EE239AS Special Topics in Signals and Systems

Project 2 Report:

Classification Analysis

Group Members:

Jiayu Guo(504513188), Yitian Hu(904516321),
Peidong Chen(204432674), Yang Yang(804522285)

Winter 2016

02/22/2016

Problem (a)

In this project we work with “20 Newsgroups” dataset. We plot a histogram for the following 8 topics: comp.graphics, comp.os.ms-windows.misc, comp.sys.ibm.pc.hardware, comp.sys.mac.hardware, rec.autos, rec.motorcycles, rec.sport.baseball and rec.sport.hockey. We use the training part of the dataset. And the plot is shown below:

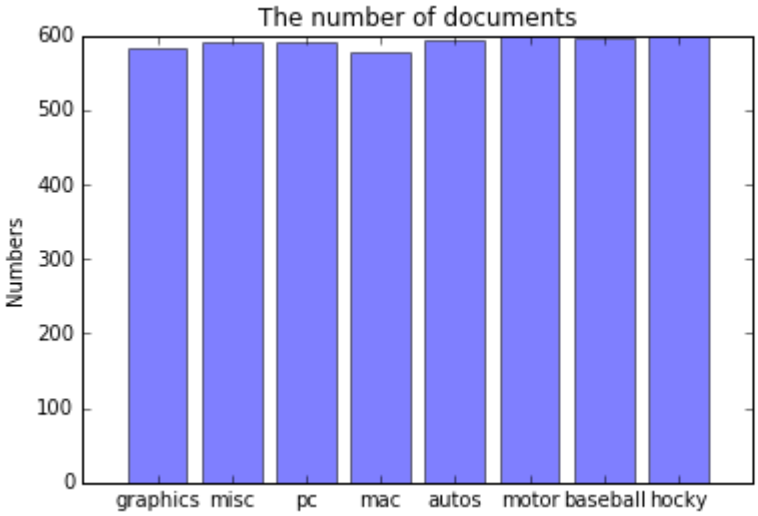


Fig.1 The plot of the number of documents per topic

From the plot above, we can see that they are evenly distributed.

And we compute that the number of documents in training part in the group "Computer Technology" is 2343, and that is the group "Recreational Activity" is 2389.

Problem (b)

We use the "SnowballStemmer" library to exclude different stems of a word. And the number of terms of the train data we extracted is 54883.

Problem (c)

The LSI can be computed by the following function:

$$\begin{aligned} \min_{\boldsymbol{w},b,\boldsymbol{\xi}} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_n \xi_n \\ \text{s.t.} \quad & y_n[\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b] \geq 1 - \xi_n, \quad \forall \ n \\ & \xi_n \geq 0, \quad \forall \ n \end{aligned}$$

To compute LSI, we firstly append all document belong to one class together. Then we get a list with length 20. For each term of the list, we get all the content of the document belong to one class. Then we can use TfidfVectorizer to process the list. This will get the TFxICF of each class. Then we can sort the value and get the 10 most significant terms.

For the class "comp.sys.ibm.pc.hardware", the 10 most significant terms that we find are:

```
[u'bios', u'card', u'controller', u'disk', u'doe', u'drive', u'ide
', u'jumper', u'problem', u'scsi']
```

For the class "comp.sys.mac.hardware", the 10 most significant terms that we find are:

```
[u'apple', u'doe', u'drive', u'mac', u'monitor', u'nubus', u'probl
em', u'quadra', u'scsi', u'simms']
```

For the class "misc.forsale", the 10 most significant terms that we find are:

```
[u'condition', u'do', u'game', u'manual', u'new', u'offer', u'pric
e', u'sale', u'shipping', u'wolverine']
```

For the class "soc.religion.christian", the 10 most significant terms that we find are:

```
[u'bible', u'christ', u'christian', u'church', u'doe', u'faith', u'god
', u'jesus', u'people', u'say']
```

Problem (d)

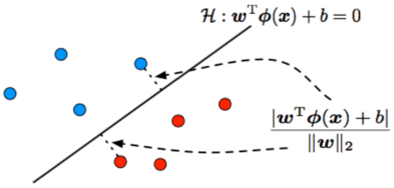
We apply LSI to the TFxIDF matrix and pick k=50, and each document is mapped to a 50-dimensional vector. The reduced vector shape of the training set is (11314, 50).

Problem (e)

We use SVM method to separate the documents into Computer Technology vs Recreational Activity groups.

The SVM is a supervised learning methods. For simplify, it will find the optimized decision boundary between the train data.
For example, as shown in the picture below, we have two class of data and have to find the best decision boundary. We can do this bu find the smalled distance between the hyperplane and all the training data by using the function below.

$$\text{MARGIN}(\boldsymbol{w}, b) = \min_{n} \frac{y_n[\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b]}{\|\boldsymbol{w}\|_2}$$


$\mathcal{H}: \boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}) + b = 0$
$\frac{|\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}) + b|}{\|\boldsymbol{w}\|_2}$

So the problem can be simplified into the following problem:

$$\begin{aligned} \min_{\boldsymbol{w}, b} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 \\ \text{s.t.} \quad & y_n[\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b] \geq 1, \quad \forall \quad n \end{aligned}$$

In problem e, we use the LinearSVC function in sklearn. In this problem, we will do a hard margin svm firstly, as we will do soft margin svm in the next problem. For simulating the svm, the code is shown below:

```
svm = SVC(kernel='linear', probability=True, random_state=40)
```

Here is the plot of ROC curve

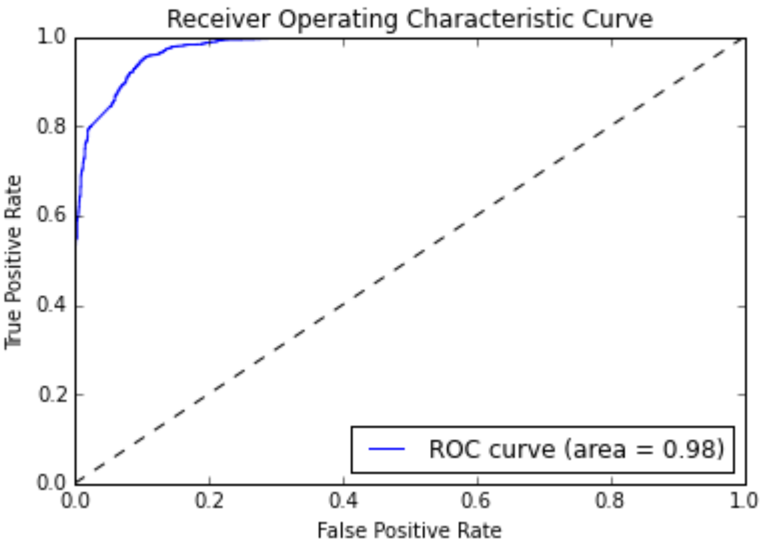


Fig. 2 ROC curve of problem (e)

The confusion matrix is

```
[[1352   208]
  [   51 1539]]
```

The accuracy score is 0.9228
The recall score is 0.97

The precision score is 0.88

Problem (f)

We have tried hard margin svm in problem e, there will also be some non-separable point in the data set. We should modify our constraints to account for non-separable point, so we can introduce slack variable. And the problem can be simplified as the following:

$$\begin{aligned} \min_{\boldsymbol{w},b,\boldsymbol{\xi}} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_n \xi_n \\ \text{s.t.} \quad & y_n[\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b] \geq 1 - \xi_n, \quad \forall \ n \\ & \xi_n \geq 0, \quad \forall \ n \end{aligned}$$

So in this problem, we use soft margin SVM approach. We find the best value of the parameter is 100.

Here is the plot of ROC curve

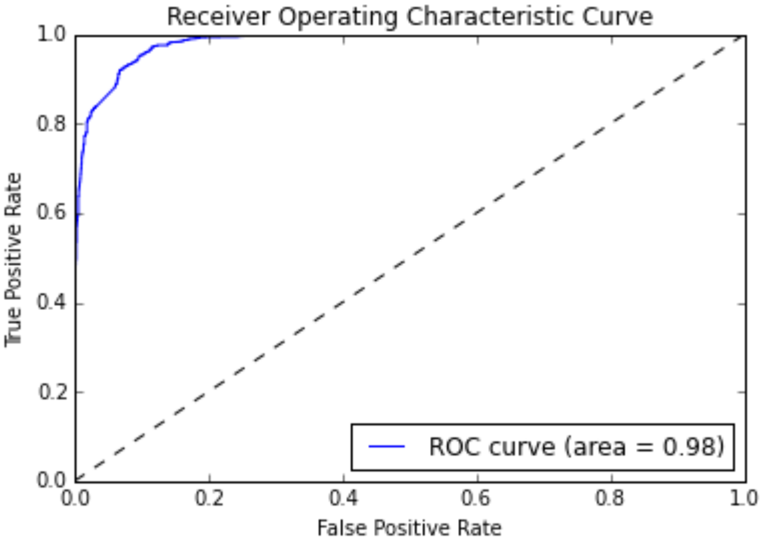


Fig. 3 ROC curve of problem (f)

The confusion matrix is

```
[[1401    159]
  [    74 1516]]
```

The accuracy score is 0.926
The recall score is 0.953
The precision score is 0.905

Problem (g)

We use naïve Bayes algorith in this problem. Given a random vairble X and a dependent variable Y, the naïve bayes model defines the joint distribution as follows:

$$
\begin{aligned}
P(X=x, Y=c) &= P(Y=c)P(X=x|Y=c) \\
&= P(Y=c)\prod_{d=1}^{\mathsf{D}} P(X_d = x_d|Y=c)
\end{aligned}
$$

With the assumption that all X_d are categorical varicables from the same domain and $P(X_d=x_d|Y=c)$ depends only on the value of x_d, not d itself. We can simplified the definition as follows:

$$P(X=x, Y=c) = P(Y=c)\prod_k P(k|Y=c)^{z_k} = \pi_c \prod_k \theta_{ck}^{z_k}$$

Where z_k is the number of times k in x.

For training data, we should optimize the model according to the following function:

$$
\begin{aligned}
\mathcal{L} = \log P(\mathcal{D}) &= \log \prod_{n=1}^{\mathsf{N}} \pi_{y_n} P(x_n|y_n) \\
&= \log \prod_{n=1}^{\mathsf{N}} \left(\pi_{y_n} \prod_k \theta_{y_n k}^{z_{nk}} \right) \\
&= \sum_n \left(\log \pi_{y_n} + \sum_k z_{nk} \log \theta_{y_n k} \right) \\
&= \sum_n \log \pi_{y_n} + \sum_{n,k} z_{nk} \log \theta_{y_n k}
\end{aligned}
$$

$$(\pi_c^*, \theta_{ck}^*) = \arg\max \sum_n \log \pi_{y_n} + \sum_{n,k} z_{nk} \log \theta_{y_n k}$$

It also has the folloing constraint:

$$\sum_c \pi_c = 1$$

$$\sum_k \theta_{ck} = \sum_k P(k|Y=c) = 1$$

After training, we can predict the probility of Y according to x. For each class, we will get different probility. And the highest probility will be the predicted result.

In this question, since the feature vectors you are dealing with are transformed via LSI and are potentially negative, we train a Gaussian naive Bayes classfier in which a Gaussian distribution is imposed on p(x|c) terms in training. The result is shown below:

Here is the plot of ROC curve

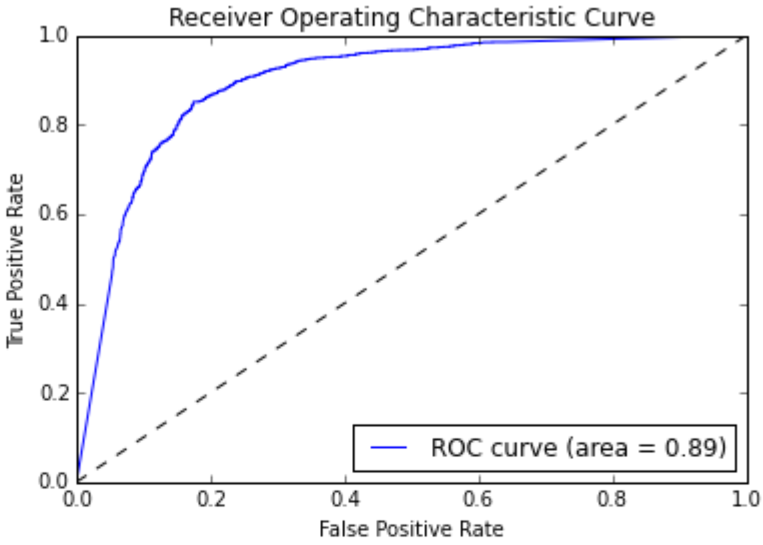


Fig. 4 ROC curve of problem (g)

The confusion matrix is

```
[[1019    541]
  [    86 1504]]
```

The accuracy score is 0.801

The recall score is 0.946

The precision score is 0.735

Because the naïve bayes has a strong assumption strong (naive)independence assumptions between the features. The dataset may not fit the assumption well. Thus we can see the result of naïve bayes is not as well as svm.

Problem (h)

We use logistic regression in this problem. Different to naïve bayes, logistic regression models the conditional distribution: P(Y|X). It is a discriminative model. It will minimizes a cost function to get the model for prediction.

Here is the plot of ROC curve

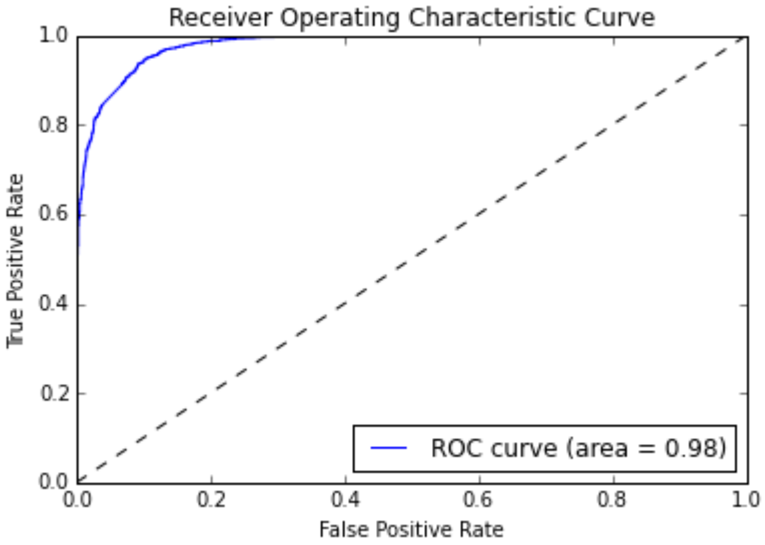


Fig. 5 ROC curve of problem (h)

The confusion matrix is

```
[[1372   188]
 [   68 1522]]
```

The accuracy score is 0.9244

The recall score is 0.957

The precision score is 0.890

Problem (i)

First, we use multiclass SVM classification. To do this, we use the multiclass function in sklearn. To do one vs one method, we will use OneVsOneClassifer. To do one vs reset method, we will use OneVsRestClassifer.

For One vs One method,

The confusion matrix is

```
[[280   75   35    2]
 [ 63 274   44    4]
[ 35   27 325    3]
[   4    5   34 355]]
```

The accuracy score is 0.788

The recall score is 0.788

The precision score is 0.793

For One vs Rest method,

The confusion matrix is

```
[[276   64   41   11]
  [ 60 264   44   17]
  [ 33   19 325   13]
  [   2     1   21 374]]
```

The accuracy score is 0.792

The recall score is 0.792

The precision score is 0.790

Second, we use Naïve Bayes classification.

The confusion matrix is

```
[[191   68   85   48]
  [ 60 167   78   80]
[ 49   23 244   74]
[   1     0   10 387]]
```

The accuracy score is 0.632

The recall score is 0.632

The precision score is 0.631