CSC411 – A3

YUHENG HUANG

Dec 2017

I. 20 Newsgroups predictions report:

1. Bernoulli Naive Bayes (Baseline):

Accuracy on training set: ***0.5987272405868835***

Accuracy on test set: ***0.4579129049389272***

2. Multinomial Naive Bayes:

Accuracy on training set: ***0.9589004772847799***

Accuracy on test set: ***0.7002124269782263*** ***(best performance)***

Alpha: ***0.01***

Q: how to pick the hyperparameter Alpha:

A: I use cross validation ***(kfold = 10)*** to pick the best hyperparameter **alpha** for Multinomial Naive Bayes. First, I select alpha's range from **0.01 to 1**, and then I uniformly random generate 100 samples of alpha. Second, I put all those 100-different alphas into cross validation and then calculate its mean score (cross validation will return me 10 scores for each alpha). After I get 100 mean scores, I will pick the highest score as the best hyperparameter alpha.

Q: Why I pick this method?

A: Because the classifier that we are going to create is 20 classes classifier, I think it is not belong to Bernoulli, but Multinomial. Therefore, I decide to try Multinomial Naïve Bayes, and they work just as I thought, much better than Bernoulli Naive Bayes.

**For detail, please visit q1.py*

3. **Linear SVM Classifier:**

Accuracy on training set: ***0.96712038182782 39***

Accuracy on test set: ***0.6972915560276155***

C: ***0.53***

how to pick the hyperparameter C:

I use cross validation ***(kfold = 5)*** to pick the best hyperparameter **C** for Linear SVM Classifier. First, I select alpha's range from **0.01 to 3**, and then I uniformly random generate 10 samples of alpha. Second, I put all those 10-different C into cross validation and then calculate its mean score (cross validation will return me 5 scores for each C). After I get 10 mean scores, I will pick the highest score as the best hyperparameter C.

Q: Why I pick this method?

A: SVM is very Effective in high dimensional spaces, so I decide to give it a try. And the result is not bad too, just like what I expected. However, the training process is a little bit long.

**For detail, please visit q1.py*

4. **Logistic Regression:**

Accuracy on training set: ***0.9399858582287431***

Accuracy on test set: ***0.6836165693043016***

C: ***2.19***

how to pick the hyperparameter C:

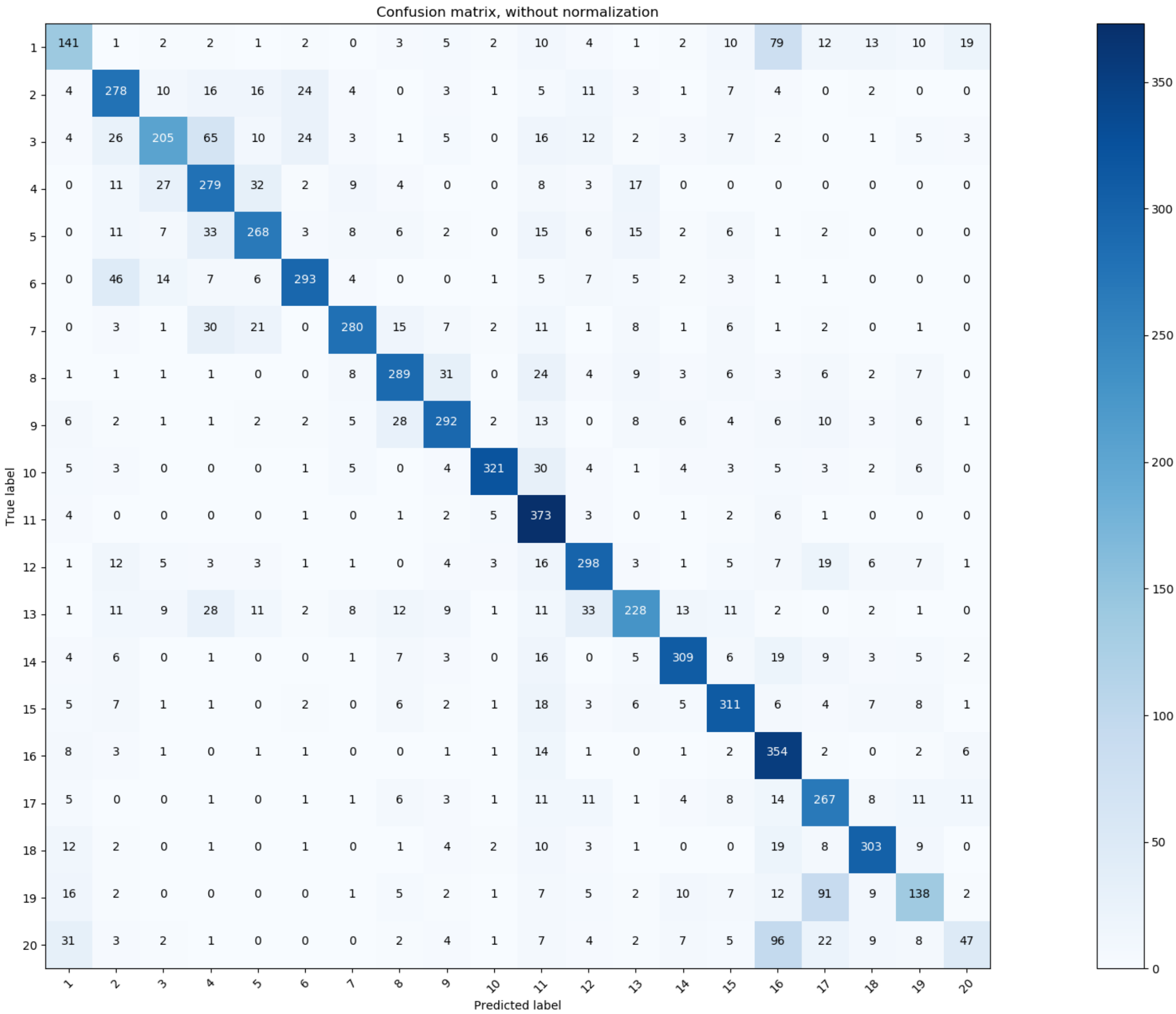
I use cross validation ***(kfold = 5)*** to pick the best hyperparameter **C** for Logistic Regression Classifier. First, I select alpha's range from **0.01 to 3**, and then I uniformly random generate 10 samples of alpha. Second, I put all those 10-different C into cross validation and then calculate its mean score (cross validation will return me 5 scores for each C). After I get 10 mean scores, I will pick the highest score as the best hyperparameter C.

Q: Why I pick this method?

A: Because Logistic Regression may handle non linear effects, and its result is as what I thought, the accuracy is higher than the baseline, but lower than SVM and Multinomial Naive Bayes.

**For detail, please visit q1.py*

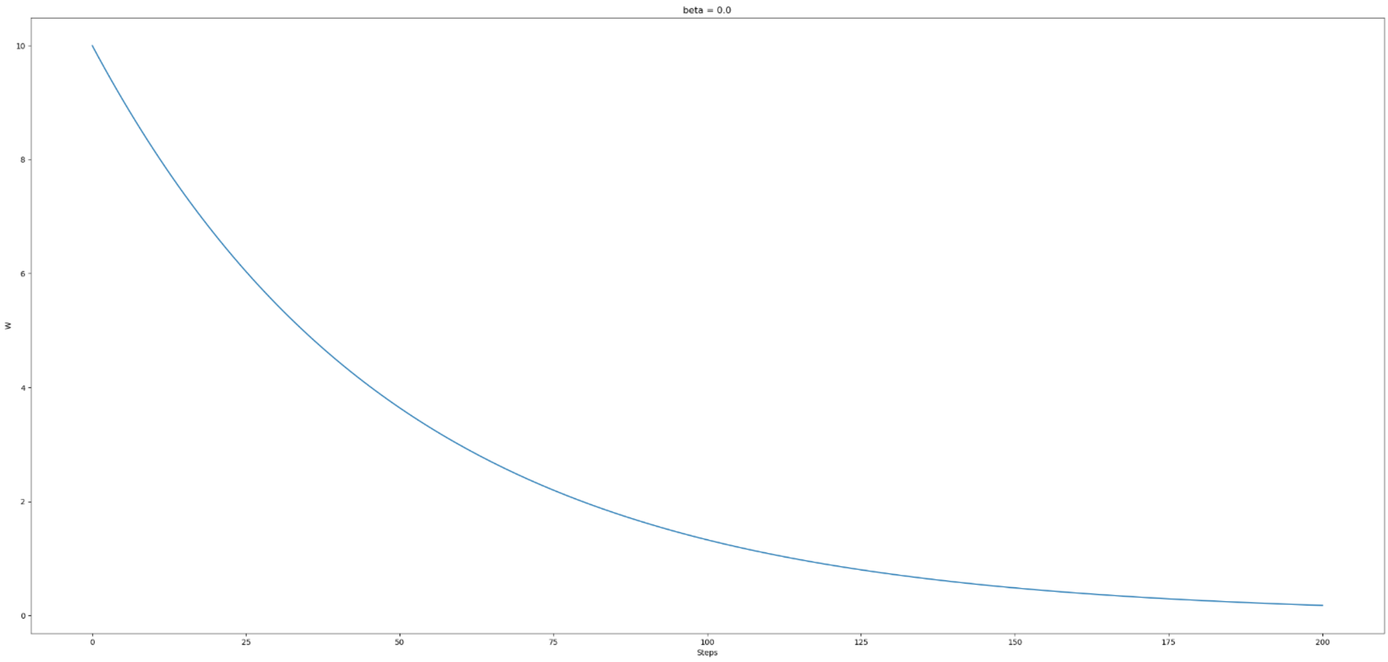
5. Confusion matrix for Multinomial Naive Bayes



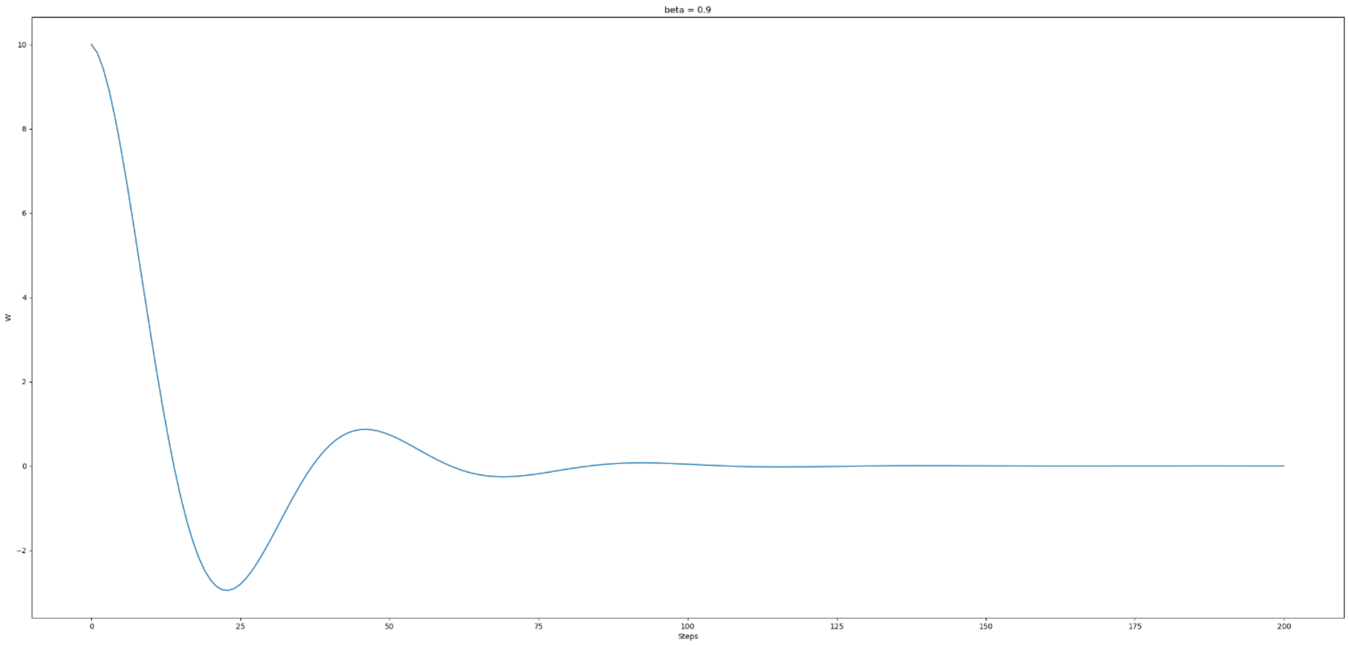
As the above confusion matrix shows, **16th class and 20th class** are most confused.

II. Training SVM with SGD

2.1 SGD With Momentum:



Beta = 0.0



Beta = 0.9

2.2 Training SVM:

For detail, please visit q2.py

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

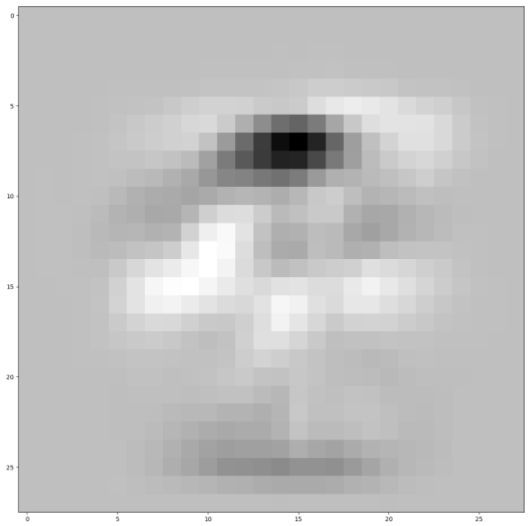
1) <u>model use β = 0:</u>

   The training loss: 0.342191688002

   The test loss: 0.337671534146

   Training accuracy = 0.9329705215419501

   Test accuracy = 0.9328980776206021



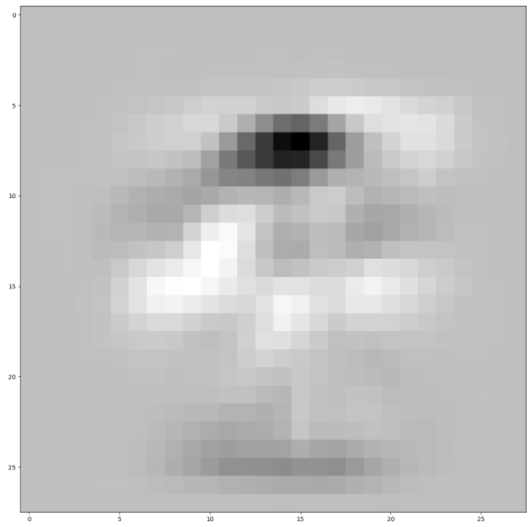
2) <u>model use β = 0.1:</u>

   The training loss: 0.342207800632

   The test loss: 0.337795184641

   Training accuracy = 0.9340589569160997

   Test accuracy = 0.9339862169024302



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

III. Kernels

3.1

$\because K \in R^{d\times d}$ and it is symmetric

$\because$ positive semidefinite ~~require~~ require all ~~ve~~ elements.

matrix

x have $x^T A x \geq 0$, if A is positive semidefinite

Matrix.

$\because$ all vectors $x \in R^d$ we have $x^T K x \geq 0$.

$\therefore K \in R^{d\times d}$ is positive semidefinite.

3.2.

1. a function $k(x,y)$ is kernel function have to.

Satisfies two properties:

1. symmetry: $k(x,y) = k(y,x)$
2. positive semi-definiteness.

for $k(x,y) = \alpha$, $\alpha > 0$;

first, $k(x,y) = \alpha$ and $k(y,x) = \alpha$.

and, because $\alpha > 0$. so $k(x,y)$ is positive semi-definiteness.

$\therefore$ $k(x,y)$ is kernel.

2.

$\because$ $k(x,y) = f(x) \cdot f(y)$

$\because$ $k(y,x) = f(y) \cdot f(x)$

$\therefore$ $k(x,y) = k(y,x)$

$\because$ $f: R^d \to R$.

$\therefore$ $f(x) \cdot f(y) : R^d \to R$

$\therefore$ $f(x) \cdot f(y)$ is positive semi-definiteness.

$\therefore$ $k(x,y)$ is symmetry and positive semi-definiteness

$\therefore$ $k(x,y) = f(x) \cdot f(y)$ is kernel.

3.

$\because$ $k(x,y) = a k_1(x,y) + b k_2(\cancel{y,x}\ x,y)$

$\because$ $k(y,x) = a k_1(y,x) + b k_2(y,x)$

$\therefore$ $k(x,y) = k(y,x)$ (symmetry)

$\because$ $a, b > 0$ and k_1, k_2 are kernel.

$\therefore$ $k(x,y) = a k_1(x,y) + b k_2(x,y)$ is positive semi-definiteness

$\therefore$ $k(x,y)$ is kernel