample predictions matching the expected predictions.

[8]

At this point in the build process, having built and tested a random forest for a given, randomly chosen subsample size, we can go back a choose different subsample sizes and generate models for each. These are similarly tested for accuracy and compared to each other, that is the performance of different random forests is compared. The best performing random forest with the optimal subsample size becomes the final forest representing the trained model for our problem domain given the data we have.

[9]

In summary, we have seen the application of ensemble learning applied to decision trees to build a more accurate and robust model training strategy. A collection of decision trees, called a random forest, is built using random bootstrapping and random subspace sampling. The resultant model is tested against the out-of-bag data to evaluate its accuracy. This process is repeated with different subspace sample sizes until the most accurate random forest model is derived. While the model development process is more complex and time consuming, random forests generally overcome the main weaknesses with simple decision trees, name a lack of accuracy, sensitivity to training data and the risk of overfitting.