**EE219 Project 1**

**Classification Analysis on Textual Data**

Jiahui Lu - 204945099

Chenguang Yuan - 005030313

1. **Introduction**

This project analyzes a data set – *20 Newsgroups*. In this project, different methods are implemented so as to classify textual data. (Basic work outline… to be continued…)

The 20 Newsgroups data set is a collection of approximately 20,000 newsgroup documents which is partitioned (nearly) evenly across 20 different newsgroups. The data is organized into 20 different newsgroups, each corresponding to a different topic. Some of the newsgroups are very closely related to each other (e.g. *comp.sys.ibm.pc.hardware* / *comp.sys.mac.hardware*), while others are highly unrelated (e.g *misc.forsale* / *soc.religion.christian*). Table 1 is a list of the 20 newsgroups, partitioned (more or less) according to subject matter:

Table 1. Data Organization

|  |  |
| --- | --- |
| *comp.graphics*  *comp.os.ms-windows.misc*  *comp.sys.ibm.pc.hardware*  *comp.sys.mac.hardware*  *comp.windows.x* | *rec.autos*  *rec.motorcycles*  *rec.sport.baseball*  *rec.sport.hockey* |
| *talk.politics.misc*  *talk.politics.guns*  *talk.politics.mideas* | *sci.crypt*  *sci.electronics*  *sci.med*  *sci.space* |
| *talk.religion.misc*  *alt.atheism*  *soc.religion.christian* | *misc.forsale* |

(classification) In this case, statistical classification is applied to finish our project, which refers to the task of identifying a category, from a predefined set, to which a data point belongs, given a training data set with known category memberships. Moreover, classification differs from the task of clustering, which concerns grouping data points with no predefined category memberships, where the objective is to seek inherent structures in data with respect to suitable measures. Classification turns out as an essential element of data analysis, especially when dealing with a large amount of data.

1. **20 Newsgroup Data Set Overview**

The features captured in data set are as follows:

1. **DESCR**. This is NoneType of built-in modules

2. **Data**. It consists of the email data

3. **Description**. Just the data set.

4. **Filenames**. ndArray object of numpy module

5. **Target**. Array of the data set size.

6. **Target\_names**. The name of the target.

For the purposes of this project we will be working with only 8 of the classes as shown in Table 2. Load the training and testing data for the following 8 subclasses of two major classes ‘Computer Technology’ and ‘Recreational activity’.

Table 2. Subclasses of ‘Computer technology’ and ‘Recreational activity’

|  |  |
| --- | --- |
| Computer Technology | Recreational Activity |
| *comp.graphics*  *comp.os.ms-windows.misc*  *comp.sys.ibm.pc.hardware*  *comp.sys.mac.hardware* | *rec.autos*  *rec.motorcycles*  *rec.sport.baseball*  *rec.sport.hockey* |

In a classification problem, the imbalance in the relative sizes of the data sets corresponding to different classes should be taken into account so as to make sure the classification is handled properly. To do so, one can either modify the penalty function (i.e. assign more weight to errors from minority classes), or alternatively, down-sample the majority classes, to have the same number of instances as minority classes. The plot of histogram of the number of training documents per class of following to the data of the 8 classes above is showing to check whether the they are evenly distributed. Note that the data set is already balanced and so in this case we do not need to balance. As a result, this 8 class data set has balanced data length as show in the Figure 1.

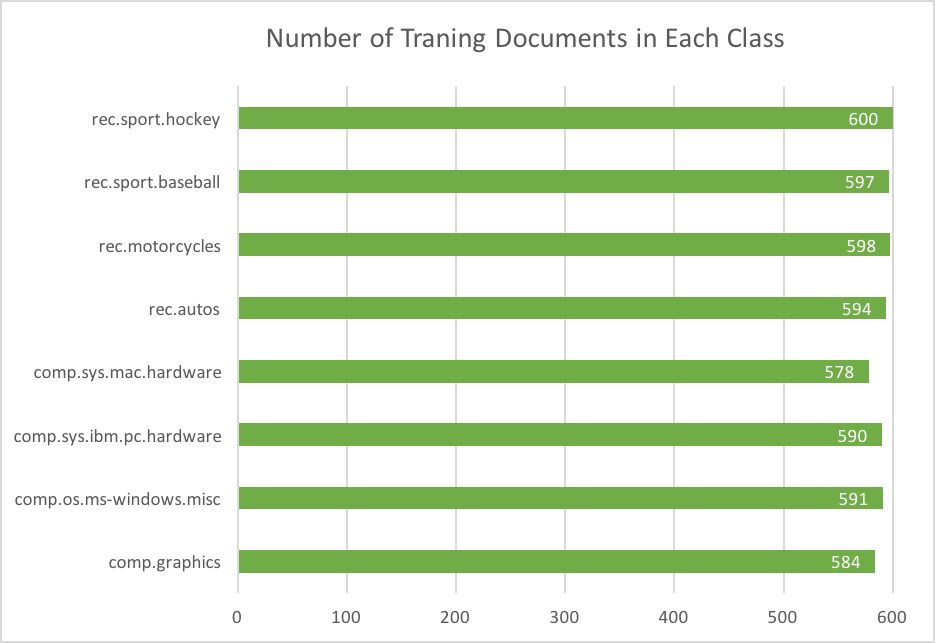


Figure 1. Number of Training Document in Each Class

1. **Data Preprocessing**
   1. Modeling Text Data

The first step is to do the data preprocessing. a common sense filtering is done to drop certain words or terms: To avoid unnecessarily large feature vectors (vocabulary size), terms that are too frequent in almost every document, or are very rare, are dropped out of the vocabulary. Here, we use the “WordNetLemmatizer” to do the lemmatize so as to return the words back to stem. Then we applied the stop words, which acts as a filter to get rid of the words that are too common.

After we first tokenize each document into words and exclude the stop words, punctuations as well as using stemmed version of words, the TFxIDF vector representations can be applied.

* 1. Feature Extraction with TFxIDF

Next, considering using the normalized count of the vocabulary words in each document to build representation vectors.

where 𝑡𝑓(𝑡,𝑑) represents the frequency of term 𝑡 in document d, and inverse document frequency is defined as:

where 𝑛 is the total number of documents, and 𝑑𝑓(𝑡) is the document frequency; the document frequency is the number of documents that contain the term 𝑡.

Here, we have two settings when it comes to the minimum document frequency, which acts also as a filter to “stop” the words that show up less often. It is trivial that the number of setting is smaller, the number of terms will be larger. The result shows that for each class in two settings.

Table 3. Number of terms for min\_df = 2

|  |  |
| --- | --- |
| *comp.graphics* | 2661 |
| *comp.os.ms-windows.misc* | 4633 |
| *comp.sys.ibm.pc.hardware* | 2292 |
| *comp.sys.mac.hardware* | 2173 |
| *rec.autos* | 2729 |
| *rec.motorcycles* | 2763 |
| *rec.sport.baseball* | 2668 |
| *rec.sport.hockey* | 3250 |
| *Total* | 23169 |

Table 4. Number of terms for min\_df = 5

|  |  |
| --- | --- |
| *comp.graphics* | 995 |
| *comp.os.ms-windows.misc* | 1543 |
| *comp.sys.ibm.pc.hardware* | 869 |
| *comp.sys.mac.hardware* | 850 |
| *rec.autos* | 1158 |
| *rec.motorcycles* | 1215 |
| *rec.sport.baseball* | 1162 |
| *rec.sport.hockey* | 1310 |
| *Total* | 9102 |

* 1. Top 10 Words with TFxICF

Now we can look up into the whole document to see the top 10 significant words. In order to do so, a measure called TFxICF should be introduced. It is the same as TFxIDF, except that a class sits in place of a document; that is for a term *t* and a class *c*, the measure is computed as

where 𝑡𝑓(𝑡,𝑐) represents the term frequency in a class 𝑐 , and inverse class frequency is defined as:

and 𝑛𝑐𝑙𝑎𝑠𝑠𝑒𝑠 is the total number of classes, and 𝑐𝑓(𝑡) is the class frequency which is the number of classes within which there is at least a document containing the term 𝑡.

Let’s have a look at the results using the the unbalanced dataset of all 20 classes for this part respect to TFxICF measure.

Table 5. Top 10 Words

|  |  |
| --- | --- |
| **comp.sys.ibm.pc.hardwa** | **comp.sys.mac.hardware** |
| *'monitor',*  *'eisa',*  *'vlb',*  *'motherboard',*  *'floppy',*  *'isa',*  *'bios',*  *'controller',*  *'ide',*  *'scsi'* | *'macs',*  *'modem',*  *'simm',*  *'vram',*  *'ram',*  *'lc',*  *'fpu',*  *'monitor',*  *'simms',*  *'scsi'* |

Table 6. Top 10 Words

|  |  |
| --- | --- |
| **misc.forsale** | **soc.religion.christian** |
| *'sale',*  *'new',*  *'offer',*  *'discount',*  *'manuals',*  *'money',*  *'price',*  *'sell',*  *'use',*  *'forsale'* | *'belief',*  *'religion', 'resurrection', 'catholic',*  *'sin',*  *'scripture', 'christianity',*  *'bible',*  *'christ',*  *'christians'* |

* 1. Feature Selection

In this part, LSA and NMF are applied to reduced to dimension.

First apply LSI to the TFxIDF matrix corresponding to the 8 classes picking k=50. After this each document is mapped to a 50-dimensional vector.

By selecting a non-sparse subset of the total feature to reduce the dimensionality using mean-squared errors, it minimizes mean squared residual between the original data and reconstruction from its low-dimensional approximation. The LSI representation is obtained by computing left and right singular vectors corresponding to the largest values of the term-document TFxIDF matrix.

Also Non-Negative Matrix Factorization (NMF) will also provide a way of dimensionality reduction. The difference of these two method will be compared in the next section of learning algorithm.

1. **Learning Algorithms**
   1. Support Vector Machines

4.1.1 Hard Margin SVM classifier

* + - 1. Latent Semantic Indexing (LSI) Method:

(A) Minimum Document Frequency of Vocabulary Terms (min\_df) = 2

Linear SVM classifiers work by interpreting the sign of the vector representation of the document multiplied by a weight. A positive sign means a document belongs to one class, while a negative sign means it belongs to the other class. We used a linear kernel to train out classifier. We then ran it on our training dataset. The statistics obtained for the classifier are:

Table 7. Statistics--Hard Margin SVC using LSI, min\_df=2

|  |  |
| --- | --- |
| Statistic | Result |
| Accuracy | 0.971746 |
| Precision | 0.966999 |
| Recall | 0.977344 |

Table 8. Confusion Matrix--Hard Margin SVC using LSI, min\_df=2

|  |  |  |
| --- | --- | --- |
|  | Predicted Computer Technology | Predicted Recreational Activity |
| Actual Computer Technology | 1508 | 53 |
| Actual Recreational Activity | 36 | 1553 |

We also plotted a ROC (Receiver Operating Characteristic) curve, to observe the trade-off between the two components of the predictions. We plotted the probabilities of true positives versus the false positives.

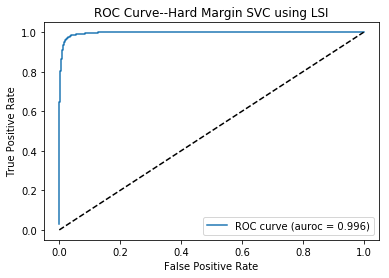


Figure 2. ROC curve--Hard Margin SVC using LSI, min\_df=2

The ROC we obtained shows all the classes have an area approximately equal to 1. Thus, we can safely say that all our test cases are correctly classified.

(B) Minimum Document Frequency of Vocabulary Terms (min\_df) = 5

Linear SVM classifiers work by interpreting the sign of the vector representation of the document multiplied by a weight. The statistics obtained for the classifier are:

Table 9. Statistics--Hard Margin SVC using LSI, min\_df=5

|  |  |
| --- | --- |
| Statistic | Result |
| Accuracy | 0.971429 |
| Precision | 0.966978 |
| Recall | 0.976715 |

Table 10. Confusion Matrix--Hard Margin SVC using LSI, min\_df=5

|  |  |  |
| --- | --- | --- |
|  | Predicted Computer Technology | Predicted Recreational Activity |
| Actual Computer Technology | 1508 | 53 |
| Actual Recreational Activity | 37 | 1552 |

We also plotted a ROC (Receiver Operating Characteristic) curve, to observe the trade-off between the two components of the predictions. We plotted the probabilities of true positives versus the false positives.

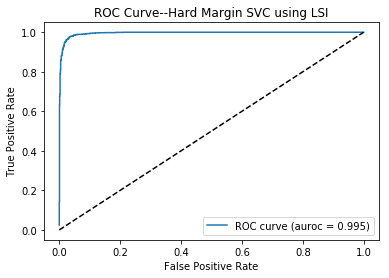


Figure 3. ROC curve--Hard Margin SVC using LSI, min\_df=5

The ROC we obtained shows all the classes have an area approximately equal to 1. Thus, we can safely say that all our test cases are correctly classified.

By comparing the results, we can see that both the settings that min\_df=2 and min\_df=5 did a good job in the Hard Margin SVC using LSI.

4.1.1.2 Non-Negative Matrix Factorization (NMF), min\_df=2:

Table 11. Statistics--Hard Margin SVC using NMF

|  |  |
| --- | --- |
| Statistic | Result |
| Accuracy | 0.963175 |
| Precision | 0.957738 |
| Recall | 0.969792 |

Table 12. Confusion Matrix--Hard Margin SVC using NMF

|  |  |  |
| --- | --- | --- |
|  | Predicted Computer Technology | Predicted Recreational Activity |
| Actual Computer Technology | 1493 | 68 |
| Actual Recreational Activity | 48 | 1541 |

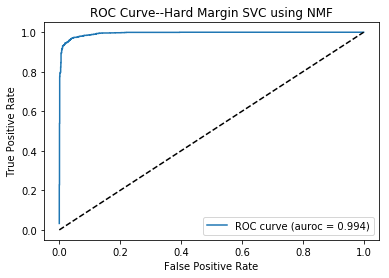


Figure 4. ROC curve--Hard Margin SVC using NMF

* + 1. Soft Margin SVM classifier

In this task we used Soft Margin SVM, to classify the documents. (gamma=0.001)

* + - 1. LSI:

1. min\_df=2

Table 13. Statistics--Soft Margin SVC using LSI, min\_df=2

|  |  |
| --- | --- |
| Statistic | Result |
| Accuracy | 0.504444 |
| Precision | 0.504444 |
| Recall | 1.000000 |

Table 14. Confusion Matrix--Soft Margin SVC using LSI, min\_df=2

|  |  |  |
| --- | --- | --- |
|  | Predicted Computer Technology | Predicted Recreational Activity |
| Actual Computer Technology | 0 | 1561 |
| Actual Recreational Activity | 0 | 1589 |

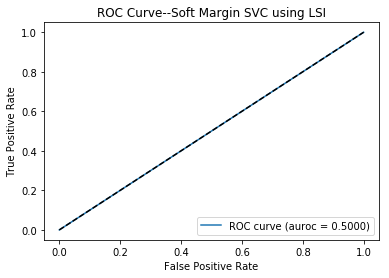


Figure 5. ROC curve--Soft Margin SVC using LSI, min\_df=2

1. min\_df=5

Table 15. Statistics--Soft Margin SVC using LSI, min\_df=5

|  |  |
| --- | --- |
| Statistic | Result |
| Accuracy | 0.504444 |
| Precision | 0.504444 |
| Recall | 1.000000 |

Table 16. Confusion Matrix--Soft Margin SVC using LSI, min\_df=5

|  |  |  |
| --- | --- | --- |
|  | Predicted Computer Technology | Predicted Recreational Activity |
| Actual Computer Technology | 0 | 1561 |
| Actual Recreational Activity | 0 | 1589 |

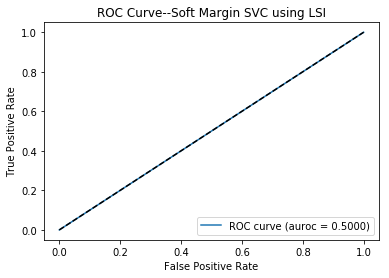


Figure 6. ROC curve--Soft Margin SVC using LSI, min\_df=5

4.1.2.2 NMF, min\_df=2:

Table 17. Statistics--Soft Margin SVC using NMF

|  |  |
| --- | --- |
| Statistic | Result |
| Accuracy | 0.504444 |
| Precision | 0.504444 |
| Recall | 1.000000 |

Table 18. Confusion Matrix--Soft Margin SVC using NMF

|  |  |  |
| --- | --- | --- |
|  | Predicted Computer Technology | Predicted Recreational Activity |
| Actual Computer Technology | 0 | 1561 |
| Actual Recreational Activity | 0 | 1589 |

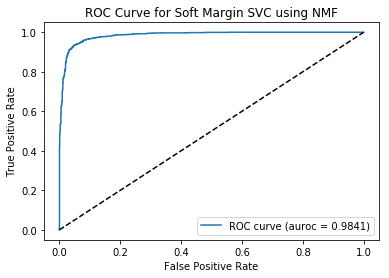


Figure 7. ROC curve--Soft Margin SVC using NMF

A short conclusion will be that the performance of hard margin SVM is normally better than the performance of soft margin SVM. In the meanwhile, the LSI works better than NMF.

* 1. SVC with Cross Validation

4.2.1 LSI

(A) min\_df=2

In this task we use Soft Margin SVM, to classify the documents. Cross validation is used with 5 folds, to determine the correct value of the L2 regularization parameter. We determined that k = 2 gave the best training score, and use this parameter for the final classification.

Table 19. Statistics--SVC using LSI, with CV, gamma=100, min\_df=2

|  |  |
| --- | --- |
| Statistic | Result |
| Accuracy | 0.972063 |
| Precision | 0.963558 |
| Recall | 0.981750 |

Table 20. Confusion Matrix--SVC using LSI, with CV, gamma=100, min\_df=2

|  |  |  |
| --- | --- | --- |
|  | Predicted Computer Technology | Predicted Recreational Activity |
| Actual Computer Technology | 1502 | 59 |
| Actual Recreational Activity | 29 | 1560 |

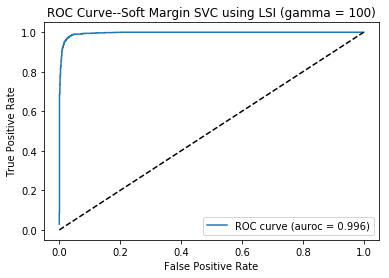


Figure 8. ROC curve-- SVC using LSI, with CV, gamma=100, min\_df=2

(B)min\_df=5

We determined that k = 2 gave the best training score, and use this parameter for the final classification.

Table 21. Statistics--SVC using LSI, with CV, gamma=100, min\_df=5

|  |  |
| --- | --- |
| Statistic | Result |
| Accuracy | 0.972063 |
| Precision | 0.963558 |
| Recall | 0.981750 |

Table 22. Confusion Matrix--SVC using LSI, with CV, gamma=100, min\_df=5

|  |  |  |
| --- | --- | --- |
|  | Predicted Computer Technology | Predicted Recreational Activity |
| Actual Computer Technology | 1502 | 59 |
| Actual Recreational Activity | 29 | 1560 |

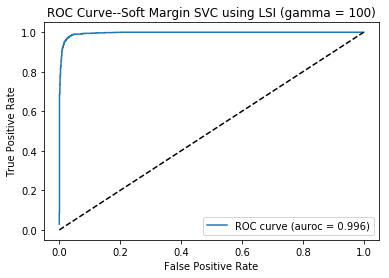


Figure 9. ROC curve-- SVC using LSI, with CV, gamma=100, min\_df=5

4.2.2 NMF, min\_df=2

We determined that k = 3 gave the best training score, and use this parameter for the final classification.

Table 23. Statistics--SVC using NMF, with CV, gamma=1000

|  |  |
| --- | --- |
| Statistic | Result |
| Accuracy | 0.961905 |
| Precision | 0.951444 |
| Recall | 0.974198 |

Table 24. Confusion Matrix--SVC using NMF, with CV, gamma=1000

|  |  |  |
| --- | --- | --- |
|  | Predicted Computer Technology | Predicted Recreational Activity |
| Actual Computer Technology | 1482 | 79 |
| Actual Recreational Activity | 41 | 1548 |

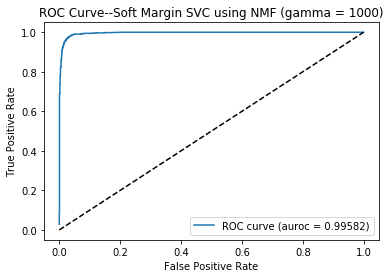


Figure 10. ROC curve-- SVC using NMF, with CV, gamma=1000

Taking the average of the accuracy, recall and precision under different values for γ as shown above.

The maximum accuracy, precision and recall achieve the best result with γ=1000.

* 1. Naïve Bayes

We turn to Naive Bayes to perform the same tests as in question (f). The algorithm estimates the maximum likelihood probability of a class given a document with feature set X, using Bayes rule, based upon the assumption that given the class, the features are statistically independent

* + 1. LSI

Table 25. Statistics—Naïve Bayes Classifier using LSI

|  |  |
| --- | --- |
| Statistic | Result |
| Accuracy | 0.898730 |
| Precision | 0.846616 |
| Recall | 0.976086 |

Table 26. Confusion Matrix--Naïve Bayes Classifier using LSI

|  |  |  |
| --- | --- | --- |
|  | Predicted Computer Technology | Predicted Recreational Activity |
| Actual Computer Technology | 1280 | 281 |
| Actual Recreational Activity | 38 | 1551 |

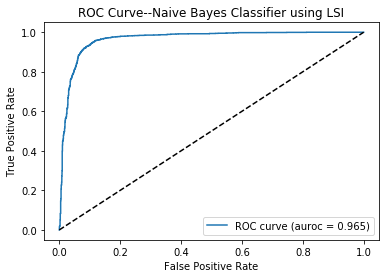


Figure 11. ROC curve-- Naïve Bayes Classifier using LSI

4.3.2 NMF

Table 27. Statistics—Naïve Bayes Classifier using NMF

|  |  |
| --- | --- |
| Statistic | Result |
| Accuracy | 0.940000 |
| Precision | 0.935323 |
| Recall | 0.946507 |

Table 28. Confusion Matrix--Naïve Bayes Classifier using NMF

|  |  |  |
| --- | --- | --- |
|  | Predicted Computer Technology | Predicted Recreational Activity |
| Actual Computer Technology | 1457 | 104 |
| Actual Recreational Activity | 85 | 1504 |

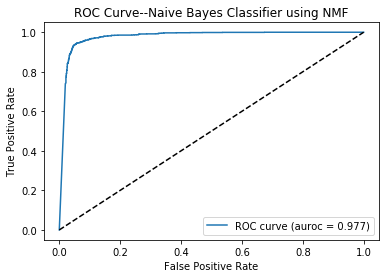


Figure 12. ROC curve-- Naïve Bayes Classifier using NMF

Generally, using NMF to select features tend to help to advance the performance of the Naïve Bayes Classifier.

* 1. Logistic Regression

The same tests are now applied to Logistic Regression (LR). LR quantifies the relationship

between the categorical dependent variable and one or more independent variables by estimating

probabilities using a logistic function, which is the cumulative logistic distribution.

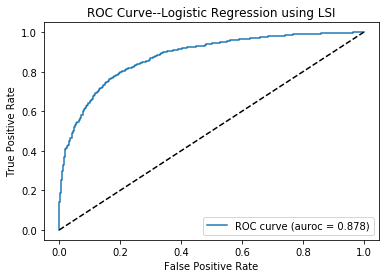
* + 1. LSI

Table 29. Statistics—Logistic Regression using LSI

|  |  |
| --- | --- |
| Statistic | Result |
| Accuracy | 0.756825 |
| Precision | 0.885047 |
| Recall | 0.595597 |

Table 30. Confusion Matrix—Logistic Regression using LSI

|  |  |  |
| --- | --- | --- |
|  | Predicted Computer Technology | Predicted Recreational Activity |
| Actual Computer Technology | 1437 | 123 |
| Actual Recreational Activity | 643 | 947 |

 Figure 13. ROC curve -- Logistic Regression using LSI

We can see that LR and SVM have nearly the same area under the ROC, implying that they

classify records most accurately, while Naive Bayes lacks the same accuracy.

4.4.2 NMF

Table 31. Statistics—Logistic Regression using NMF

|  |  |
| --- | --- |
| Statistic | Result |
| Accuracy | 0.768571 |
| Precision | 0.776139 |
| Recall | 0.761006 |

Table 32. Confusion Matrix—Logistic Regression using NMF

|  |  |  |
| --- | --- | --- |
|  | Predicted Computer Technology | Predicted Recreational Activity |
| Actual Computer Technology | 1211 | 349 |
| Actual Recreational Activity | 380 | 1210 |

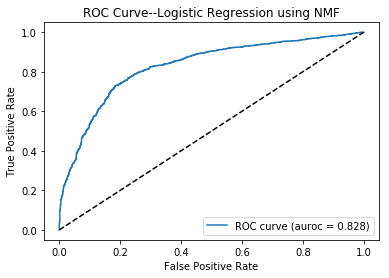


Figure 14. ROC curve -- Logistic Regression using NMF

4.4.3 Logistic Regression with Regularization

Now, we repeat the tests performed in question (h), but we’ve now added a regularization term

to the optimization objective. Both l1 and l2 are norm regularizations.

We observed that for very small values of the parameter, excessive regularization took place, and the testing error was very high. However, on increasing the parameter, the testing error steadily reduced before increasing again.

We would typically use l1 loss function when we need a robust solution, but have the computational

power, and are willing to tolerate multiple stable solutions.

If the dataset is very large, or if a single unstable solution will work, we would use the l2 loss function.

We also observed that as the regularization parameter increased, the fitted hyperplane shifts away from the origin, before coming closer again.

For graphical support, it turns out that l1 norm regularization with the value of regularization coefficient being 0.05 and l2 norm regularization with the value of regularization coefficient being 100.

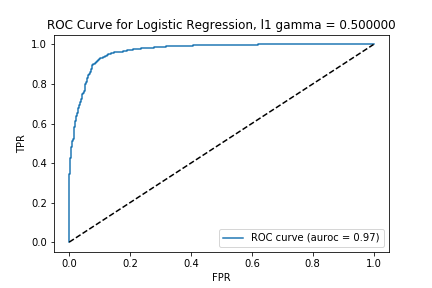


Figure 15. Logistic Regression with Regularization

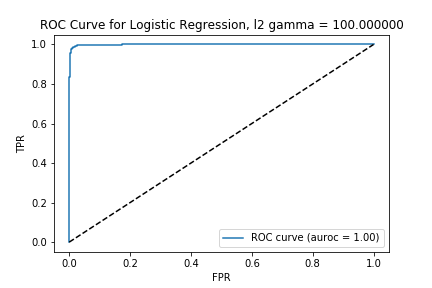


Figure 16. Logistic Regression with Regularization

1. **Multiclass Classification**

In the previous sections, we mainly focus on the binary classification. In this part, multiclass classification techniques will be explored through different algorithms to have a sense of the multiclass classification result. In this section, we aim to learn classifiers on the documents belonging to the classes mentioned before. Naming these set as I, II, III and IV.

Table 33. Target data sets

|  |  |
| --- | --- |
| Set I | *comp.sys.ibm.pc.hardware,* |
| Set II | *comp.sys.mac.hardware,* |
| Set III | *misc.forsale,* |
| Set IV | *soc.religion.christian* |

* 1. Naïve Bayes Classification

As is known, Naïve Bayes algorithm finds the class with maximum likelihood given the data, regardless of the number of classes, which performs the multiclass classification inherently.

Table 34. Statistics of Naïve Bayes Classification

|  |  |
| --- | --- |
| **Statistics** | **Value** |
| Accuracy | 0.768690 |
| Precision | 0.771042 |
| Recall | 0.768690 |

We can have the confusion matrix.

Table 35. Confusion Matrix for Naïve Bayes Classification

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ***Predict I*** | ***Predict II*** | ***Predict III*** | ***Predict IV*** |
| ***Actual I*** | 299 | 36 | 51 | 6 |
| ***Actual II*** | 97 | 216 | 68 | 4 |
| ***Actual III*** | 49 | 33 | 298 | 10 |
| ***Actual IV*** | 3 | 2 | 3 | 390 |

* 1. Multiclass SVM Classification (One Vs One)

For SVM, we need to extend the binary classification techniques when there are multiple classes. It can be performed along with a one versus one classification on all pairs of classes so as to grant the document the class assigned to have the majority vote. If there are more than one class with the highest vote, we pick the class in the binary classification with the highest total classification confidence levels.

Table 36. Statistics of Multiclass SVM Classification (One Vs One)

|  |  |
| --- | --- |
| **Statistics** | **Value** |
| Accuracy | 0.80000 |
| Precision | 0.80665 |
| Recall | 0.80000 |

We can have the confusion matrix.

Table Confusion Matrix Multiclass SVM Classification (One Vs One)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ***Predict I*** | ***Predict II*** | ***Predict III*** | ***Predict IV*** |
| ***Actual I*** | 299 | 36 | 51 | 6 |
| ***Actual II*** | 97 | 216 | 68 | 4 |
| ***Actual III*** | 49 | 33 | 298 | 10 |
| ***Actual IV*** | 3 | 2 | 3 | 390 |

* 1. Multiclass SVM Classification (One Vs Rest)

We can fit one classifier per class reducing the number of classifiers to be learnt to |𝐶| by one versus rest of the method.

For each classifier, the class is fitted against all the rest of classes. However, we should handle the unbalanced cases.

In all, through learning a single classifier for each class, the insights on the interpretation of the classes based on the features can be obtained.

Table Statistics Multiclass for SVM Classification (One Vs Rest)

|  |  |
| --- | --- |
| **Statistics** | **Value** |
| Accuracy | 0.813419 |
| Precision | 0.813628 |
| Recall | 0.813419 |

We can have the confusion matrix.

Table Confusion Matrix Multiclass SVM Classification (One Vs Rest)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ***Predict I*** | ***Predict II*** | ***Predict III*** | ***Predict IV*** |
| ***Actual I*** | 299 | 36 | 51 | 6 |
| ***Actual II*** | 97 | 216 | 68 | 4 |
| ***Actual III*** | 49 | 33 | 298 | 10 |
| ***Actual IV*** | 3 | 2 | 3 | 390 |

* 1. Conclusion for Multiclass Classification

OneVSRest SVM achieves the best performance based the results shown above.

**Reference:**

[1]20Newsgroups <http://qwone.com/~jason/20Newsgroups/>

[2]20 Newsgroups text data set

<http://scikit-learn.org/stable/datasets/twenty_newsgroups.html>