CSC2515: Assignment 3

Due on Wednesday, December 06, 2017

Professor Ethan Fetaya

Zhaoyu Guo (999008069)

Question One

In this question, we will work with 20 newsgroups dataset. This dataset contains posts from 20 different newsgroups and the task is to classify each post to the correct newsgroup. The goal will be to evaluate several models of our own choice and select those with the best test performance on this task.

3 Good Models

I used Support Vector Machine, Logistic Regression, and Multinomial Naive Bayes as the 3 good algorithms in this report. Before I selected the final 3 algorithms, I've tried KNN as well, however, with very poor performance. I believe KNN is not suitable for this dataset because of computing efficiency in high dimension. In addition, I used **tf-idf features** instead of **bow features** to extract features since we can get higher accuracy.

How to select the best hyperparameters

I used the Grid Search to find the best hyper-parameters for the 3 algorithms. We import **GridSearchCV** from sklearn in python. For **SVM**, we searched from a range between $C = 0.01$ and $C = 2$, and found the best hyper-parameter is $C = 1.2$. For **logistic regression**, we searched from a range between $C = 0.1$ and $C = 3$, and found the best hyper-parameter is $C = 2.9$. For **Multinomial Naive Bayes**,we searched from a range between $\alpha = 0.001$ and $\alpha = 1$, and found the best hyper-parameter is $\alpha = 0.01$. Then we fitted those 3 models with the optimized hyper-parameter. Please see **q1.py** for details. I've commented the codes for hyper-parameter tuning since it will take very long time.

Baseline - Bernoulli Naive Bayes

The classification accuracy on the training set is: 59.8727%
The classification accuracy on the test set is: 45.7913%

Support Vector Machine (with best hyper-parameter)

The classification accuracy on the training set is: 96.0757%
The classification accuracy on the test set is: 66.3702%

Logistic Regression (with best hyper-parameter)

The classification accuracy on the training set is: 95.0327%
The classification accuracy on the test set is: 68.667%

Multinomial Naive Bayes (with best hyper-parameter)

The classification accuracy on the training set is: 95.89%
The classification accuracy on the test set is: 70.021%

Discussion

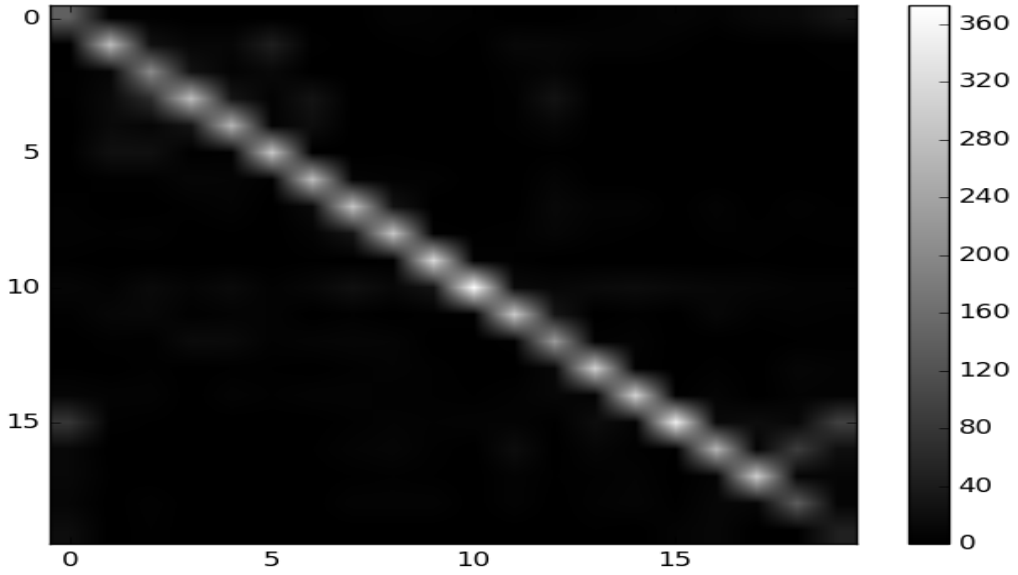
Based on the test set classification accuracy, we can see that all of those 3 models can beat the baseline(Bernoulli Naive Bayes). The best model will be Multinomial Naive Bayes, this is same as I expected. The Multinomial Naive Bayes only require a small amount of training data, and based on the research from internet, the Multinomial Naive Bayes can work well in text data. In addition, the model can be trained very fast. For Logistic Regression and Support Vector Machine, we can get higher test set classification

accuracy compared with the baseline. However, it will take very long time to find the best hyperparameter. In addition, the test set classification accuracy still lower than the Multinomial Naive Bayes. In Logistic Regression, we build the model based on conditional likelihood, and we need to choose initial weights and then use method of gradient descent to find the weights which maximize the likelihood, and then we get the predicted class. In addition, we do not need to assume features are independent for logistic regression which Naive Bayes needs. Similarly, for the SVM, we need to use hinge loss for "maximum-margin" classification. In our dataset, we have 11314 train data points, with dimension of the feature is 101631. Therefore, we should expect that we won't be able to get very good test set classification accuracy without doing any feature selections.

Confusion Matrix

The best classifier is Multinomial Naive Bayes, we computed the confusion matrix and draw the heatmap to visualize the matrix. I found that the two most confused classes are **soc.religion.christian** and **talk.religion.misc**

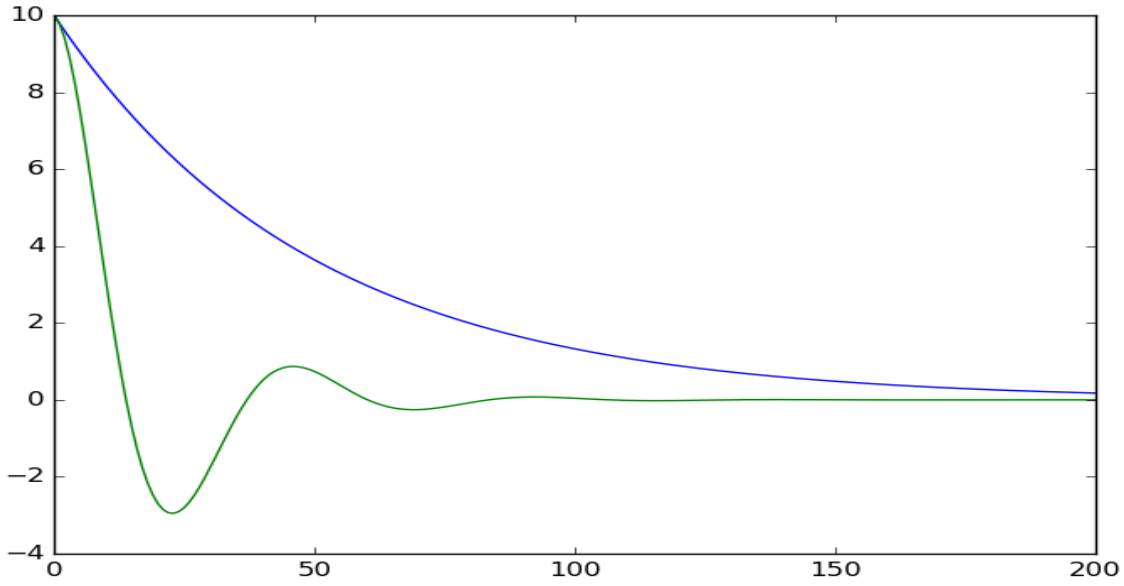
```
[141.  4.  4.  0.  0.  0.  0.  1.  6.  5.  4.  1.  1.  4.  5.  8.  5.  12.  16.  31. ]
[  1. 278.  26.  11.  11.  46.   3.   1.   2.   3.   0.  12.  11.   6.   7.   3.   0.   2.   2.   3. ]
[  2.  10. 205.  27.   7.  14.   1.   1.   1.   0.   0.   5.   9.   0.   1.   1.   0.   0.   0.   2. ]
[  2.  16.  65. 279.  33.   7.  30.   1.   1.   0.   0.   3.  28.   1.   1.   0.   1.   1.   0.   1. ]
[  1.  16.  10.  32. 268.   6.  21.   0.   2.   0.   0.   3.  11.   0.   0.   1.   0.   0.   0.   0. ]
[  2.  24.  24.   2.   3. 293.   0.   0.   2.   1.   1.   1.   2.   0.   2.   1.   1.   1.   0.   0. ]
[  0.   4.   3.   9.   8.   4. 280.   8.   5.   5.   0.   1.   8.   1.   0.   0.   1.   0.   1.   0. ]
[  3.   0.   1.   4.   6.   0.  15. 289.  28.   0.   1.   0.  12.   7.   6.   0.   6.   1.   5.   2. ]
[  5.   3.   5.   0.   2.   0.   7.  31. 292.   4.   2.   4.   9.   3.   2.   1.   3.   4.   2.   4. ]
[  2.   1.   0.   0.   0.   1.   2.   0.   2. 321.   5.   3.   1.   0.   1.   1.   1.   2.   1.   1. ]
[ 10.   5.  16.   8.  15.   5.  11.  24.  13.  30. 373.  16.  11.  16.  18.  14.  11.  10.   7.   7. ]
[  4.  11.  12.   3.   6.   7.   1.   4.   0.   4.   3. 298.  33.   0.   3.   1.  11.   3.   5.   4. ]
[  1.   3.   2.  17.  15.   5.   8.   9.   8.   1.   0.   3. 228.   5.   6.   0.   1.   1.   2.   2. ]
[  2.   1.   3.   0.   2.   2.   1.   3.   6.   4.   1.   1.  13. 309.   5.   1.   4.   0.  10.   7. ]
[ 10.   7.   7.   0.   6.   3.   6.   6.   4.   3.   2.   5.  11.   6. 311.   2.   8.   0.   7.   5. ]
[ 79.   4.   2.   0.   1.   1.   1.   3.   6.   5.   6.   7.   2.  19.   6. 354.  14.  19.  12.  96. ]
[ 12.   0.   0.   0.   2.   1.   2.   6.  10.   3.   1.  19.   0.   9.   4.   2. 267.   8.  91.  22. ]
[ 13.   2.   1.   0.   0.   0.   0.   2.   3.   2.   0.   6.   2.   3.   7.   0.   8. 303.   9.   9. ]
[ 10.   0.   5.   0.   0.   0.   1.   7.   6.   6.   0.   7.   1.   5.   8.   2.  11.   9. 138.   8. ]
[ 19.   0.   3.   0.   0.   0.   0.   0.   1.   0.   0.   1.   0.   2.   1.   6.  11.   0.   2.  47. ]
```



Question 2

2.1 - SGD with Momentum

Please see ***q2.py*** for the implementation.



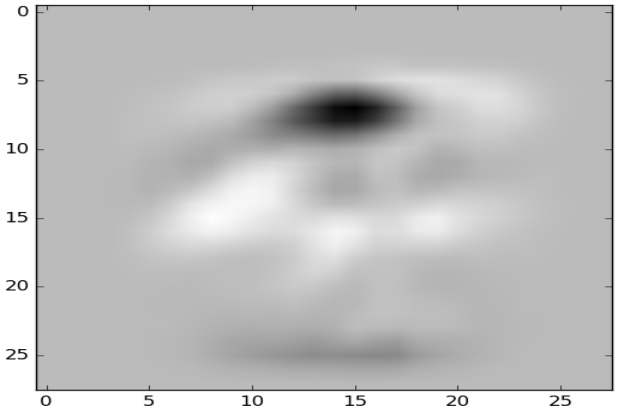
The green line represents $\beta = 0.9$ and blue line represents $\beta = 0.0$. We found the minimum of $f(w)$ occurs at $w = 0$ for both green line and blue line.

2.2 - Training SVM

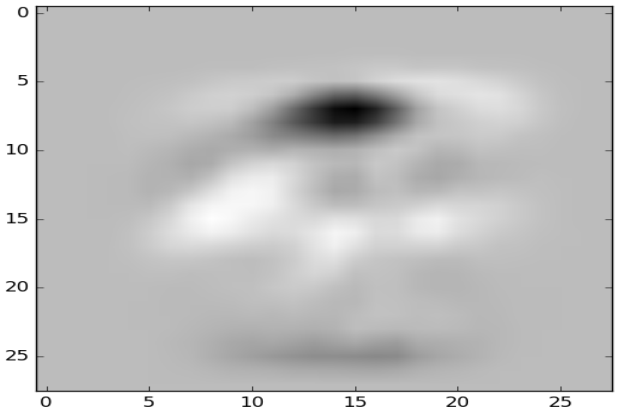
We implemented the SVM objective, gradients, and prediction in ***q2.py***.

2.3 - Apply on 4-vs-9 digits on MNIST

When momentum $\beta = 0$:
The training loss: 0.40345
The test loss: 0.40699
The classification accuracy on the training set: 91.147%
The classification accuracy on the test set: 91.331%
Plot w as a 28 by 28 image: Please see ***q2.py*** for the implementation.



When momentum $\beta = 0.1$:
The training loss: 0.3562
The test loss: 0.34429
The classification accuracy on the training set: 90.1497%
The classification accuracy on the test set: 90.1704%
Plot w as a 28 by 28 image: Please see ***q2.py*** for the implementation.



Question 3

3.1 - Positive semidefinite and quadratic form

Prove that a symmetric matrix $K \in \mathbb{R}^{d\times d}$ is positive semidefinite iff for all vectors $\mathbb{x} \in \mathbb{R}^d$ we have $\mathbb{x}^T K\mathbb{x} \geq 0$.

Solution:

Since this is an "if and only if" statement, we need to show both directions.

$\Longrightarrow$ Suppose that a symmetric matrix $K \in \mathbb{R}^{d\times d}$ is positive semidefinite. This will be equivalent to say that the symmetric matrix K has no negative eigenvalues. We want to show for all vectors $\mathbb{x} \in \mathbb{R}^d$, we have $\mathbb{x}^T K\mathbb{x} \geq 0$. By the spectral decomposition, the symmetric matrix K can be factorized as $K = Q\Lambda Q^T$, where Q is the square(D by D) matrix whose i^{th} column is the eigenvector of K and Λ is the diagonal matrix whose diagonal elements are the corresponding eigenvalues,(i.e. $\Lambda_{ii} = \lambda_i$)

$$\begin{aligned} K &= Q\Lambda Q^T \\ &= Q \begin{bmatrix} \lambda_1 & 0 & 0 & \dots & 0 \\ 0 & \lambda_2 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \lambda_d \end{bmatrix} Q^T \end{aligned}$$

Then, let $\mathbb{u} = Q^T\mathbb{x}$, then $\mathbb{x}^T K\mathbb{x} = \mathbb{x}^T Q\Lambda Q^T\mathbb{x} = \mathbb{u}^T\Lambda\mathbb{u}$. We know $\mathbb{x}$ is an $d \times 1$ matrix, and Q is an $d \times d$ matrix. So, $\mathbb{u}$ will be $d \times 1$ matrix, and so $\mathbb{u}^T$ will be $1 \times d$ matrix.

Thus $\mathbb{u}^T\Lambda\mathbb{u} = \sum_{i=1}^d \lambda_i u_i^2$. we know all eigenvalues(λ_i are non-negative, and all u_i^2 are non-negative as well.

Therefore, $\mathbb{x}^T K\mathbb{x} = \mathbb{x}^T Q\Lambda Q^T\mathbb{x} = \mathbb{u}^T\Lambda\mathbb{u} = \sum_{i=1}^d \lambda_i u_i^2 \geq 0$

$\Longleftarrow$ Suppose that a symmetric matrix $K \in \mathbb{R}^{d\times d}$, for all vectors $x \in \mathbb{R}^d$, we have $\mathbb{x}^T K\mathbb{x} \geq 0$. We want to show symmetric matrix $K \in \mathbb{R}^{d\times d}$ is positive semidefinite.

Let $\mathbb{x}$ be arbitrary vectors. By spectral decomposition, $K = Q\Lambda Q^T$, where Q and Λ are defined same as above. Let $\mathbb{u} = Q^T\mathbb{x}$. Then, we know $\mathbb{x}^T K\mathbb{x} = \mathbb{x}^T Q\Lambda Q^T\mathbb{x} = \mathbb{u}^T\Lambda\mathbb{u} = \sum_{i=1}^d \lambda_i u_i^2 \geq 0$. Since this inequality relation has to hold for all vector $\mathbb{x}$. We have to ensure all $\lambda_i u_i^2$ are non-negative, so all λ_i have to be non-negative. Then,we can conclude that the symmetric matrix K has no negative eigenvalues. Therefore, symmetric matrix $K \in \mathbb{R}^{d\times d}$ is positive semidefinite.

3.2 - Kernel Properties

Prove the following properties:

1. The function k($\mathbb{x}$,$\mathbb{y}$) = α is a kernel for $\alpha > 0$

Solution:

We need to proof k($\mathbb{x}$,$\mathbb{y}$) = α is a kernel by finding an embedding $\phi(x)$ such that $k(\mathbb{x}, \mathbb{y}) = \langle\phi(\mathbb{x}), \phi(\mathbb{y})\rangle$.

Let $\phi(x) = \sqrt{\alpha}$, so $\phi(y) = \sqrt{\alpha}$ as well.

Then k($\mathbb{x}$,$\mathbb{y}$) = $\langle\phi(\mathbb{x}), \phi(\mathbb{y})\rangle = (\sqrt{\alpha})^T \cdot (\sqrt{\alpha}) = \alpha$ when $\alpha > 0$

Therefore, $k(x, y) = \alpha$ is a kernel for $\alpha > 0$

2. k($\mathbb{x}$,$\mathbb{y}$) = $f(\mathbb{x}) \cdot f(\mathbb{y})$ is a kernel for all f: $\mathbb{R}^d \to \mathbb{R}$.

Solution:

We need to proof k($\mathbb{x}$,$\mathbb{y}$) = $f(x)\cdot f(y)$ is a kernel by finding an embedding $\phi(x)$ such that $k(\mathbb{x}, \mathbb{y}) = \langle\phi(\mathbb{x}), \phi(\mathbb{y})\rangle$.

Let $\phi(\mathbb{x}) = f(\mathbb{x})$, so $\phi(\mathbb{y}) = f(\mathbb{y})$ because the function f mapping from $\mathbb{R}^d$ to $\mathbb{R}$, so $f(\mathbb{x})$ and $f(\mathbb{y})$ are real

numbers.
Then k(𝔵,𝔶) $= \langle \phi(\mathfrak{x}), \phi(\mathfrak{y}) \rangle = f(x)^T \cdot f(y) = f(x) \cdot f(y)$ because $f(x)^T = f(x)$

Therefore, $k(x, y) = f(\mathfrak{x}) \cdot f(\mathfrak{y})$ is a kernel for all f: $\mathbb{R}^d \to \mathbb{R}$.

3. If $k_1(\mathfrak{x}, \mathfrak{y})$ and $k_2(\mathfrak{x}, \mathfrak{y})$ are kernels then $k(\mathfrak{x}, \mathfrak{y}) = a \cdot k_1(\mathfrak{x}, \mathfrak{y}) + b \cdot k_2(\mathfrak{x}, \mathfrak{y})$ for $a, b > 0$ is a kernel.
Solution:
We need to show for all $\mathfrak{x}^{(1)}$, $\mathfrak{x}^{(2)}$,...$\mathfrak{x}^{(n)}$ the gram matrix $K_{ij} = K(\mathfrak{x}^{(i)}, \mathfrak{x}^{(j)})$ is positive semi-definite.
Based on the question, we know $k_1(\mathfrak{x}, \mathfrak{y})$ and $k_2(\mathfrak{x}, \mathfrak{y})$ are kernels. Then the gram matrix for k_1 and k_2 are positive semi-definite.
Then, we have:
$\mathfrak{x}^T a K_1 \mathfrak{x} = a \mathfrak{x}^T K_1 \mathfrak{x} \geq 0$ because the K_1 are positive semi-definite and $a > 0$
$\mathfrak{x}^T b K_2 \mathfrak{x} = b \mathfrak{x}^T K_2 \mathfrak{x} \geq 0$ because the K_2 are positive semi-definite and $b > 0$
Then, $\mathfrak{x}^T (aK_1 + bK_2) \mathfrak{x} = a \mathfrak{x}^T K_1 \mathfrak{x} + b \mathfrak{x}^T K_2 \mathfrak{x} \geq 0$ Then, the gram matrix for $k(\mathfrak{x}, \mathfrak{y})$ is positive semi-definite by the theorem in 3.1.
Therefore, $k(\mathfrak{x}, \mathfrak{y}) = a \cdot k_1(\mathfrak{x}, \mathfrak{y}) + b \cdot k_2(\mathfrak{x}, \mathfrak{y})$ for $a, b > 0$ is a kernel.

4. If $k_1(\mathfrak{x}, \mathfrak{y})$ is a kernel then $k(\mathfrak{x}, \mathfrak{y}) = \frac{k_1(\mathfrak{x}, \mathfrak{y})}{\sqrt{k_1(\mathfrak{x}, \mathfrak{x})}\sqrt{k_1(\mathfrak{y}, \mathfrak{y})}}$ is a kernel (hint: use the features ϕ such that $k_1(\mathfrak{x}, \mathfrak{y})$ $= \langle \phi(\mathfrak{x}), \phi(\mathfrak{y}) \rangle$
Solution:
We need to show for all $\mathfrak{x}^{(1)}$, $\mathfrak{x}^{(2)}$,...$\mathfrak{x}^{(n)}$ the gram matrix $K_{ij} = K(\mathfrak{x}^{(i)}, \mathfrak{x}^{(j)})$ is positive semi-definite.
Since $k_1(\mathfrak{x}, \mathfrak{y})$ is a kernel. We have $k_1(\mathfrak{x}, \mathfrak{y}) = \langle \phi(\mathfrak{x}), \phi(\mathfrak{y}) \rangle$, then we have
$k_1(\mathfrak{x}, \mathfrak{x}) = \langle \phi(\mathfrak{x}), \phi(\mathfrak{x}) \rangle = \|\phi(\mathbf{x})\|^2$
$k_1(\mathfrak{y}, \mathfrak{y}) = \langle \phi(\mathfrak{y}), \phi(\mathfrak{y}) \rangle = \|\phi(\mathbf{y})\|^2$
So $\sqrt{k_1(\mathfrak{x}, \mathfrak{x})} = \|\phi(\mathbf{x})\|$ and $\sqrt{k_1(\mathfrak{y}, \mathfrak{y})} = \|\phi(\mathbf{y})\|$ We know $\|\phi(\mathbf{x})\|$ and $\|\phi(\mathbf{y})\|$ are real numbers and greater than 0. In addition, the gram matrix for $k_1(\mathfrak{x}, \mathfrak{y})$ is positive semi-definite. Thus for all $\mathfrak{x}^{(1)}$,...$\mathfrak{x}^{(n)}$, $\mathfrak{x}^T K_1 \mathfrak{x} \geq 0$.
This implies $\mathfrak{x}^T \frac{K_1}{\|\phi(\mathbf{x})\| \|\phi(\mathbf{y})\|} \mathfrak{x} \geq 0$, where $K(\mathfrak{x}, \mathfrak{y}) = \frac{K_1}{\|\phi(\mathbf{x})\| \|\phi(\mathbf{y})\|}$.
Therefore, the gram matrix for $k(\mathfrak{x}, \mathfrak{y})$ is positive semi-definite by the theorem in 3.1. Therefore, $k(\mathfrak{x}, \mathfrak{y})$ is a kernel.