Lecture 06
Evaluation Metrics

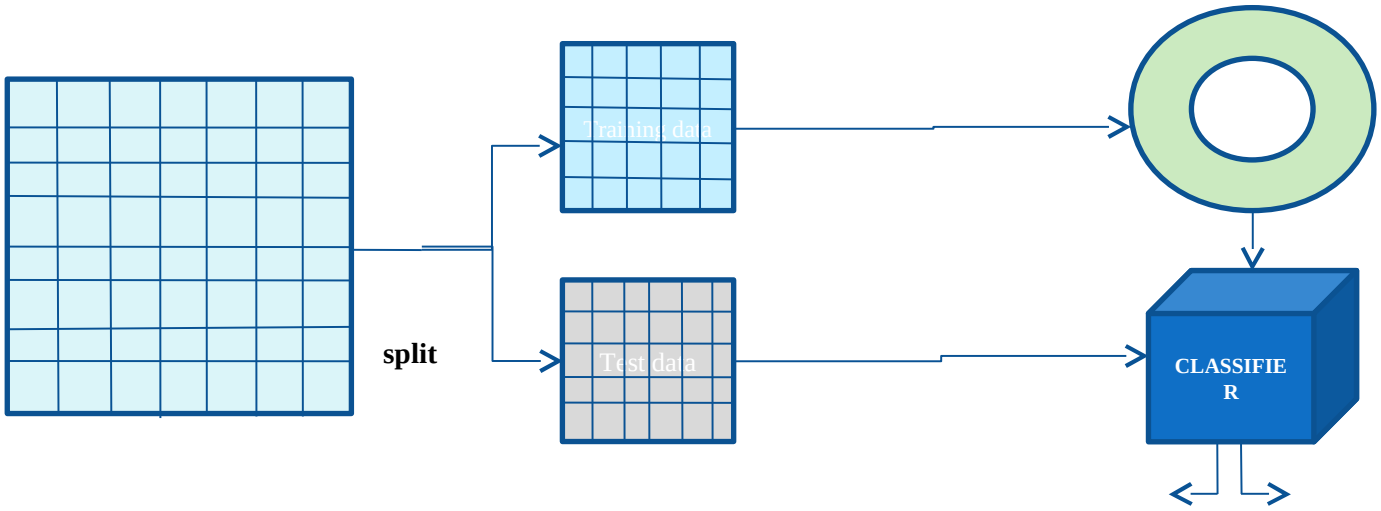
Introduction

- A classifier is used to predict an outcome of a test data. Such a prediction is useful in many applications
  - Business forecasting, cause-and-effect analysis, etc.
- A number of classifiers have been evolved to support the activities.
    - Each has their own merits and demerits
- There is a need to estimate the accuracy and performance of the classifier with respect to few controlling parameters in data sensitivity
- As a task of sensitivity analysis, we have to focus on
  - Estimation strategy
  - Metrics for measuring accuracy
  - Metrics for measuring performance

Estimation Strategy

Planning for Estimation

- Using some “training data”, building a classifier based on certain principle is called “learning a classifier”.
- After building a classifier and before using it for classification of unseen instance, we have to validate it using some “test data”.
- Usually training data and test data are outsourced from a large pool of data already available.



Estimation Strategies

- Accuracy and performance measurement should follow a strategy.
- Most widely used strategies are
  - (1) Holdout method
  - (2) Random subsampling
  - (3) Cross-validation
  - (4) Bootstrap approach

Holdout Method

- This is a basic concept of estimating a prediction.
- Given a dataset, it is partitioned into two disjoint sets called training set and testing set.
- Classifier is learned based on the training set and get evaluated with testing set.
- Proportion of training and testing sets is at the discretion of analyst; typically **1:1** or **4:1**, and there is a trade-off between these sizes of these two sets.
- If the training set is too large, then model may be good enough, but estimation may be less reliable due to small testing set and vice-versa.

Random Subsampling

- It is a variation of Holdout method to overcome the drawback of over-presenting a class in one set thus under-presenting it in the other set and vice-versa.

- In this method, Holdout method is repeated ***k*** times, and in each time, two **disjoint sets are chosen at random** with a predefined sizes.

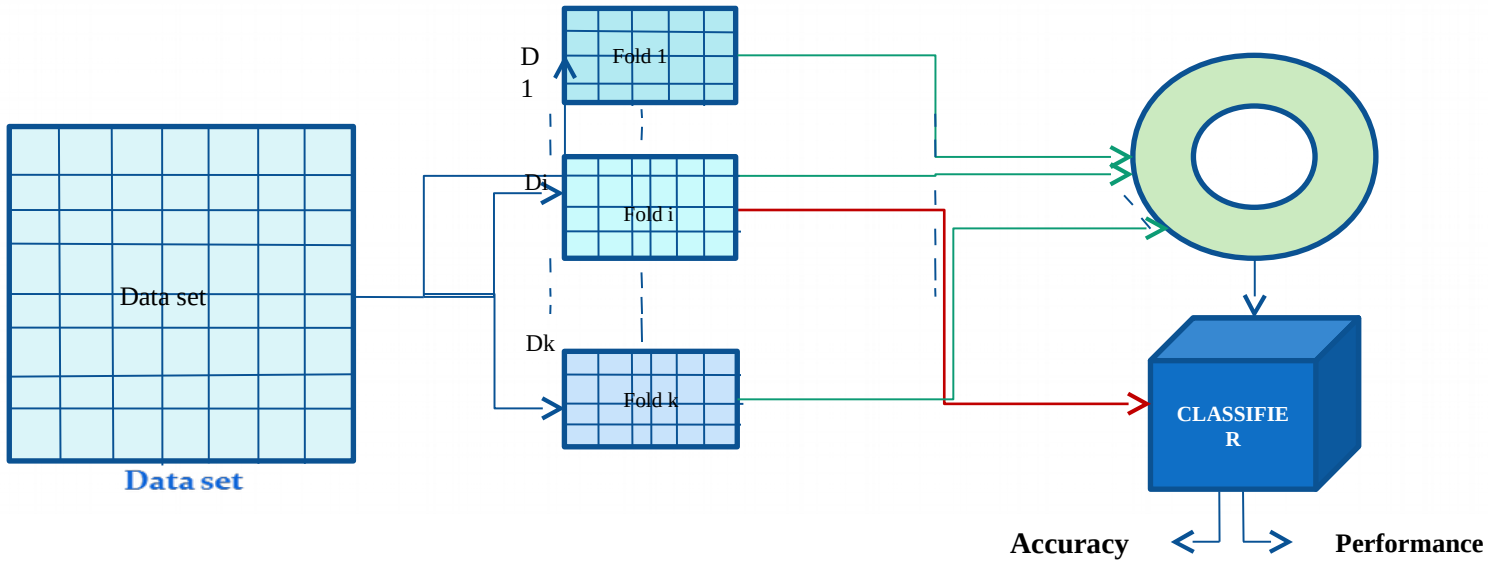
- Overall estimation is taken as the average of estimations obtained from each iteration.

Cross-Validation

- The main drawback of Random subsampling is, it does not have control over the number of times each tuple is used for training and testing.
- Cross-validation is proposed to overcome this problem.
- There are two variations in the cross-validation method.
  - k-fold cross-validation
  - *N*-fold cross-validation

k-fold Cross-Validation

- Dataset consisting of N tuples is divided into k (usually, 5 or 10) equal, mutually exclusive parts or folds $(D_1, D_2, \ldots, D_k)$, and if N is not divisible by k, then the last part will have fewer tuples than other (k-1) parts.
- A series of k runs is carried out with this decomposition, and in i^{th} iteration D_i is used as test data and other folds as training data
  - Thus, each tuple is used same number of times for training and once for testing.
- Overall estimate is taken as the average of estimates obtained from each iteration.



N-fold Cross-Validation

- In *k*-fold cross-validation method, $\frac{k-1}{N}$ part of the given data is used in training with *k*-tests.
- *N*-fold cross-validation is an extreme case of *k*-fold cross validation, often known as "Leave-one-out" cross-validation.
- Here, dataset is divided into as many folds as there are instances; thus, all most each tuple forming a training set, building *N* classifiers.
- In this method, therefore, *N* classifiers are built from *N-1* instances, and each tuple is used to classify a single test instances.
- Test sets are mutually exclusive and effectively cover the entire set (in sequence). This is as if trained by entire data as well as tested by entire data set.
- Overall estimation is then averaged out of the results of *N* classifiers.

N-fold Cross-Validation : Issue

- So far the estimation of accuracy and performance of a classifier model is concerned, the *N*-fold cross-validation is comparable to the others we have just discussed.

- The drawback of *N*-fold cross validation strategy is that it is computationally expensive, as here we have to repeat the run *N* times; this is particularly true when data set is large.

- In practice, the method is extremely beneficial with very small data set only, where as much data as possible to need to be used to train a classifier.

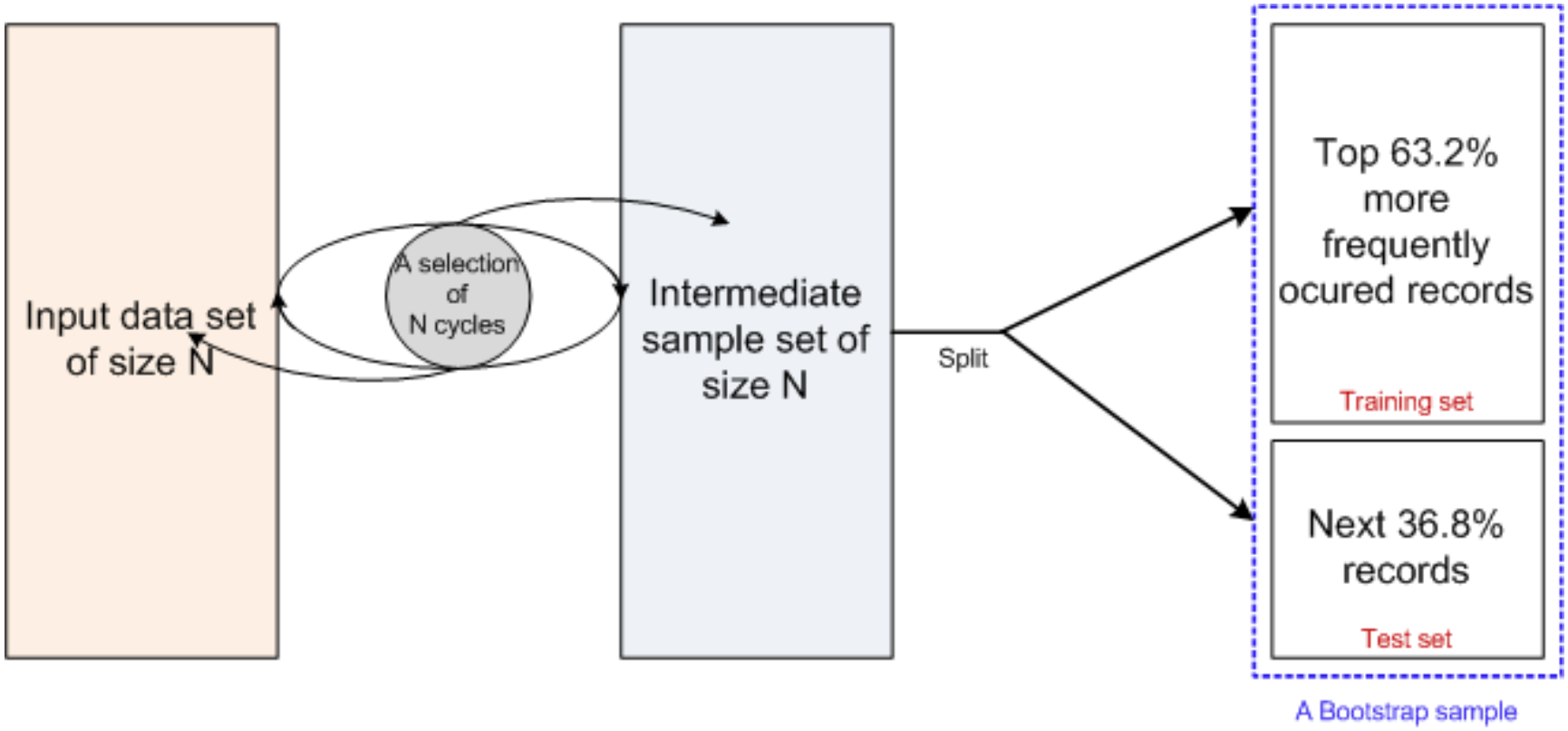
Bootstrap Method

- The Bootstrap method is a variation of repeated version of Random sampling method.
- The method suggests the sampling of training records with replacement.
  - Each time a record is selected for training set, is put back into the original pool of records, so that it is equally likely to be redrawn in the next run.
  - In other words, the Bootstrap method samples the given data set uniformly with replacement.
- The rational of having this strategy is that let some records be occur more than once in the samples of both training as well as testing.
  - What is the probability that a record will be selected more than once?

Bootstrap Method

- Suppose, we have given a data set of N records. The data set is sampled N times with replacement, resulting in a bootstrap sample (i.e., training set) of I samples.
  - Note that the entire runs are called a bootstrap sample in this method.
- There are certain chance (i.e., probability) that a particular tuple occurs one or more times in the training set
  - If they do not appear in the training set, then they will end up in the test set.
  - Each tuple has a probability of being selected $\frac{1}{N}$ (and the probability of not being selected is $\left(1-\frac{1}{N}\right)$.
  - We have to select N times, so the probability that a record will not be chosen during the whole run is $\left(1-\frac{1}{N}\right)^N$
  - Thus, the probability that a record is chosen by a bootstrap sample is $1-\left(1-\frac{1}{N}\right)^N$
  - For a large value of N, it can be proved that $\left(1-\frac{1}{N}\right)^N \approx e^{-1}$
  - Thus, the probability that a record chosen in a bootstrap sample is $1-e^{-1}=0.632$

Bootstrap Method : Implication



- This is why, the Bootstrap method is also known as 0.632 bootstrap method

Accuracy Estimation

Accuracy Estimation

- We have learned how a classifier system can be tested. Next, we are to learn the metrics with which a classifier should be estimated.
- There are mainly to things to be measured for a given classifier
  - Accuracy
  - Performance
- **Accuracy estimation**
  - If N is the number of instances with which a classifier is tested and p is the number of correctly classified instances, the accuracy can be denoted as

$$\epsilon = \frac{p}{N}$$

  - Also, we can say the error rate (i.e., is misclassification rate) denoted by $\bar{\epsilon}$ is denoted by

$$\bar{\epsilon} = 1 - \epsilon$$

Accuracy : True and Predictive

- Now, this accuracy may be true (or absolute) accuracy or predicted (or optimistic) accuracy.
- True accuracy of a classifier is the accuracy when the classifier is tested with all possible unseen instances in the given classification space.
  - However, the number of possible unseen instances is potentially very large (if it is not infinite)
  - For example, classifying a hand-written character
  - Hence, measuring the true accuracy beyond the dispute is impractical.
- Predictive accuracy of a classifier is an accuracy estimation for a given test data (which are mutually exclusive with training data).
  - If the predictive accuracy for test set is $\in$ and if we test the classifier with a different test set it is very likely that a different accuracy would be obtained.
  - The predictive accuracy when estimated with a given test set it should be acceptable without any objection

Predictive Accuracy

Example 11.1 : Universality of predictive accuracy

- Consider a classifier model *MD* developed with a training set D using an algorithm *M*.

- Two predictive accuracies when *MD* is estimated with two different training sets T1 and T2 are

(MD)T1 = 95%

(MD)T2 = 70%

- Further, assume the size of T1 and T2 are

|T1| = 100 records

|T2| = 5000 records.

Based on the above mentioned estimations, neither estimation is acceptable beyond doubt.

Predictive Accuracy

- With the above-mentioned issue in mind, researchers have proposed two heuristic measures
  - Error estimation using **Loss Functions**
  - Statistical Estimation using **Confidence Level**
- In the next few slides, we will discus about the two estimations

Error Estimation using Loss Functions

- Let T be a matrix comprising with N test tuples

$$\begin{bmatrix} X_1 & y_1 \\ X_2 & y_2 \\ & \\ X_N & y_N \end{bmatrix}_{N\times(n+1)}$$

where X_i ($i = 1, 2, \ldots, N$) is the n-dimensional test tuples with associated outcome y_i.

- Suppose, corresponding to (X_i, y_i), classifier produces the result (X_i, y_i')
- Also, assume that $(y_i - y_i')$ denotes a difference between y_i and y_i' (following certain difference (or similarity), (e.g., $(y_i - y_i') = 0$, if there is a match else 1)
- The two loss functions measure the error between y_i (the actual value) and y_i' (the predicted value) are

Absolute error: $|y_i - y_i'|$

Squred error: $|y_i - y_i'|^2$

Error Estimation using Loss Functions

- Based on the two loss functions, the test error (rate) also called generalization error, is defined as the average loss over the test set T. The following two measures for test errors are

  Mean Absolute Error (MAE): $\frac{\sum_{i=1}^{N} |y_i - y_i'|}{N}$

  Mean Squared Error(MSE):): $\frac{\sum_{i=1}^{N} (y_i - y_i')^2}{N}$

- Note that, MSE aggregates the presence of outlier.
- In addition to the above, a relative error measurement is also known. In this measure, the error is measured relative to the mean value $\tilde{y}$ calculated as the mean of y_i ($i = 1, 2, \ldots, N$) of the training data say D. Two measures are

  Relative Absolute Error (RAE: $\frac{\sum_{i=1}^{N} |y_i - y_i'|}{\sum_{i=1}^{N} |y_i - \tilde{y}|}$

  Relative Squared Error (RSE): $\frac{\sum_{i=1}^{N} (y_i - y_i')^2}{\sum_{i=1}^{N} (y_i - \tilde{y})^2}$

Statistical Estimation using Confidence Level

- If we know the value of predictive accuracy, then we can guess the true accuracy within a certain range given a **confidence level**.

Example

- When a coin is tossed, there is a probability that the head will occur. We have to experiment the value for this probability value. A simple experiment is that the coin is tossed many times and both numbers of heads and tails are recorded.

N=10		N=50		N=100		N=250		N=500		N=1000	
H	T	H	T	H	T	H	T	H	T	H	T
3	7	29	21	54	46	135	115	241	259	490	510
0.30	0.70	0.58	0.42	0.54	0.46	0.54	0.46	0.48	0.42	0.49	0.51

- Thus we can say p-->0.5 after a large number of trials in each experiment