[1]

In this lecture, we will examine a machine learning approach based the formal concepts from “information theory”. We will use information theory to build a data structure called a “decision tree” which can be built as a supervised learning model on some training data and then used to make predictions about data not previously seen. The approach captured by decision trees is one of the simplest and intuitive machine learning approaches of all those that you will study on this course because it resembles very closely the likely manual approach that most programmers would take if coding a learning model by hand.

[2]

Assume that we have some set of descriptive features and a target label for a dataset of instances collected from some problem domain. We can build a learning model from that data using a decision tree. A decision tree is a recursive, ordered graph of nodes having no cycles. The tree comprises a root node, some interior nodes and leaf nodes, also called terminating nodes. Non-leaf nodes (root or interior nodes) have one or more branches connecting to other non-leaf nodes and leaf nodes. A branch represents a relational test to be carried out on the value of a descriptive feature in the parent node and a decision as to which child node path to follow. Descriptive feature values can be categorical or continuous. A tree can be arbitrarily wide and arbitrarily deep. The tree’s shape will depend on the number of distinct features being considered and the spread of values of those features to be considered. Using a programming analogy, you can think of a decision tree as a big, automatically generated “if-then-else” statement.

[3]

As you will recall from studying the tree abstract data structure in computer science, the computational complexity of processing a tree is proportional to logarithmic in the number of nodes in a tree. But this performance guarantee is only valid if the tree is balanced. A balanced tree is a tree in which the subtrees of any given interior node do not differ in height by more than one or two levels. Recall that the height of a sub-tree is the number of levels from the root of that subtree to it lowest leaf node. It follows from performance property of trees that it would be an important goal when building decision trees is that they should be balanced too. The problem we must solve is how to partition the training dataset to choose the most appropriate descriptive features to place at each node and at each level to ensure our resulting decision tree is balanced with respect to the decisions at each branch. Achieving balance is another way of reducing the overall depth of the tree. A shallower tree means fewer computations are required to traverse from the root node to the leaf nodes of the tree. To tackle this problem, we turn to information theory and the concepts of “entropy” and “information gain”.

[4]

The famous computer science theorist and researcher, Claude Shannon, proposed a model for computing a measure of the impurity of elements in a set. To understand this idea, we need to go back to probability theory. Suppose we have set of elements in a set, for example an opaque bag of coloured balls. Suppose the bag contains sixteen balls of eight different colours. If you were to randomly draw a ball from the bag, then there would a significant degree of uncertainly as to which colour you might pick out. But suppose, on the other hand, that twelve of those ball was the same colour and there were only two different colours in total, then there would be a lot less uncertainly about what coloured ball you might draw. The formal term of the level of uncertainty in these experiments is entropy. Low entropy means low uncertainty. High entropy means high uncertainty. You can think of entropy as a measure of the heterogeneity of a set. This means how different to each other elements of the set are.

[5],

In mathematical terms, we define the entropy of a set as a function of the relative frequencies of the members of that set, in other words, the probability of randomly choosing an element from that set. The higher the probability, the lower the uncertainly and, therefore the lower the entropy that that element contributes to the set. Recall that the domain of a set is the allowed values of that set and that those values form a distribution. Formally, entropy is defined as the sum of probability of a member having a value from the domain multiplied by the logarithm of that probability. The reason that a logarithm is used here is to invert the effect of the relative frequency of an element so that less frequently occurring elements contribute more to entropy than do more frequently occurring elements.

[6]

So how can entropy be used to help build a decision tree? The idea is to use a concept called “information gain” to help to chose which descriptive feature to place at the root of each subtree to offer the best discrimination of the remaining features with respect to the target feature. Information gain measures how well a chosen descriptive feature would reduce the entropy, that is, add greater certainty at the next branching decision. The more efficiently that we reduce the entropy, the more quickly we can discriminate between feature values that contribute to our target result. Remember the goal of the decision tree is to represent how the descriptive features can lead to a labelled target value. To calculate the potential information gain associated with a particular feature, we must partition our original dataset into multiple subsets, one subset for each of the distinct values of the feature in question. For example, suppose a feature had a domain of “true” or “false”, then this would result in two data subsets. One subset would represent all of the “true” values for our feature and the other would represent all of the “false” values. The number of subsets generated by a feature is equal to the number of distinct values of that feature in the training set. For continuous valued features, we may consider reducing the total number of subsets actually considered, a topic we will return to shortly. Once we have each of the partitions for a feature, we can compute the remaining entropy associated with that feature as the sum of the weighted entropies of each partition with respect to the target feature.

[7]

The information gain for a descriptive feature is the difference between the total entropy with respect to the target feature and the remaining entropy, having partitioned the dataset using that feature. The feature offering the best information gain at a given level of the tree is the best candidate feature to place as the root node of the next subtree. This is because it implies that this feature is the best discriminator of the data at this point in the tree construction.

[8]

The ID3 algorithm is an example of a decision tree builder based on information gain. The algorithm recursively builds a set of subtrees from the root node down to the leaf nodes by selecting the best discriminating feature to be the root of each subtree. When a branch fully discriminates a target feature value, then it is terminated as a leaf node. When the final leaf node is created, the decision tree is complete. The ID3 algorithm accepts a set of labelled training instances of descriptive features and proceeds as shown. This description of the steps assumes a binary classification target variable, but the approach generalises to multiple target values. As you might expect, the definition of ID3 is recursive in the function itself.

[9]

When feature values are categorical and have nominal values from a relatively small domain, we can consider of the domain’s values when creating partitions over than feature. However, when features are continuous, there could be a large range of possible values that the feature could take on. This could be impractical and inefficient so we may need to consider reducing the total number of considered values to a manageable size. Further, while categorical nominal values are always considered for equality in a branch test, continuous values are often tested within some range of values. Some of the ways we can reduce the total number of continuous values to consider include bucketing the values into specified ranges either as fixed-sized intervals from the minimum value to the maximum value or considering values within some number of standard deviations of the mean in a Gaussian distribution of the value range.

[10]

By way of illustration, let’s consider the sample data set shown here. This data describes patient examination results for heart disease with a binary categorical target feature for positive and negative diagnosis outcomes. The dataset includes a mix of categorical and continuous descriptive features measured for three hundred and three samples from a range of patient ages. Our goal is to build a decision tree using a subset of this data (a training set) to predict heart disease outcomes for unseen patient data.

[11]

Taking a 75% split for training data, the computed entropy of the target feature with respect to the total training set is 0.69 bits. To build the decision tree, we need to select a descriptive feature from our data set which best discriminates the data at this level. Following the procedure described by the ID3 algorithm, we partition each of the dataset into value-sepa3rated ranges for each distinct feature and compute the resulting information gain. The feature showing the best information gain will be our candidate for root node. For example, the feature “sex” (describing whether male for female patients) partitions the dataset into two groups of sixty-nine and one hundred and fifty-three samples respectively. The information gain for this feature is 0.03 bits. Another feature, the “cp” feature splits the dataset into four separate partitions and yields an information gain of 0.16 bits. Of all of the features considered at this level, the “cp” feature produces the most information gain and we therefore proceed with this as our root node having four emanating branches for each of the possible four values of the “cp” feature. We proceed to the next level, by excluding the “cp” feature from consideration and partitioning dataset using the remaining features to determine the subtree root features for the next level down. The process completes when we have considered the final leaf node of the decision tree.

[12]

When the training of the model (i.e. building a tree) has completed it can be tested for accuracy using the test data held back when we separated the training data out. The test data is also labelled so we can submit each of the instances to the decision tree and compare the outputted prediction with the expected to prediction to evaluate the accuracy of the trained model.

[13]

There are several issues to consider when building decision trees, some of which are common to all machine learning scenarios. The first of these is the issue of model overfitting. As usual, if the dataset being used for training is not large enough or does not have the same or close to the same data distribution as the real-world problem domain, then the model will have a certain bias towards the training data and may perform poorly when being used to make predictions against unseen data. This can be a problem for continuous feature data where selecting the right approach to splitting and be difficult. Decision trees built with greedy algorithms like ID3, do not also generate the most optimal global solutions tending towards local optima in data.

[14]

In summary, we have explored a supervised machine learning model called a decision tree built with the ID3 algorithm which we can consider as an optimised, automatically generated branch-prediction system based on samples of training data. The choice of descriptive features and the branching at each subtree root is determined by the best information gain produced at that subtree. Information gain is calculated using Shannon’s entropy model which is a measure of the heterogeneity of data elements in a set. The main advantages of decision trees is that they are easy to build, easy to use and intuitively easy to interpret. But they suffer, compared with other kinds of machine learning models, with accuracy problems. In a later lecture we will explore variations on information-based learning approaches which address some the shortcomings of basic decision trees such as model bias, sensitivity to training data, accuracy and overfitting.