

**Master of Technology in**

Knowledge Engineering

**Computational Intelligence I**

Bharat Nagaraju - A0178258N

Vigneshram Selvaraj – A0178215A

Laxman Singh – A0178223E

Classification of Mail

To detect if a given mail is a Spam.

# Introduction

*Neural Network* is an advanced Machine Learning Technique which has the ability to mimic Human brain. They can learn by both supervised and unsupervised methods. They can be used to solve both Classification and Regression problems.

In statistics and machine learning, ensemble methods use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone. Unlike a statistical ensemble in statistical mechanics, which is usually infinite, a machine learning ensemble consists of only a concrete finite set of alternative models, but typically allows for much more flexible structure to exist among those alternatives

# Data Understanding

The "spam" concept is diverse: advertisements for products/websites, make money fast schemes, chain letters, online lottery scam etc.

Our collection of spam e-mails came from our postmaster and individuals who had filed spam. Our collection of non-spam e-mails came from filled work and personal e-mails, and hence the word 'george' and the area code '650' are indicators of non-spam. These are useful when constructing a personalized spam filter. One would either have to blind such non-spam indicators or get a very wide collection of non-spam to generate a general purpose spam filter.

The last column of 'spambase.data' denotes whether the e-mail was considered spam (1) or not (0), i.e. unsolicited commercial e-mail. Most of the attributes indicate whether a particular word or character was frequently occurring in the e-mail. The run-length attributes (55-57) measure the length of sequences of consecutive capital letters. For the statistical measures of each attribute, see the end of this file. Here are the definitions of the attributes:

48 continuous real [0,100] attributes of type word\_freq\_WORD

= percentage of words in the e-mail that match WORD, i.e. 100 \* (number of times the WORD appears in the e-mail) / total number of words in e-mail. A "word" in this case is any string of alphanumeric characters bounded by non-alphanumeric characters or end-of-string.

6 continuous real [0,100] attributes of type char\_freq\_CHAR]

= percentage of characters in the e-mail that match CHAR, i.e. 100 \* (number of CHAR occurrences) / total characters in e-mail

1 continuous real [1,...] attribute of type capital\_run\_length\_average

= average length of uninterrupted sequences of capital letters

1 continuous integer [1,...] attribute of type capital\_run\_length\_longest

= length of longest uninterrupted sequence of capital letters

1 continuous integer [1,...] attribute of type capital\_run\_length\_total

= sum of length of uninterrupted sequences of capital letters

= total number of capital letters in the e-mail

1 nominal {0,1} class attribute of type spam

= denotes whether the e-mail was considered spam (1) or not (0), i.e. unsolicited commercial e-mail.

Data Splitting

The entire dataset is split into 75% Training and 25% Testing set.

Data Normalization

Normalization is done so that a variable which is on a higher scale does not affect the outcome just because it is on a higher scale. For example consider a credit card dataset having two variables credit cards and income and you intend to cluster records to find similar applicants based on these attributes. As we can well imagine these two will be on different scales and income being on a much higher scale will influence the distance measures much more than credit cards. Thus normalization is done to avoid this.

Data Modelling

**1) Multilayer Feed Forward with Backward Propagation**

Confusion Matrix

|  |  |  |
| --- | --- | --- |
|  | Predicted |  |
| Actual | 0 | 1 |
| 0 | 666 | 32 |
| 1 | 42 | 410 |

Accuracy

93.56

**2) Multilayer Feed Forward with Backward Propagation with Normalized data**

Confusion Matrix

|  |  |  |
| --- | --- | --- |
|  | Predicted |  |
| Actual | 0 | 1 |
| 0 | 666 | 48 |
| 1 | 29 | 407 |

Accuracy:

93.30

**3)Radial Basis Function**

Confusion Matrix

|  |  |  |
| --- | --- | --- |
|  | Predicted |  |
| Actual | 0 | 1 |
| 0 | 454 | 249 |
| 1 | 88 | 359 |

Accuracy

70.69

**4)Multinomial Logistic Regression**

Confusion Matrix :

|  |  |  |
| --- | --- | --- |
|  | Predicted |  |
| Actual | 0 | 1 |
| 0 | 669 | 30 |
| 1 | 54 | 387 |

Accuracy :

0.9269565

**Ensemble 1**

Ensemble model is built using

1) Multilayer feed forward with Backward Propagation with non-normalized data

2) Multilayer feed forward with Backward Propagation with normalized data

3) Radial Basis Function with non-normalized data.

|  |  |  |
| --- | --- | --- |
|  | Predicted |  |
| Actual | 0 | 1 |
| 0 | 443 | 259 |
| 1 | 256 | 192 |

Accuracy:

0.5521739

**Ensemble 2**

Ensemble model is built using

1) Multilayer feed forward with Backward Propagation with non-normalized data

2) Multilayer feed forward with Backward Propagation with normalized data

3) Radial Basis Function with non-normalized data.

4)Multinomial Logistic Regression

Confusion Matrix:

|  |  |  |
| --- | --- | --- |
|  | Predicted |  |
| Actual | 0 | 1 |
| 0 | 434 | 268 |
| 1 | 149 | 299 |

Accuracy:

0.6373913

Conclusion

1) The individual networks are performing well than the Ensemble

2) The Best among the individual networks is Multi Layer Feed Forward with Backward Propagation

3) Ensemble 2 has performed better than Ensemble 1 because of diversity of models in Ensemble 2. Ensemble 2 has Logistic regression included in the Ensemble apart from Neural Networks.