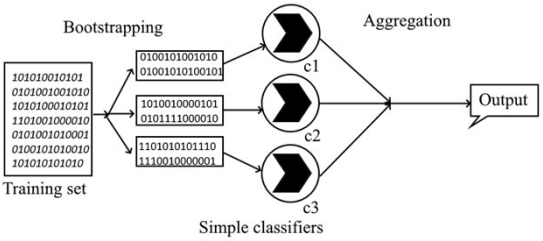
In this lecture we will take a closer look on how to create an ensemble using bagging and simple learners.

Bagging and ensembles revisited

- Remember:
  - An ensemble is a combination of simple methods
  - Output of each model is combined to form the final output
  - Bootstrap aggregation (bagging)
    - Split the training data into smaller sets (Bootstrapping)
    - Train each classifier on one of the sets
    - Combine the output of each classifier (Aggregation)

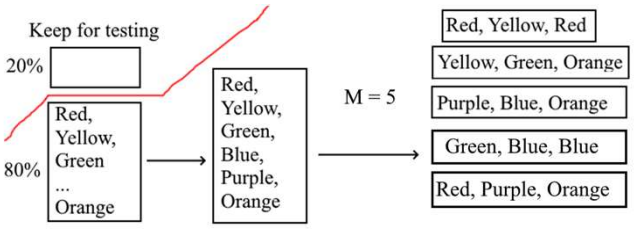


Remember from Lecture 2 – an ensemble is a combination of simple classifiers, where the outputs of each classifier are combined in some way to form the actual output. The process we will look at in this lecture is bagging, or bootstrap aggregation.

The bagging process can be summarized as follows: Start by splitting the training set into smaller sets, based on random samples. Then we train each classifier on one of the smaller datasets. When we test the ensemble with a new datapoint, we combine, or aggregate, the output of each model to form the actual output.

Bootstrapping

- Split the data into a training set and test set
  - 70-90% for training, the rest for testing and validation
- Split the remaining training set into smaller «bootstrapped» sets
  - We pick a random number of samples from the training set, with replacement
    - This implies that a sample may be picked multiple times for each bootstrapped set



First, we start by splitting our data into a training set and a testing set. A good rule of thumb is to use between 70 and 90 % as the training set, and the rest for testing. After we have split our data, we do the bootstrap aggregation, or bagging. As mentioned in Lecture 2, this is done by inducing a set of M learners starting with the same initial parameters with each training set chosen uniformly at random with replacement from the original training set.

What this means is that we have some data, and we decide a number M of learners to create. What this

means is that we have some data, and we decide a number M of learners to create. Let's say that M is 5.. We then generate 5 unique datasets by selecting random samples from the training set. This number should be smaller than the total size of the data, say 60%. The term "with replacement" means that we do not "remove" the randomly selected sample from the possible candidates. This means that the same sample may be chosen multiple times, or not at all for some of the bootstrapped sets.

Bootstrapping pseudocode

- all_data = read_file("datafile.csv")
- FOR EACH sample in all_data:
  - IF random(0, 1] < 0.2 //we split the data 20/80 between test and train
    - THEN put sample into the test data array
  - ELSE
    - put sample into the training data array
- create a set of *m* classifiers/models
- FOR EACH classifier c_i
  - bootstrap_data_ci = pick *n'* random samples from training data
- *(n'* is constant and the same for all bootstrapped sets)

The bootstrapping process may be summarized in the following psuedocode:

Read all the data into an array

Split the data into training set and testing set. In this example, we simply generate a random number between 0 and 1. Given that the distribution is uniform, we will end up with 20% of the data in a test set, and 80% in the training set.

Create a set of m simple classifiers

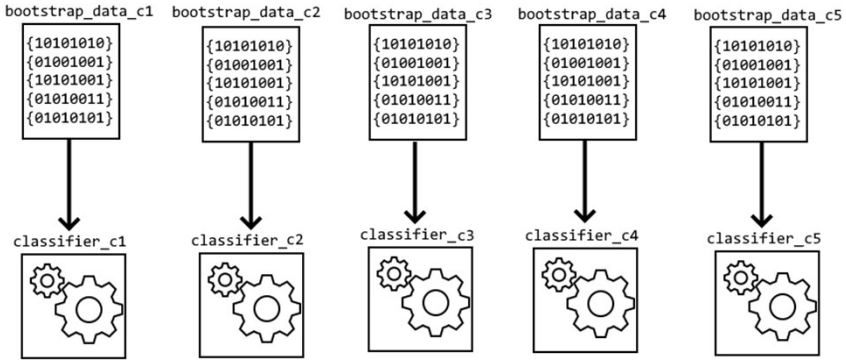
For each classifier, pick n' random samples from the

training set.

Each classifier needs to have their own bootstrapped training set.

Training the ensemble

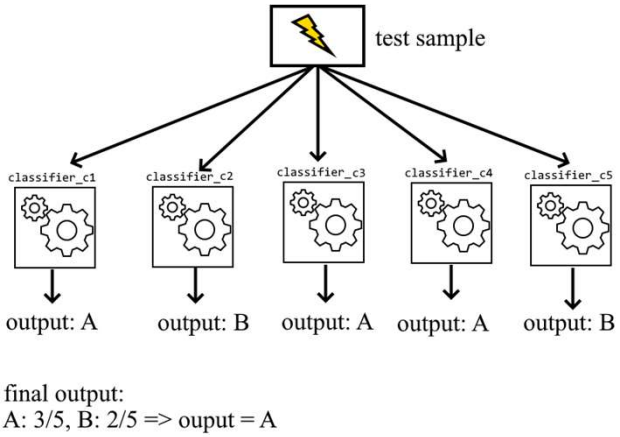
- We now have a set of *m* classifiers
- Each classifier is trained on its own bootstrapped training set



The next step is to train each simple classifier on their own bootstrapped dataset. The actual implementation and methods used for the classifier is not important in this example, but this can be something like a K-NN, Naïve Bayes or other.

Combining the outputs - aggregation

- We must combine the outputs of all the classifiers to get a prediction
  - Average value
  - Majority vote



The aggregation part of bagging comes into play when we combine, or aggregate, the outputs of each trained classifier in order to make a prediction. The simplest way is to use the average value for regressions, or the majority vote for classification problems. We test all 5 classifiers on the same sample. Let's say that 3 of our fictitious classifiers predict the class A, and the other predict class B. If we use the majority vote principle, the final output of the ensemble will thus be A.