CS 5785: Applied Machine Learning **(Due: 10/10/2019)**

Homework #2

Instructor: Serge Belongie *Name:* Yanlin Chen, Yang Ma*, Netid:* yc2565, ym473

Part 1: Programming Exercises

1.Summary:

In this exercise, we found that when using SVD(r=10) as a feature reduction method and Logistic Regression as a classifier, we can reach a classification accuracy of 0.79 on the test set. What's more, when we change r from 1 to 200, the classification accuracy increases rapidly when r is under 20 and converges to almost 0.95(r>0.95).

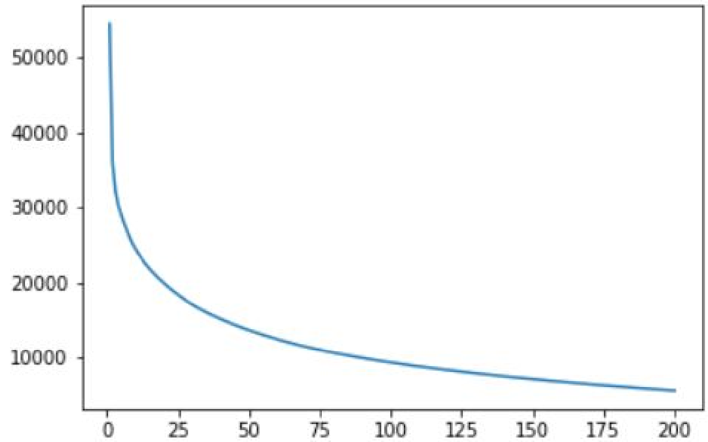
(b)

(c)

(d)

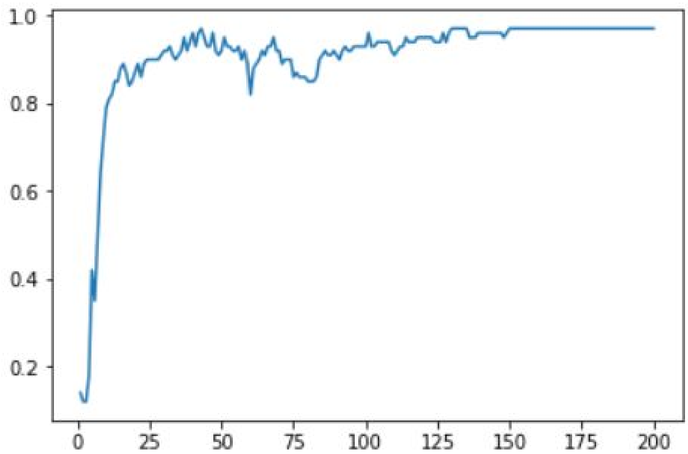
(e)

(f)



(g)

```
[ 7  1  5  3  1  1  1  1  1  4  3  2  2  2  2  2  2  2  2  3  2  2  3  3
  3  3  3  3  3  3  7  1  4  4  7  4  4  4  4  4  5  5  5  6  5  5  5  5
  5  5  6  6  6  1  6  6  6  6  6  6  7  7  3  7  7  6  9  7  7  5  8  8
  8  8  8  8  4  8  8  8  8  9  9  9  9  9  9  9  9  9  4 10 10 10  1 10
 10 10 10 10]
0.79
```

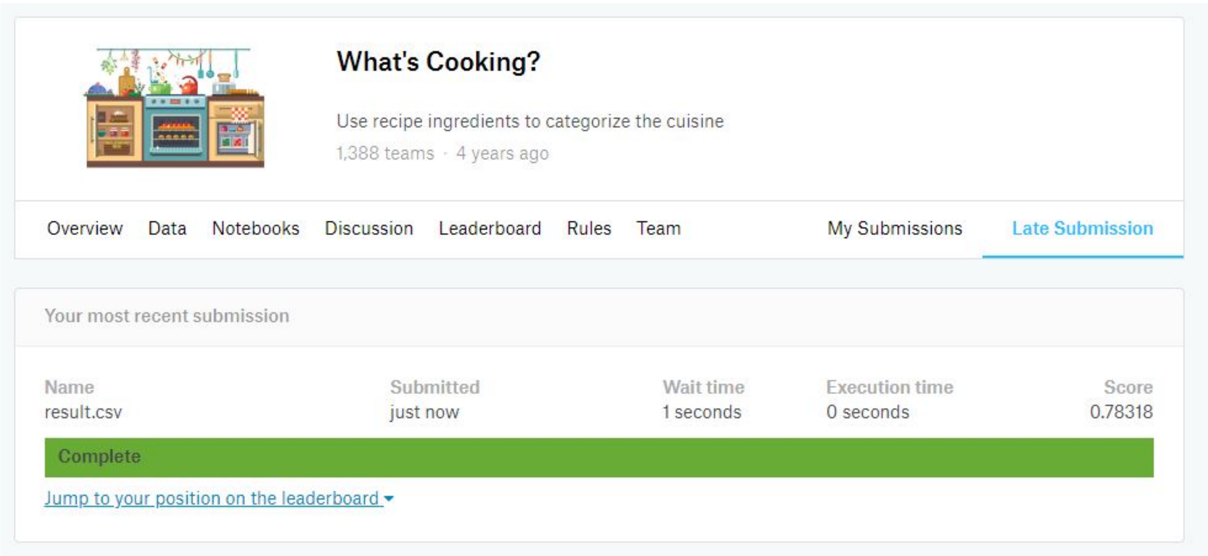


2. Summary:

In this exercise, we compare three different classifiers: Naive Bayes-Gaussian, Naive Bayes-Bernoulli and Logistic Regression. We found that when using 3 fold cross-validation on our train set, Logistic Regression performs best with an accuracy of 0.78 while Naive Bayes-Gaussian classifier performs worst with an accuracy of 0.38 and the longest run time.

(b)

How many samples(dishes) in trainset: 39774
How many categories(types of cuisine): 20
How many unique ingredients: 6714
(c),(d),(f)
average acc for 3 fold cross-validation on trainset:
1. Naive Bayes-Gaussian: 0.3794937938208349
2. Naive Bayes-Bernoulli: 0.6835876576455521
3. Logistic Regression: 0.7757586704089684
(g)



Part 2: Written Exercises

1.

2.

After applying the Lagrange multiplier,

$$L(\lambda) = a^T B a - \lambda (a^T W a - 1)$$

$$\frac{\partial L}{\partial a} = (B + B^T) a - \lambda (W - W^T) a = 0$$

Then, we can derive

$$((W + W^T))^{-1} (B + B^T) a = \lambda a \quad \text{(standard eigenvalue problem)}$$

Assuming W and B are symmetric. Then,

$$W^{-1} B a = \lambda a$$

(a) From the equation given in the book, the LDA rule classifies to class 2 if

$$x^T \hat{\Sigma}^{-1} (\hat{\mu}_2 - \hat{\mu}_1) > \frac{1}{2} (\hat{\mu}_2 + \hat{\mu}_1)^T \hat{\Sigma}^{-1} (\hat{\mu}_2 - \hat{\mu}_1) - \log(N_2/N_1)$$

$\because$ The right side $= \frac{1}{2} (\hat{\mu}_2 + \hat{\mu}_1)^T \hat{\Sigma}^{-1} (\hat{\mu}_2 - \hat{\mu}_1) - \log(N_2/N_1)$

$\therefore$ The LDA rule classifies to class 2 if

$$x^T \hat{\Sigma}^{-1} (\hat{\mu}_2 - \hat{\mu}_1) > \frac{1}{2} (\hat{\mu}_2 + \hat{\mu}_1)^T \hat{\Sigma}^{-1} (\hat{\mu}_2 - \hat{\mu}_1) - \log(N_2/N_1)$$

and class 1 otherwise

(b) Let $\beta' = \begin{pmatrix} \beta \\ \beta_0 \end{pmatrix}$. Then, we have

$$\frac{\partial RSS(\beta')}{\partial \beta_0} = -2\sum_{i=1}^{N}(y_i - \beta_0 - \beta^T x_i) = 0 \quad (1)$$

$$\frac{\partial RSS(\beta')}{\partial \beta} = -2\sum_{i=1}^{N} x_i(y_i - \beta_0 - \beta^T x_i) = 0 \quad (2)$$

From (1), we know that

$$\beta_0 = \frac{1}{N}\sum_{i=1}^{N}(y_i - \beta^T x_i) \quad (3)$$

From (2) & (3), we can derive that

$$\sum_{i=1}^{N} x_i[\beta^T(x_i - \bar{x})] = \sum_{i=1}^{N} x_i\left(y_i - \frac{1}{N}\sum_{j=1}^{N} y_j\right)$$

$$= \sum_{k=1}^{2}\sum_{y_i=k} x_i\left(y_k - \frac{N_1 y_1 + N_2 y_2}{N}\right)$$

$$\because \sum_{y_i=k} x_i = N_k \hat{\mu}_k$$

$$\Rightarrow = \sum_{k=1}^{2} N_k \hat{\mu}_k\left(\frac{N y_k - (N_1 y_1 + N_2 y_2)}{N}\right)$$

$$= N(\hat{\mu}_2 - \hat{\mu}_1) \quad (4)$$

To compute the left side of the original equation,

$$(N-2)\hat{\Sigma} = \sum_{k=1}^{2}\sum_{y_i=k}(x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T$$

$$\because x_i^T \hat{\mu}_k = x_i \hat{\mu}_k^T \text{ and } \sum_{y_i=k} x_i = N_k \hat{\mu}_k$$

$$\therefore (N-2)\hat{\Sigma} = \sum_{i=1}^{N} x_i x_i^T - (N_1 \hat{\mu}_1 \hat{\mu}_1^T + N_2 \hat{\mu}_2 \hat{\mu}_2^T)$$

$$= \frac{1}{N}(N_1\hat{\mu}_1 + N_2\hat{\mu}_2)(N_1\hat{\mu}_1 + N_2\hat{\mu}_2)^T - (N_1 \hat{\mu}_1 \hat{\mu}_1^T + N_2 \hat{\mu}_2 \hat{\mu}_2^T)$$

Then, the other way to compute $\sum_{i=1}^{N} x_i[\beta^T(x_i - \bar{x})]$ is

$$\sum_{i=1}^{N} x_i[\beta^T(x_i - \bar{x})] = \left(\sum_{i=1}^{N} x_i x_i^T - \sum_{i=1}^{N} x_i \bar{x}^T\right)\beta$$

From (4), we know $\sum_{i=1}^{N} x_i x_i^T = \frac{1}{N}(N_1\hat{\mu}_1 + N_2\hat{\mu}_2)(N_1\hat{\mu}_1 + N_2\hat{\mu}_2)^T$

$$\longrightarrow = \left((N-2)\hat{\Sigma} + \frac{N_1 N_2}{N}(\hat{\mu}_2\hat{\mu}_2^T - \hat{\mu}_1\hat{\mu}_2^T - \hat{\mu}_2\hat{\mu}_1^T + \hat{\mu}_1\hat{\mu}_1^T)\right)\beta$$

$$= \boxed{\left((N-2)\hat{\Sigma} + N\hat{\Sigma}_B\right)\beta = N(\hat{\mu}_2 - \hat{\mu}_1)} \Leftarrow \text{from } (4)$$

(c) $\because \hat{\Sigma}_B \hat{\beta} = (\hat{\mu}_2 - \hat{\mu}_1)(\hat{\mu}_2 - \hat{\mu}_1)^T \beta$ where $(\hat{\mu}_2 - \hat{\mu}_1)^T \beta$ is a number

$\therefore \hat{\Sigma}_B \hat{\beta}$ is in the direction of $\hat{\mu}_2 - \hat{\mu}_1$

Combined with the result of (b),

$$\hat{\beta} = \frac{1}{N-2} \hat{\Sigma}^{-1} \left(N(\hat{\mu}_2 - \hat{\mu}_1) - \frac{N_1 N_2}{N} \hat{\Sigma}_B \beta \right)$$

$$= \frac{1}{N-2} \left(N - \frac{N_1 N_2}{N} (\hat{\mu}_2 - \hat{\mu}_1)^T \beta \right) \hat{\Sigma}^{-1} (\hat{\mu}_2 - \hat{\mu}_1) \quad (5)$$

$$\sim \hat{\Sigma}^{-1} (\hat{\mu}_2 - \hat{\mu}_1)$$

(d) Since (5) holds for any N, the result hold for any coding.

(e) $\because y_i = 0$ is the boundary condition

$\therefore$ from (3), $\hat{\beta}_0 = \frac{1}{N} \sum_{i=1}^{N} \hat{\beta}^T x_i = \hat{\beta}^T \hat{\mu} = \hat{\mu}^T \hat{\beta}$

$\therefore f(x) = x^T \hat{\beta} + \hat{\beta}_0 = (x^T - \mu^T) \hat{\beta}$

$\because \hat{\beta} = \lambda \hat{\Sigma}^{-1} = \lambda \hat{\Sigma}^{-1} (\hat{\mu}_2 - \hat{\mu}_1) \quad \lambda \in \mathbb{R}$

$\therefore x^T \hat{\Sigma}^{-1} (\hat{\mu}_2 - \hat{\mu}_1) > \mu^T \hat{\Sigma}^{-1} (\hat{\mu}_2 - \hat{\mu}_1)$

where $\mu^T = \frac{N_1}{N} \hat{\mu}_1^T + \frac{N_2}{N} \hat{\mu}_2^T$

when $N_1 = N_2$, it will be the same as the LDA rule.

3.

```
(a):
M.T*M=
[[ 39  57  60]
 [ 57 118  53]
 [ 60  53 127]]
M*M.T=
[[ 10   9  26   3  26]
 [  9  62   8  -5  85]
 [ 26   8  72  10  50]
 [  3  -5  10   2  -1]
 [ 26  85  50  -1 138]]

(b):
Eigenvalues of M.T*M=
[2.14670489e+02 9.32587341e-15 6.93295108e+01]
Eigenvalues of M*M.T=
[ 2.14670489e+02 -8.88178420e-16  6.93295108e+01 -3.34838281e-15
  7.47833227e-16]

(c):
Eigenvectors of M.T*M=
[[ 0.42615127  0.90453403 -0.01460404]
 [ 0.61500884 -0.30151134 -0.72859799]
 [ 0.66344497 -0.30151134  0.68478587]]
Eigenvectors of M*M.T=
[[-0.16492942 -0.95539856  0.24497323 -0.54001979 -0.78501713]
 [-0.47164732 -0.03481209 -0.45330644 -0.62022234  0.30294097]
 [-0.33647055  0.27076072  0.82943965 -0.12704172  0.2856551 ]
 [-0.00330585  0.04409532  0.16974659  0.16015949  0.43709105]
 [-0.79820031  0.10366268 -0.13310656  0.53095405 -0.13902319]]
```

```
(d):
SVD of M:
U = eigenvectors of M*M.T =
[[-0.16492942 -0.24497323]
 [-0.47164732  0.45330644]
 [-0.33647055 -0.82943965]
 [-0.00330585 -0.16974659]
 [-0.79820031  0.13310656]]
Sigma = diag(sqrt(eigenvalues of M*M.T)) =
[[14.65163776  0.        ]
 [ 0.          8.32643446]]
V.T = eigenvectors of M.T*M =
[[-0.42615127 -0.61500884 -0.66344497]
 [ 0.01460404  0.72859799 -0.68478587]]
U * Sigma * V.T =
[[ 1.00000000e+00  6.66133815e-16  3.00000000e+00]
 [ 3.00000000e+00  7.00000000e+00  2.00000000e+00]
 [ 2.00000000e+00 -2.00000000e+00  8.00000000e+00]
 [-4.85722573e-17 -1.00000000e+00  1.00000000e+00]
 [ 5.00000000e+00  8.00000000e+00  7.00000000e+00]]

(e):
One-dimentinal approximation of M =
[[-0.16492942]
 [-0.47164732]
 [-0.33647055]
 [-0.00330585]
 [-0.79820031]]
*
[[14.65163776]]
*
[[-0.42615127 -0.61500884 -0.66344497]]
=
[[1.02978864 1.48616035 1.60320558]
 [2.94487812 4.24996055 4.58467382]
 [2.10085952 3.031898   3.27068057]
 [0.02064112 0.02978864 0.0321347 ]
 [4.9838143  7.19249261 7.75895028]]
```