Intrusion Detection using LDA and Random Forests on KDD CUP 99 Dataset

Girish Palya

Overview

Two wildly different learning algorithms are applied on a multi-class classification problem involving a dataset with a number of categorical and numeric variables, and performance is compared. *Linear Discriminant Analysis (LDA)* is used after selecting features through dimensional reduction (employing Principal Component Analysis), and *Random Forests* is used after selecting variables based on mean decrease of Gini coefficient. Random Forests outperformed LDA by a significant margin, even with sub-optimal number of trees.

Dataset Description

KDD CUP 99 is a particularly challenging dataset with massive size, redundancy, a large number of variables (including both numeric and categorical), and a skewed target variable. It is a popular dataset used for intrusion detection in academic literature. The dataset was originally created in 1999 at MIT's Lincoln Laboratory during a DARPA sponsored event, where a number of attack scenarios were simulated and features were extracted.

Training dataset has 4,898,431 observations. However, due to high redundancy (78%) only 1,074,992 are unique data points. Test dataset has 311,029 examples.

Problem involves *classifying test examples into one of 4 network attack categories and a non-attack ("normal") category*.

Features

There are 41 variables, 34 numeric and 7 categorical (nominal). One of the nominal variables (“service”) has high cardinality (66 classes). Distribution of numeric variables is closer to normal. However, some of the variables are highly correlated.

Target Classes

Dataset has five classes for target variable: Four types of attacks and a type for normal connection. Attacks are classified as follows:

- DOS: denial-of-service, e.g., SYN flood attack
- R2L: unauthorized access from a remote machine, e.g. guessing password
- U2R: unauthorized access to local superuser (root) privileges, e.g., various buffer overflow attacks
- Probing: surveillance and other types of probing, e.g., port scanning

Data Preparation

Following transformations are applied on training and test data:

- Remove redundant rows
- Remove variables that have all 0 values
- Convert strings to factors (for nominal variables)

Evaluation Metric

Since this is a multi-class classification problem, weighted F1 score is chosen as the analysis metric.

F1 score is the harmonic mean of precision and recall.

$$F1 = \frac{2}{recall^{-1} + precision^{-1}} = \frac{2.precision.recall}{precision + recall}$$

Weighted F1 score is the average of F1 scores over all classes of target variable, weighed proportional to the frequency of occurrence of target class (aka “support”).

$$Weighted\ F1 = \frac{1}{n} \sum_{i=1}^{m} Support_i.F1_i$$

Linear Discriminant Analysis (LDA)

LDA is a natural choice for multi-class classification. Unlike logistic regression which does binary classification by directly modelling conditional distribution of response variable, LDA takes an alternate approach based on Bayes' theorem. Here the distribution of the predictors is modeled (instead of distribution of response variable) for each response class, and then flipped around using Bayes' theorem (to get conditional probability of response variable). This means predictions for multiple classes of response variable is obtained in one step.

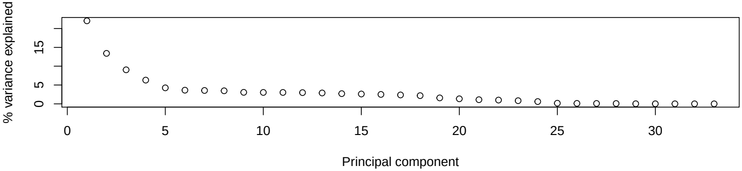
Before applying LDA,

- redundant continuous variables are eliminated using dimension reduction
- nominal variables are converted into dummy variables and redundancy is also eliminated among dummy variables.

Feature Selection using PCA

Dimensionality reduction is a process of reducing the dimensions of a matrix while still preserving a good amount of information. *Principal component analysis (PCA)* is a popular technique used in dimensionality reduction. The idea is as follows: We think of rows of a matrix as vectors representing points in Euclidean space. So, a (m x n) matrix will have m points in a n-dimensional space. We can “rotate” the axes of this space in a such a way that the first axis (“x” axis) is oriented along the direction that yields the maximum variance of values of the coordinates of original points. Similarly, second axis (chosen to be orthogonal to the first) is in a plane that yields second highest variance, and so on. If this process is repeated, we will likely hit a plateau where subsequent axes capture only a small amount of variance (“information”). We can drop these less significant axes, thereby reducing the dimensions of our matrix (and size of our dataset).

Contd.


% variance explained
0
5
15
0
5
10
15
20
25
30
Principal component

Contd.

Before considering nominal variables, benchmark LDA's classification performance with only numeric variables included in the model.

Weighted F1 Score for model with only numeric variables is 39.4.

Dummy variables

Nominal variable `service` has very high cardinality (66 classes). Encoding 65 dummy variables would result in a wide and sparce input matrix. For pragmatic reasons this variable is dropped.

Following table shows total number of classes in each nominal variable.

```
##    protocol_type service flag land logged_in is_gu
## 1:             3      66   11    2         2
```

Training LDA

We fit LDA model on the full training set (with PCs from both continuous and dummy variables). Confusion matrix is as shown below.

```
##            Reference
## Prediction    DoS normal  Probe    R2L    U2R
##     DoS     58629    113    242    110      1
##     normal 169462  59792    367  15440     22
##     Probe     721    496   3557    123      2
##     R2L      1043    169      0    663     17
##     U2R         0     23      0      9     28
```

Contd.

Weighted F1 score (below) shows no improvement over a LDA model fitted on only PCAs from numeric variables. This may be expected. Discriminant analysis assumes a normal distribution of dependent variables. When categorical variables are encoded as integer values of 0 and 1 (values of dummy variables) the resulting distribution is neither normal nor multivariate, and therefore LDA handles them poorly.

```
##                            Model Weighted F1 Score
## 1: LDA: Only numeric variables              39.4
## 2:          LDA: All variables              39.1
```

Random Forests

Decision trees can be constructed by recursively dividing (binary splitting) the predictor space into distinct and non-overlapping regions until a termination criteria is reached. For every observation that falls into a region (leaf), we make the same prediction, which (for classification problems) is simply the *most commonly occurring class* of training observations in the region. For classification trees, the criteria for binary splits is the *Gini index* given by

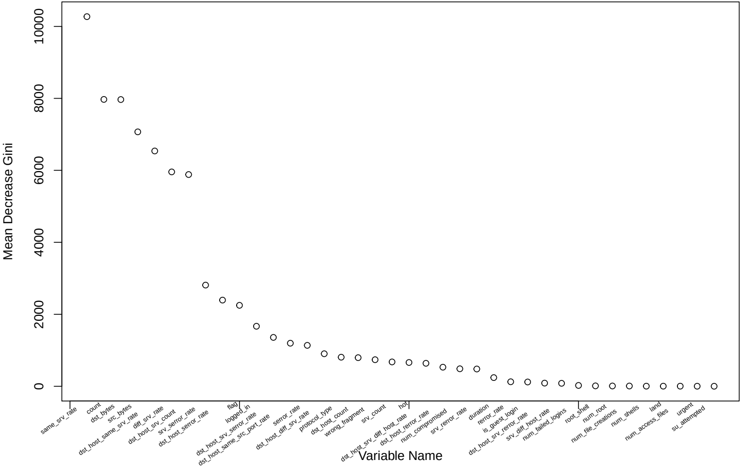
$$G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk}),$$

Feature Selection using Gini index

In a collection of bootstrapped classification trees we can obtain the importance of a variable using *Gini index*. We can add the total amount that the *Gini index* is decreased by splits over a given predictor, averaged over all trees. Mean decrease in *Gini index* for each variable (relative to the largest) represents its *importance*. We can remove redundant variables after ranking all variables in the order of their *importance*.

After ranking variables based on *mean reduction Gini*, *11 variables are removed*.

Contd.


Mean Decrease Gini
0
2000
4000
6000
8000
10000
Variable Name
same_srv_rate
count
dst_bytes
src_bytes
dst_host_same_srv_rate
diff_srv_rate
dst_host_srv_count
srv_serror_rate
dst_host_serror_rate
flag
logged_in
dst_host_srv_serror_rate
dst_host_same_src_port_rate
serror_rate
dst_host_diff_srv_rate
protocol_type
dst_host_count
wrong_fragment
srv_count
hot
dst_host_srv_diff_host_rate
dst_host_rerror_rate
num_compromised
srv_rerror_rate
duration
rerror_rate
is_guest_login
dst_host_srv_rerror_rate
srv_diff_host_rate
num_failed_logins
root_shell
num_root
num_file_creations
num_shells
land
num_access_files
urgent
su_attempted

Training Random Forests

Number of trees is chosen to be 100, and it gives a reasonably good result. More trees will increase the weighted F1 score at the expense of time needed to compute.

Both confusion matrix and weighted F1 score shows marked improvement over LDA.

```
##            Reference
## Prediction    DoS normal  Probe    R2L    U2R
##     DoS    223768     67    207      0      0
##     normal   6051  60298    664  15954     60
##     Probe      36    226   3295     21      0
##     R2L         0      1      0    369      4
##     U2R         0      1      0      1      6
```

Results

Random Forests outperforms *LDA* by a wide margin. Presence of high cardinality categorical variables does not adversely affect the performance of *Random Forests*. However, it takes longer to train compared to *LDA*. *LDA* fails to take advantage of categorical variables. It suffers from being unable to deal with the predominance of outliers in the distribution of dummy variables.

```
##    Training Algorithm Weighted F1 Score
## 1:                LDA              39.1
## 2:     Random Forests              90.6
```