CSC411 Assignment 3

Yanzhi Zhang

December 2017

1 20 Newsgroups predictions

1.1 Train and test loss of three training model

Among all the models that I used to train my model, three models stand out to be performing the best, and they are logistic regression, support vector machine and neural network.

The following are the performance of the model.
BernoulliNB baseline train accuracy = 0.5987272405868835
BernoulliNB baseline test accuracy = 0.4579129049389272
LogisticRegression train accuracy = 0.9705674385716812
LogisticRegression test accuracy = 0.6167020711630377
SVM train accuracy = 0.8162453597313063
SVM test accuracy = 0.5450079660116834
Neural network train accuracy = 0.8877496906487538
Neural network test accuracy = 0.5382368560807222

1.2 How to pick the best hyper-parameter

I have used two ways of choosing the hyper-parameters. One is that manual search. Using this method. I will firstly try to guess a range of value that could improve the performance before manually tuning the parameters. Otherwise, the performance will be drastically awful if I just blindly tune some parameters.For example, in logistic regression I already knew that the data set is very sparse the flat, so setting the tolerance too small like "1e-5" will not necessarily boost the accuracy of the prediction, instead, it will definitely slow the training down. Therefore, I set the tolerance to be "1e-2", which not only speed up the training process but also have a relatively high accuracy.

The other method I used is grid search which is sequentially picking value from a selective range of values. For example, we knew that in SVM, doing too many of iterations will slow the process down by a lot, and a small number of iteration will speed up the process but compromise the accuracy. Therefore, I tried out the iteration between 500 and 1000 with a step of 100 and pick the one with best results.

1.3 Explain why you picked these 3 methods, did they work as you thought? Why/Why not?

The reason that Bernoulli naive Bayes classifier has such a low accuracy is that it over-simplified the model by assuming the data is binary.

In order to better predict the data set, we have to use something more sophisticated. The first thing I think about is using logistic regression, since it is the most widely used model and it works well almost all the time. The second the model I used is SVM since it the most effective model for NLP problem by far. The last one is the neural network which I think will fit into the this problems since it has a lot of potential in solving any kinds of problem including this one.

As a result, the logistic regression performs the best in all three models, which I did not expect this result. At first, I thought the best model would be SVM and next is neural network and logistic regression would be the worst. As the words and the thesis of essay usually have a strong a relationship with each others. For example, "Christian" is a strong indication that this essay should be classified as something related to religions.

The reason that SVM perform so poorly might be because I did not choose the optimal parameter for SVM and can not produce good result and gradients, I might need to do a SVM with Gaussian kernel. I think the extremely high feature dimension also contributes to the low accuracy of SVM therefore, we need to implement a better kernel function and give this model more time to train.

And for neural network, I think that I did not choose the proper number layers and internal neurons so it compromised the training time for accuracy. Same as SVM, neural network also suffers from the high feature space so we might need more internal layers and neuron.

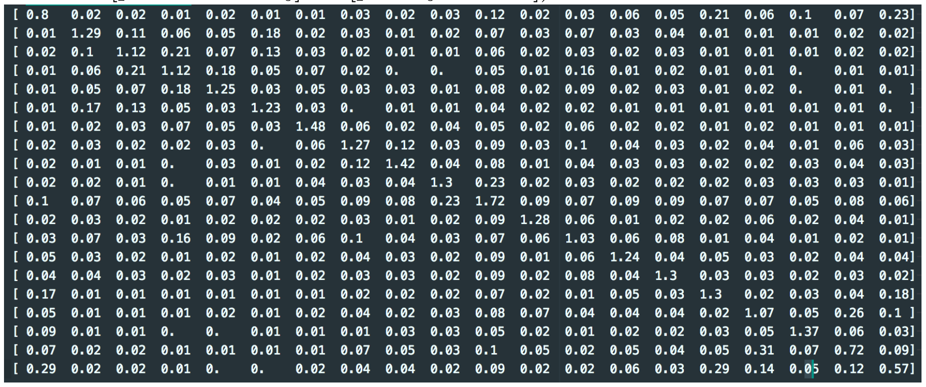
As the SVM and neural network performed so poorly, the logistic regression stands out as the best classifier.

1.4 Confusion matrix of Logistic regression model

```
[140 3 3 1 3 4 1 7 11 13 2 3 10 9 12 31 10 9 6 41]
[ 5 250 21 7 6 23 6 4 4 12 1 5 17 3 11 1 2 3 5 3]
[ 3 25 227 45 19 13 2 5 4 19 1 3 3 6 8 1 2 3 3 2]
[ 1 17 43 221 32 6 13 4 2 8 1 3 34 2 2 1 1 0 1 0]
[ 0 9 10 32 243 2 12 11 7 16 4 5 26 1 5 1 1 0 0 0]
[ 1 55 33 15 5 243 5 3 2 11 0 6 4 1 5 3 1 2 0 0]
[ 1 4 7 15 12 1 299 7 12 16 1 1 8 1 2 1 1 1 0 0]
[ 7 4 3 4 5 1 10 254 19 32 2 2 19 3 6 4 5 3 8 5]
[ 3 2 1 4 3 1 5 33 279 23 0 3 12 4 6 1 4 3 9 2]
[ 6 3 0 1 3 0 6 7 9 303 25 0 3 4 5 5 4 4 7 2]
[ 5 2 2 0 0 0 3 5 6 31 326 2 1 2 3 1 3 0 2 5]
[ 3 10 7 3 6 2 4 4 7 26 2 255 18 5 8 2 13 5 8 8]
[ 6 14 15 29 19 8 16 16 11 18 1 13 199 9 11 3 1 2 2 0]
[ 5 14 2 3 4 2 8 14 14 19 1 4 10 255 7 11 9 4 6 4]
[ 8 16 3 4 4 2 1 9 6 24 3 4 16 8 258 5 9 2 7 5]
[ 27 3 1 5 1 0 3 4 5 16 2 1 3 8 4 270 1 9 6 29]
[ 9 4 3 2 3 2 2 11 13 21 1 15 2 7 7 9 201 9 25 18]
[ 30 1 0 3 1 0 0 7 8 14 0 6 3 4 2 13 13 242 20 9]
[ 16 2 0 3 0 2 0 9 10 14 4 5 4 7 8 7 78 15 112 14]
[ 34 5 2 3 3 2 3 5 4 11 3 5 1 10 8 44 18 11 11 68]
```

1.5 What were the two classes your classifier was most confused about?

Since logistic regression had the highest accuracy on my model, I used this model to predict the data and compute the confusion matrix. Since the number of data sets in each class is different, we have to normalize the matrix before we actually compute the two most confused classes. Therefore, I normalize each class by divide the the sum of each class, and take the maximum number of errors between two different classes (i.e. take the maximum of [pred=i label=j] + [pred=j label=i]) and round to the 2 decimals.

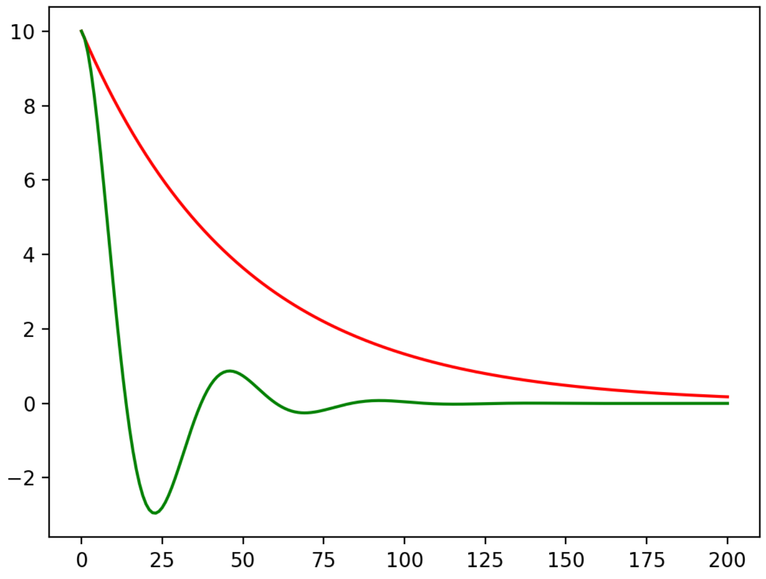


```
[ 0.8   0.02  0.02  0.01  0.02  0.01  0.01  0.03  0.02  0.03  0.12  0.02  0.03  0.06  0.05  0.21  0.06  0.1   0.07  0.23]
[ 0.01  1.29  0.11  0.06  0.05  0.18  0.02  0.03  0.01  0.02  0.07  0.03  0.07  0.03  0.04  0.01  0.01  0.01  0.02  0.02]
[ 0.02  0.1   1.12  0.21  0.07  0.13  0.03  0.02  0.01  0.01  0.06  0.02  0.03  0.02  0.03  0.01  0.01  0.01  0.02  0.02]
[ 0.01  0.06  0.21  1.12  0.18  0.05  0.07  0.02  0.    0.    0.05  0.01  0.16  0.01  0.02  0.01  0.01  0.    0.01  0.01]
[ 0.01  0.05  0.07  0.18  1.25  0.03  0.05  0.03  0.03  0.01  0.08  0.02  0.09  0.02  0.03  0.01  0.02  0.    0.01  0.  ]
[ 0.01  0.17  0.13  0.05  0.03  1.23  0.03  0.    0.01  0.01  0.04  0.02  0.02  0.01  0.01  0.01  0.01  0.01  0.01  0.  ]
[ 0.01  0.02  0.03  0.07  0.05  0.03  1.48  0.06  0.02  0.04  0.05  0.02  0.06  0.02  0.02  0.01  0.02  0.01  0.01  0.01]
[ 0.02  0.03  0.02  0.02  0.03  0.    0.06  1.27  0.12  0.03  0.09  0.03  0.1   0.04  0.03  0.02  0.04  0.01  0.06  0.03]
[ 0.02  0.01  0.01  0.    0.03  0.01  0.02  0.12  1.42  0.04  0.08  0.01  0.04  0.03  0.03  0.02  0.02  0.03  0.04  0.03]
[ 0.02  0.02  0.01  0.    0.01  0.01  0.04  0.03  0.04  1.3   0.23  0.02  0.03  0.02  0.02  0.02  0.03  0.03  0.03  0.01]
[ 0.1   0.07  0.06  0.05  0.07  0.04  0.05  0.09  0.08  0.23  1.72  0.09  0.07  0.09  0.09  0.07  0.07  0.05  0.08  0.06]
[ 0.02  0.03  0.02  0.01  0.02  0.02  0.02  0.03  0.01  0.02  0.09  1.28  0.06  0.01  0.02  0.02  0.06  0.02  0.04  0.01]
[ 0.03  0.07  0.03  0.16  0.09  0.02  0.06  0.1   0.04  0.03  0.07  0.06  1.03  0.06  0.08  0.01  0.04  0.01  0.02  0.01]
[ 0.05  0.03  0.02  0.01  0.02  0.01  0.02  0.04  0.03  0.02  0.09  0.01  0.06  1.24  0.04  0.05  0.03  0.02  0.04  0.04]
[ 0.04  0.04  0.03  0.02  0.03  0.01  0.02  0.03  0.03  0.02  0.09  0.02  0.08  0.04  1.3   0.03  0.03  0.02  0.03  0.02]
[ 0.17  0.01  0.01  0.01  0.01  0.01  0.01  0.02  0.02  0.02  0.07  0.02  0.01  0.05  0.03  1.3   0.02  0.03  0.04  0.18]
[ 0.05  0.01  0.01  0.01  0.02  0.01  0.02  0.04  0.02  0.03  0.08  0.07  0.04  0.04  0.04  0.02  1.07  0.05  0.26  0.1 ]
[ 0.09  0.01  0.01  0.    0.    0.01  0.01  0.01  0.03  0.03  0.05  0.02  0.01  0.02  0.02  0.03  0.05  1.37  0.06  0.03]
[ 0.07  0.02  0.02  0.01  0.01  0.01  0.01  0.07  0.05  0.03  0.1   0.05  0.02  0.05  0.04  0.05  0.31  0.07  0.72  0.09]
[ 0.29  0.02  0.02  0.01  0.    0.    0.02  0.04  0.04  0.02  0.09  0.02  0.02  0.06  0.03  0.29  0.14  0.05  0.12  0.57]
```

Ignoring the data on the main diagonal, which are the correct prediction, then maximal of the rest is at [19, 17] which is 0.31. Therefore, the two most confused classes are class 19 and class 16 namely "talk.politics.misc" and "talk.politics.guns".

2 Training SVM with SGD

2.1 SGD With Momentum

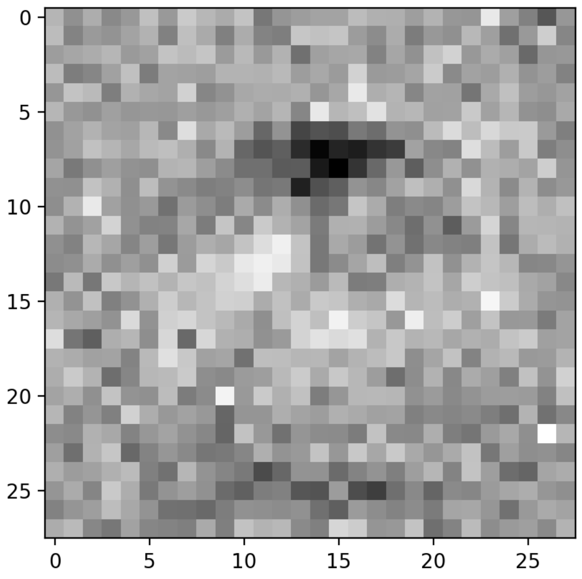


The green line represent that beta = 0.9 and the red line represent that beta = 0.

2.2 Apply on 4-vs-9 digits on MNIST

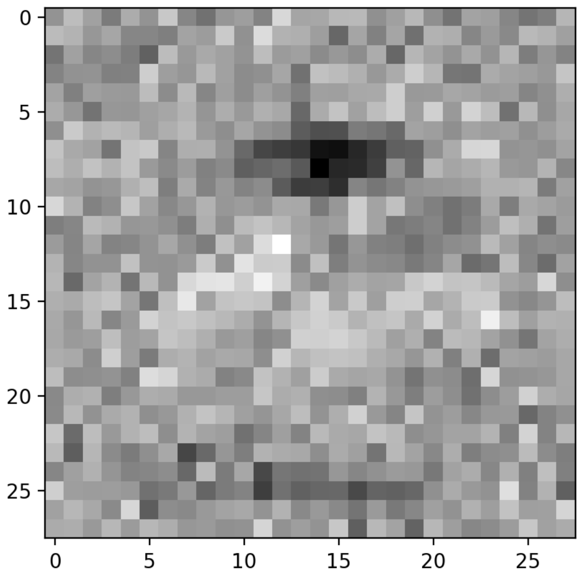
——————————————-

When beta = 0
The training set accuracy is: 0.962630385488
The test set accuracy is: 0.96046427276
The hinge loss of training set is : 0.110749068726
The hinge loss of test set is : 0.114090695288



_____________________________-

```
When beta = 0.1
The training set accuracy is: 0.961360544218
The test set accuracy is: 0.961915125136
The hinge loss of training set is : 0.111436179692
The hinge loss of test set is : 0.113295678308
```


0
5
10
15
20
25
0
5
10
15
20
25

3 Kernels

3.1 Positive semi-definite and quadratic form

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

3.1: 1° "$\Rightarrow$" Given that k is a positive semidifinite matrix

we want to prove that $\forall x \in R^d \quad x^T K x \geqslant 0$

Sine k is symmetric $\Rightarrow$ K has no negative eigenvalue.

$\Rightarrow K = Q^T \Lambda Q$, Λ is a diagnal matrix withall its entries non-negative. and $Q^T Q = I$

$\Rightarrow \Lambda = \Lambda^{\frac{1}{2}} \cdot \Lambda^{\frac{1}{2}}$

$\Rightarrow \forall x \in R$, $x^T k x = x^T Q^T \Lambda Q x = x^T Q^T \Lambda Q Q^T \Lambda Q x \geqslant 0$ $\blacksquare$

2° "$\Leftarrow$" Given that $\forall x \in R^d$, $x^T K x \geqslant 0$

we want to prove K is positive semidifinite.

K is a symmetric matrix such that it has eigenvector x and eigenvalue a : $Kx = \lambda x$

and since $x^T k x = \lambda x^T x \geqslant 0$.

$\Rightarrow$ K is positive semidifinite.

In sum, K is psd $\Leftrightarrow \forall x \in R^d$, $x^T A x \geqslant 0$ $\blacksquare$

3.2 Kernel properties

3.2.1 The function $k(x, y) = \alpha$ is a kernel for $\alpha \geqslant 0$.

1. let $\phi(x) = \sqrt{\alpha}$, $\alpha \geqslant 0$.

$K(x,y) = \langle \phi(x), \phi(y) \rangle = \sqrt{\alpha} \cdot \sqrt{\alpha} = \alpha$

$\Rightarrow K(x,y) = \alpha$ is a well-defined embedding function.

$\Rightarrow K(x,y) = \alpha$ is a kernel for $\alpha \geqslant 0$.

3.2.2 $k(x, y) = f(x) \cdot f(y)$**is a kernel for all** $f : R^d \to R$**.**

2. let $\phi(x) = f(x)$ for all $f: R^d \to R$

$k(x, y) = \langle \phi(x), \phi(y) \rangle = \langle f(x), f(y) \rangle = f(x)f(y)$

$\Rightarrow k(x, y) = f(x)f(y)$ is a well-defined embedding function

$\Rightarrow k(x, y) = f(x)f(y)$ is a a kernal for all $f: R^d \to R$

3.2.3 **If** $k_1(x, y)$ **and** $k_2(x, y)$ **are kernels then** $k(x, y) = a \cdot k1(x, y) + b \cdot k2(x, y)$ **for** $a, b > 0$ **is a kernel.**

3. Given that $k_1(x,y)$, $k_2(x,y)$ are kernels.

① prove that $k'(x,y) = a k_1(x,y)$ is a kernel:

let $k_1 = \langle \phi(x), \phi(y) \rangle$, $k'(x,y) = \langle \sqrt{a}\phi'(x), \sqrt{a}\phi'(y) \rangle = \sqrt{a} \cdot \sqrt{a} \langle \phi(x), \phi(y) \rangle =$

$= a \langle \phi(x), \phi(y) \rangle = a k_1(x,y)$

$\Rightarrow k' = a k_1(x,y)$

② prove that $k(x,y) = k_1(x,y) + k_2(x,y)$ is a kernel

let k_1, k_2 be positive semidefinite matrix on $R^{d \times d}$

then, $\forall x \in R^d$ $x^T k_1 x \geq 0$, $x^T k_2 x \geq 0$

$x^T k_1 x + x^T k_2 x = x^T (k_1 + k_2) x \geq 0$

$\Rightarrow$ by 3.1 $(k_1 + k_2)$ is a positive semidefinite. matrix

$\Rightarrow$ $k(x,y) = k_1(x,y) + k_2(x,y)$ is a well-defined kernel.

by ①, ②, $k(x,y) = a k_1(x,y) + b(x,y)$, $a, b > 0$ is a well-defined kernel.

3.2.4 If $k_1(x, y)$ is a kernel then $k(x, y) = \frac{k_1(x,y)}{\sqrt{k_1(x,y)}\sqrt{k_1(x,y)}}$ is a kernel (hint: use the features ϕ such that $k_1(x, y) = \langle \phi(x), \phi(y) \rangle$).

4. Given that $k_1(x,y)$ is a kernel

let $\phi_1(x)$ be $k_1(x,y)$ embedding function

$$k_1(x,x) = \langle \phi_1(x), \phi_1(x) \rangle = \phi_1(x) \cdot \phi_1(x)$$

$\Rightarrow$ by definition of inner product $\sqrt{k_1(x,x)} = \phi_1(x)$

let $\phi(x)$ be emedding function of $k(x,y)$

$$\phi(x) = \frac{\phi_1(x)}{\|\phi_1(x)\|}$$

$$k(x,y) = \langle \phi(x), \phi(y) \rangle = \left\langle \frac{\phi_1(x)}{\|\phi_1(x)\|}, \frac{\phi_1(y)}{\|\phi_1(y)\|} \right\rangle$$

$$= \|\phi_1(x)\| \|\phi_1(y)\| \langle \phi_1(x), \phi_1(y) \rangle$$

$$= \frac{k_1(x,y)}{\sqrt{k_1(x,x)}\sqrt{k_2(y,y)}} = k(x,y)$$

$\Rightarrow k(x,y)$ has an embedding function $\phi(x) = \frac{\phi_1(x)}{\sqrt{\phi_1(x)}}$

$\Rightarrow k(x,y)$ is a kernel.