AI5213/EC4213/CT5303/AI4021: Machine learning and deep learning **Spring 2023**

Report for Coding Assignment #1

(due on Wed May. 10, 2023. 11 :59PM)

Instructor: Jeany Son
Student name (GIST ID# / GitHub ID#): Dongjun Nam (20195053 / june6423)

REPORT1

In bag of words model, we only consider the frequency of the words, ignoring the position of each words.

$$h_{NaiveBayes}(x) = arg\max_{y} P(y) \prod_{i \in position} P(w_i|y)$$

Prior and likelihood are calculated by following formula.

$$PriorP(Y) = \frac{N_y}{N_{doc}}$$

$$P(w_i|y_j) = \frac{count(w_i|y_j)}{\sum_{w_k \in V} count(w_k|y_j)}$$

There are 3 documents are labeled as ham and 3 documents are labeled as spam.
So, $P(Y = Ham) = \frac{3}{6}, P(Y = Spam) = \frac{3}{6}$
There are so many words in training set. I listed the 10 most frequent words (has largest likelihood) are shown on the table. Documents labeled as ham has 173 words and documents labeled as spam has 193 words.

word	frequency	likelihood
the	13	0.0751
to	8	0.0462
best	4	0.0231
class	4	0.0231
in	4	0.0231
of	4	0.0231
about	3	0.0173
am	3	0.0173
and	3	0.0173
dear	3	0.0173

Table 1 – Likelihood for class ham without Laplace smoothing

word	frequency	likelihood
you	12	0.0621
to	8	0.0414
the	5	0.0259
will	5	0.0259
can	4	0.0207
for	4	0.0207
it	4	0.0207
of	4	0.0207
your	4	0.0207
analysis	3	0.0155

Table 2 – Likelihood for class spam without Laplace smoothing

REPORT2

IRIS data set contains 4 features with 3 classes. This assignment recommended using the first two features and binary class, but I used 4 features with 3 classes.
Figure1 illustrates the results of logistic regression using first two features.

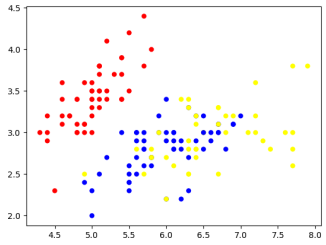

FIGURE 1 – Logistic regression result using first two features

Feature 3 and 4 are ignored in Figure1. So, I used PCA(Principal Component Analysis) to represent entire data in 2D image. I used off-the-shelf PCA solver to convert 4 dimension points into 2 dimension. PCA performs SVD(singular Value Decomposition) to get eigenvalue and eigenvectors of the covariance matrix. Then sort eigenvectors with their eigenvalue. Eigenvector with higher eigenvalue can explains more variance and it means more important. I performed PCA and take top-2 eigenvectors to express in 2 dimension image. To claim it's sufficient to use 2 dimension to express information of entire data, I calculated explained-variance-ratio per each eigenvector. Table 3 illustrates explained variance ratio per eigenvector. Top-2 eigenvector explains 99.76% of variance and it means it's sufficient to claim 2 dimension image with top-2 eigenvector can express entire data information.

Explained Variance Ratio			
0.92461872	0.05306648	0.01710261	0.00521218

TABLE 3 – Explained Variance Ratio per Eigenvector

eigenvector1	0.36138659	-0.08452251	0.85667061	0.3582892
eigenvector2	0.65658877	0.73016143	-0.17337266	-0.07548102
eigenvector3	-0.58202985	0.59791083	0.07623608	0.54583143
eigenvector4	-0.31548719	0.3197231	0.47983899	-0.75365743

TABLE 4 – Eigenvector sorted with their importance(eigenvalue)

Figure 2 shows the result of PCA with ground truth label. The red dot is class 0. The blue dot is class 1 and the yellow dot is class 2.

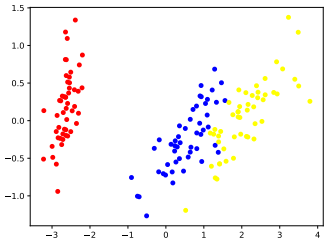

FIGURE 2 – Ground truth label with PCA

Figure 3 shows the result of logistic model on PCA. It looks weird because the decision boundary does not exactly divides the data. It's because sklearn PCA solver use bias (parallel shift) to avoid zero-division problem. I converted point and normal vector of decision hyperplane in 4 dimension to PCA space, but it's hard to convert exact vector without knowing exact value of bias. The accuracy is 94.6%.

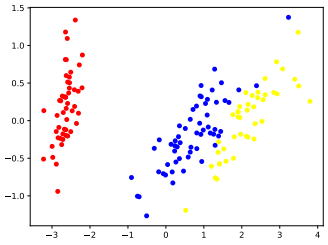

FIGURE 3 – Logistic regression result on PCA

So, I implemented logistic model in 2 dimension (with 2 feature and binary class). Result of logistic model without PCA and its decision bounary are shown in Figure 4.

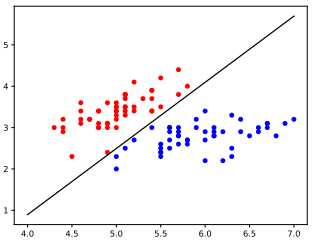

FIGURE 4 – Logistic regression result and decision boundary without PCA

I had an ablation study about eta(learning rate). I experimented the logistic regression mode with eta = 0.0001, 0.0005, 0.001, 0.005, 0.01. Maximum iteration is fixed with 1000. The result are shown in Figure 5. I used eta 0.005 to get all other figures I mentioned above. We can see the model has best performance when eta = 0.0005. When learning rate is too small, the model needs more time to converge and it makes under fitting. When learning rate is too big, weights change so rapidly causes the risk of getting out of bounds(causes overflow and disturbs learning). So, we need to find proper learning as hyper parameter.

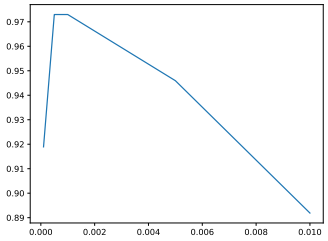

FIGURE 5 – Ablation study about eta(learning rate)

REPORT3

Stochastic gradient ascent algorithm randomly select next training data and time to pack propagate. So, Convergence of the model is not guaranteed and needs more iteration to get result. By the result of REPORT 2, I used maxiter 100000 and eta 0.0005 to run train SGA. Figure 6 shows the result of SGA. The accuracy is 97.3%

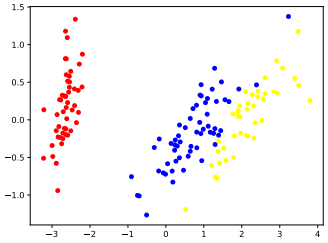

FIGURE 6 – Result of SGA on PCA

Like Gradient ascent algorithm, it's hard to visualize the decision boundary on PCA. So I expressed decision boundary in 2 dimension (with 2 feature and binary class). The results are shown in Figure 7.

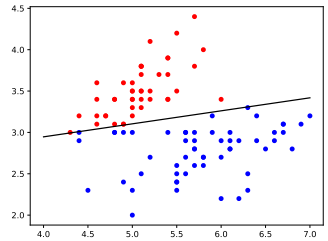

FIGURE 7 – Result of SGA and decision boundary without PCA

I also had an ablation study about maximum iteration. I experimented the Stochastic gradient ascent algorithm with fixed eta 0.0005(found in REPORT 4) and Maximum iteration = 1000, 5000, 10000, 50000, 100000 and 500000. The result are shown in Figure 8. We can see the performance of the model converges when maximum iteration grows. When model's performance is converged, additional training is meaningless. So we need to find proper maximum iteration as the hyper parameter.

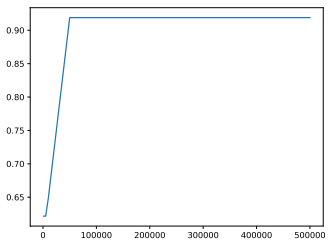

FIGURE 8 – Ablation study about maximum iteration

REPORT4

We can use regularized Stochastic gradient ascent algorithm(MCAP) to avoid over fitting. Figure 9 shows the result of regularized Stochastic gradient ascent algorithm. It's lambda value is 0.005. The accuracy is 89.2%.

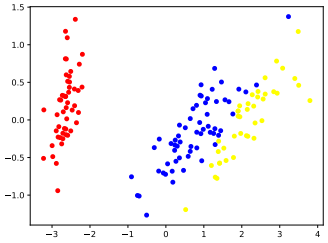

FIGURE 9 – Result of MCAP on PCA

Figure 10 shows the decision boundary of regularized Stochastic gradient algorithm without PCA. We can see slope of the decision boundary is low because of the regularization.

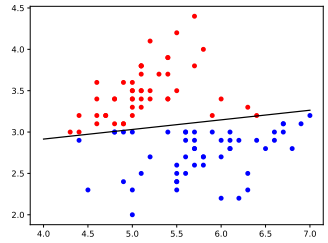

FIGURE 10 – Result of MCAP and decision boundary without PCA

I had an ablation study about lambda value of MCAP. I experimented the regularized Stochastic gradient ascent algorithm with fixed eta 0.0005 and Maximum iteration = 1000. Lambda value varies as 0, 0.0001, 0.0005, 0.001, 0.005 and 0.01. The result are shown in Figure 11. When lambda is 0, It's method is exactly same as Stochastic gradient ascent algorithm. It does not performs any regularization, so result is same. Lambda determines the strength of regularization. Higher lambda forces the model to express simpler and drops the train accuracy. In contrast, Lower lambda weakly force so model can express complex feature and it's good for train accuracy.

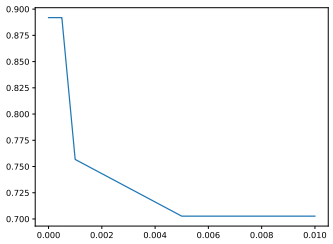

FIGURE 11 – Ablation study about lambda on MCAP

REPORT5

I implemented perceptron to classify IRIS data set. Like logistic regression, I used 4 features and 3 classes. I also performed PCA to visualize the result. I used one-verses-all strategy to classify multi classes. So the threshold value as hyper parameter become meaningless. Threshold value has its meaning without PCA so I had an ablation study about it later. The result of perceptron with PCA(without threshold) is shown in Figure 12.

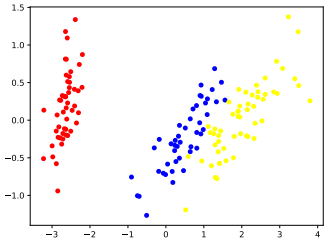

FIGURE 12 – Result of Perceptron with PCA

Figure 13 shows the decision boundary of perceptron without PCA. It's accuracy is 96%.

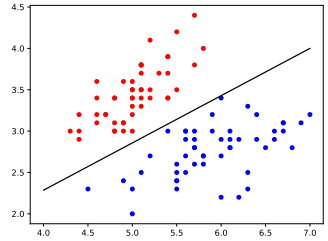

FIGURE 13 – Result of Perceptron and decision boundary without PCA

I also had an ablation study about threshold. I used 2D perceptron(without PCA) to check effect of threshold. The result is shown in Figure 14. Threshold Changed from 0 to 1 in 0.001 increments. Model's accuracy does not depends on the threshold value.

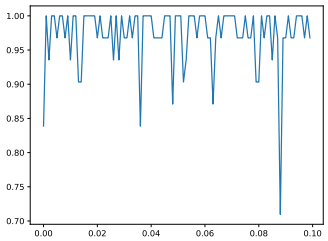

FIGURE 14 – Ablation study about threshold on Preceptron

REPORT6

Sci-kit learn Naive Bayes text classification model predict the test data set as "not spam". It's probability of "not spam" is 53.94% and probability of "spam" is 46.05%. My implementation, log probability of "not spam" is -4260 and log probability of "spam" is -4583 with natural logarithm. I think I used too dense token to predict the model so the expected probability becomes too small. Still, it's prediction is same as sci-kit learn model.

REPORT7

In REPORT 7,8,9, I tested the sci-kit model's performance on feature 4 and 3 classes. Sci-kit learn Logistic regression model's accuracy on validation set is 97.30% without regularization. My implementation's accuracy is 94.6%(Shown in Figure 3). My accuracy is little bit low but with little more iteration, performance of both model are similar.

REPORT8

Sci-kit learn regularized logistic regression model's accuracy on validation set is 78.38% with lambda = 0.005. My implementation's accuracy is 89.2%.(Shown in FIGURE 9). My accuracy is little bit higher than sci-kit learn model. SGA sometimes get poor performance because it's training is stochastic. We can not claim my implementation outperforms sci-kit model, but it has better performance on few trials that we observed.

REPORT9

Sci-kit learn perceptron model's accuracy on validation set is 91.89% with threshold = 0.005. My implementation's accuracy is 96%.(Shown in FIGURE 13). My accuracy is little bit higher than sci-kit learn model. Perceptron is online learning so sometimes get poor performance. We can not claim my implementation outperforms sci-kit model, but it has better performance on few trials that we observed.