**Lecture 3 White board notes**

## Partitioning techniques

### 10-fold Cross validation – Pictorial representation

Iteration #1

Training

Training

Testing

Training

.

.

.

.

Iteration #10

Training

Testing

Training

Training

### Bootstrap

Utilizes sampling with replacement. If there are n cases in a dataset, then

Probability of selecting any one instance = 1/n

Probability of not selecting any one instance = 1 – 1/n

Probability of a case ending up in test dataset =

Probability of an instance to be picked in training dataset = 0.632

Example:

|  |  |  |
| --- | --- | --- |
| Original | Training | Testing |
| Case 1 | Case 1 | Case 2 |
| Case 2 | Case 1 | Case 3 |
| Case 3 | Case 4 | Case 3 |
| … | … | … |
| Case n | Case n-5 | Case n |

Estimating error with bootstrap

* Error on the testing data will be pessimistic. Hence it has to be combined with training error:

## Comparing data mining algorithms

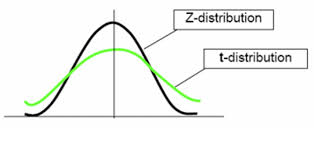
Let's assume there are two algorithms A and B. Their accuracies across samples are given below:

|  |  |  |
| --- | --- | --- |
|  | Classifier A | Classifier B |
| Sample 1 (S1) |  |  |
| Sample 2 (S2) |  |  |
|  |  |  |
| Sample k (Sk) |  |  |
| Mean |  |  |

Null hypothesis:

Alternate hypothesis:

Paired t-test:

[](https://www.google.com/imgres?imgurl=http://1.bp.blogspot.com/-lxGyIVEGSsA/VE_xinGXgmI/AAAAAAAABG0/878M74FWpTc/s1600/t%2Bdistribution.png&imgrefurl=http://econospeak.blogspot.com/2014_10_01_archive.html&h=460&w=900&tbnid=OYuT47cWXEpANM:&docid=1B0TQXGlLF2bpM&ei=QAqnVtSQBsPijgSgh7a4DA&tbm=isch&ved=0ahUKEwjUsKKh5cbKAhVDsYMKHaCDDccQMwhzKEwwTA)

If k = 10, it follows t distribution with 9 degrees of freedom.

For 0.1% confidence, Z value is 4.30 for t-distribution and 3.09 for normal distribution

If ) < 0.05, then reject null hypothesis and accept that accuracy for Classifier A is significantly better than accuracy for Classifier B.

## Performance evaluation for probabilistic models

#### 0 -1 Loss function

Loss is 0 for a correct prediction and 1 for an incorrect one

|  |  |  |  |
| --- | --- | --- | --- |
|  | Actual | Predicted | Loss |
| Case 1 | Class 1 | Class 1 | 0 |
| Case 2 | Class 1 | Class 2 | 1 |
| … | … | … | … |
| Case n | Class 2 | Class 2 | 0 |

#### Quadratic loss function

|  |  |  |  |
| --- | --- | --- | --- |
|  | Actual | Predicted | |
| P1 | P2 |
| Case 1 | Class 1 |  |  |
| Case 2 | Class 1 |  |  |
|  |  |  |  |
| Case n |  |  |  |

Quadratic loss function for each case = where

P1 – probability that case belongs to Class C1

P2 – probability that case belongs to Class C2

… = 0, except for ac which is 1

*c* is the index of the instance’s actual class

For a two class problem: Quadratic loss for each case =

If instance belongs to Class 1, then = 1 and 🡺

Overall accuracy =

Where i = index for no of cases

j = index for number of classes

On solving the Quadratic loss becomes. This loss function has to be minimized.

#### Informational loss

Expected value for loss function =

## Cost sensitive evaluation

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Predicted | |
|  |  | C1 (+ class) | C2 (- class) |
| Actual | C1 (+ class) | True Positive (TP) | False Negative (FN) |
| C2 (- class) | False Positive (FP) | True Negative (TN) |

Accuracy = TP + TN / TP + TN + FP + FN

Cost matrix example:

|  |  |  |  |
| --- | --- | --- | --- |
| Cost Matrix | | Predicted | |
| C1 (+ class) | C2 (- class) |
| Actual | C1 (+ class) | -1 | 100 |
| C2 (- class) | 1 | 0 |

Misclassification matrix Example:

|  |  |  |  |
| --- | --- | --- | --- |
| Model M1 | | Predicted | |
| C1 (+ class) | C2 (- class) |
| Actual | C1 (+ class) | 150 | 40 |
| C2 (- class) | 60 | 250 |

Accuracy = 80%, Cost = (40\*100) + 60\*1 + (150\*-1) + 0 = **3910**

|  |  |  |  |
| --- | --- | --- | --- |
| Model M2 | | Predicted | |
| C1 (+ class) | C2 (- class) |
| Actual | C1 (+ class) | 250 | 45 |
| C2 (- class) | 5 | 200 |

Accuracy = 90%, Cost = 45\*100 + 5\*1 + (250\*-1) + 0 = **4255**

## Cost Sensitive Learning

1. Modify inputs to the algorithm to reflect the cost
   1. Resampling
   2. Associate weights to data
2. Modify the learning algorithm to incorporate costs (Cost sensitive boosting)
   1. Adaboost

#### Lift Chart

1,000,000 households

0.1% response rate = 1000 people respond

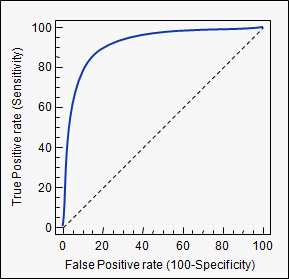
Data mining tool will find 100000 potential responders out of which 0.4% (400 people) will respond

Identify a subset of 400,000 potential responders out of which 0.2% (800 people) will respond

#### ROC Curve

Sensitivity = True Positives/Total Positives

Specificity = True Negatives/Total Negatives

[](https://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=0ahUKEwi6y8mDh8fKAhXns4MKHe4dAVwQjRwIBw&url=https://www.medcalc.org/manual/roc-curves.php&psig=AFQjCNEh_G7n5que1CxjX1JAe4f_1A5j6w&ust=1453883169285292)