Worksheet 2 – Information Theory and Decision Trees

Theory

Review the videos from week 2. Answer the following questions based on those lectures

1. Describe the main elements of a decision tree and its theory of operation.

Answer: 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. 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.

1. Explain the term “entropy” from information theory? How is it calculated?

Answer: 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.

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. As is seen in the formula shown below.

1. What is meant by “information gain” and how is it used by an algorithm like ID3?

Answer: Information gain helps to choose 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 and measures how well a chosen descriptive feature would reduce the entropy, that is, add greater certainty at the next branching decision.

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.

1. What is data partitioning in the context of decision tree construction?

Answer: 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. The number of subsets generated by a feature is equal to the number of distinct values of that feature in the training set. 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.

1. Briefly describe some of the main advantages and disadvantages of decision trees as a machine learning approach.

Advantages:

It’s easy to understand and build.

It’s efficient and only needs to be built once.

It can deal with large data in a relatively short time.

Disadvantages:

It has large model bias.

The accuracy of decision tree is low.

It has the problem of overfitting.

Decision trees are hyper sensitive to the training data.

1. What is meant by “bootstrapping” and “subspace sampling”?

Answer: 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.

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.

1. What is a model ensemble and what are its potential benefits?

Answer: A “model ensemble” is a model build from a set of cooperating models. Importantly, the models are built on different variations of the training data, usually involving a random sampling approach, which is beneficial for our prediction.

1. Describe the operation of a random forest. What role does subsampling play?

Answer: 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. We would then use the simple majority vote of our individual forest predictions to make the final prediction. 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.

Practice

Follow the tutorial videos from week 2 and carry out the following steps

1. Download the code archive and extract the file from the week 2 learning materials. Make sure that you can run the examples code as provided.
2. Use the code in **tree.py** as the starting point to build a full implementation of ID3 in in Python
3. Train and test your implementation against the heart diagnosis dataset. You should include all of the available What accuracy does your implementation achieve?

Answer: The accuracy I have achieved was approximately 67%. You can check the tree.py for further details.



1. Run the **sklearn\_tree.py** implementation and compare with your result.

Answer: The accuracy of sklearn\_tree.py implementation is 66%. But it doesn’t contain all of the features.



1. An alternative measure to entropy for implementing decision trees is to use the **Gini Index**. The Gini index for a dataset **D** with respect to a target variable **t** over a feature value domain of **V** is defined as:

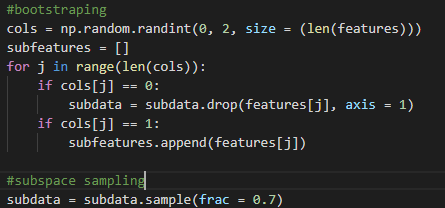
The Gini Index can be thought of as the probability that a feature would be misclassified and always has a value between 0 and 1. The advantage is that this is less intensive than calculating entropy because there is no logarithm computation required.

Modify your decision tree implementation to use the Gini Index instead of entropy. The calculation of information gain is similar. It is the Gini Index of the whole dataset minus the weighted Gini Index values for each partition.

Answer: The accuracy using Gini index is approximately 62%. As is shown in the picture. You can check the Gini\_tree.py code for further details.



* 1. Extend your decision tree implementation to implement **bootstrapping** and **subspace sampling**. Use these extensions to train and test a random forest for a set of different subspace sample sizes using the heart dataset. Compare your resulting model with the one in **sklearn\_forest.py**. How do they compare?





As is shown in the picture, I used the bootstrapping and subspace sampling methods simultaneously. And the accuracy was 68%. You can check the RandomForest.py for further details.