Machine learning HW 3

裴嘉政 519111910231

PART 1

(1) Lagrangian function :

$$L(w,b,\xi,\alpha,\beta) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i + \sum_{i=1}^{m}\alpha_i\left(1-\xi_i - y_i(w^Tx_i+b)\right) + \sum_{i=1}^{m}\beta_i(-\xi_i)$$

$$= \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i - \sum_{i=1}^{m}\alpha_i\left(y_i(w^Tx_i+b)+\xi_i-1\right) - \sum_{i=1}^{m}\beta_i\xi_i$$

$$\frac{\partial L}{\partial w} = w - \sum_{i=1}^{m}\alpha_i y_i x_i = 0 \Rightarrow w = \sum_{i=1}^{m}\alpha_i y_i x_i$$

$$\frac{\partial L}{\partial b} = -\sum_{i=1}^{m}\alpha_i y_i = 0 \Rightarrow \sum_{i=1}^{m}\alpha_i y_i = 0$$

$$\frac{\partial L}{\partial \xi_i} = C - \alpha_i - \beta_i = 0 \Rightarrow C = \alpha_i + \beta_i$$

$$L = \frac{1}{2}\left(\sum_{i=1}^{m}\alpha_i y_i x_i\right)^T\left(\sum_{i=1}^{m}\alpha_i y_i x_i\right) + C\sum_{i=1}^{m}\xi_i - \sum_{i=1}^{m}\alpha_i\left(y_i\left(\left(\sum_{j=1}^{m}\alpha_j y_j x_j\right)^T x_i + b\right)+\xi_i-1\right)$$
$$-\sum_{i=1}^{m}\beta_i\xi_i$$

$$= \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j x_i^T x_j + C\sum_{i=1}^{m}\xi_i - \sum_{i=1}^{m}\alpha_i\left(y_i\left(\sum_{j=1}^{m}\alpha_j y_j x_j^T x_i + b\right)+\xi_i-1\right) - \sum_{i=1}^{m}\beta_i\xi_i$$

$$= \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j x_i^T x_j + (C-\alpha_i-\beta_i)\sum_{i=1}^{m}\xi_i - \sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j x_i^T x_j - b\sum_{i=1}^{m}\alpha_i y_i$$
$$+\sum_{i=1}^{m}\alpha_i$$

$$= -\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j x_i^T x_j + \sum_{i=1}^{m}\alpha_i$$

Primal : $\max\limits_{\alpha} \; -\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j x_i^T x_j + \sum_{i=1}^{m}\alpha_i$

s.t. $\sum_{i=1}^{m}\alpha_i y_i = 0$

$C - \alpha_i - \beta_i = 0$

$\alpha_i \geqslant 0,\ \beta_i \geqslant 0,\ i = 1, 2, \cdots, m$

Since $\alpha_i + \beta_i = C$ and $\beta_i \geqslant 0$, we get $\alpha_i \leqslant C$

Dual $\min\limits_{\alpha} \; \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j x_i^T x_j - \sum_{i=1}^{m}\alpha_i$

s.t. $\sum_{i=1}^{m}\alpha_i y_i = 0$

$0 \leqslant \alpha_i \leqslant C$

(2) KKT:

$$\frac{\partial L}{\partial w} = w^* - \sum_{i=1}^{m} \alpha_i^* y_i x_i = 0 \quad (1)$$

$$\alpha_i^* \left(y_i (w^{*T} x_i + b^*) - 1 + \xi_i^* \right) = 0 \quad (2)$$

$$\beta_i^* \xi_i^* = 0 \quad (3)$$

$$(1) \Rightarrow w^* = \sum_{i=1}^{m} \alpha_i^* y_i x_i$$

Since $C - \alpha_i^* - \beta_i^* = 0$, if there exists α_j^*, where $0 < \alpha_j^* < C$, then

$\beta_j^* \neq 0$. According to (2)、(3), we can derive $y_j (w^{*T} x_j + b^*) - 1 = 0$.

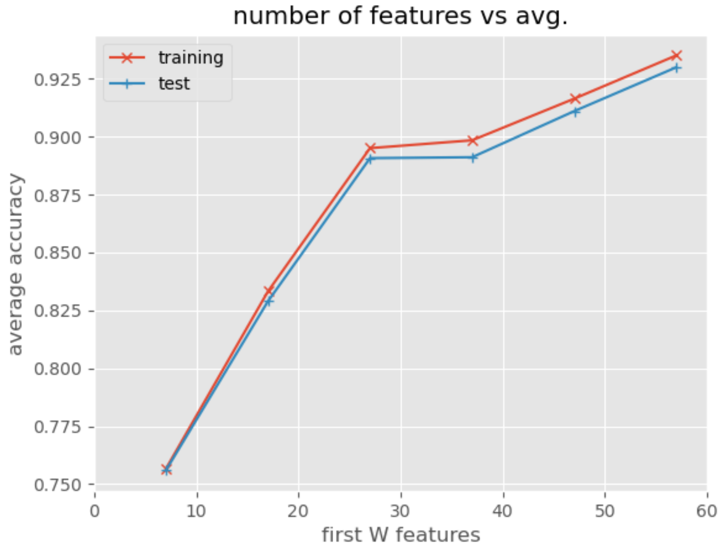
$$\Rightarrow y_j \left(\sum_{i=1}^{m} \alpha_i^* y_i x_i^T x_j + b^* \right) - 1 = 0$$

$$b^* = \frac{1}{y_j} - \sum_{i=1}^{m} \alpha_i^* y_i x_i^T x_j$$

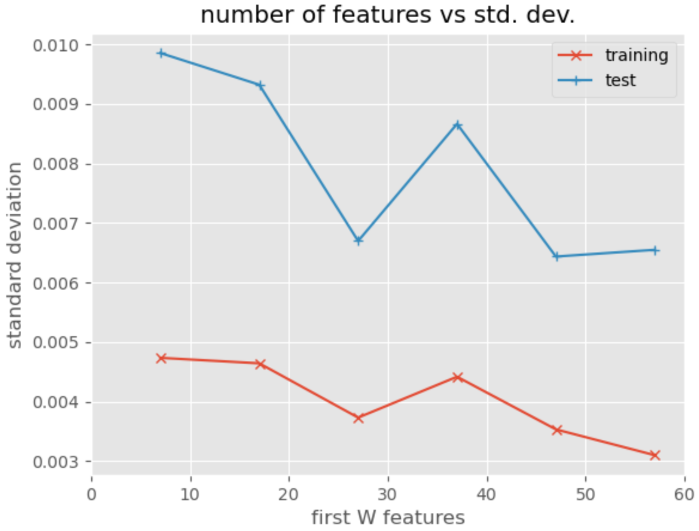
PART2

Problem1:

The average accuracy on training data with different number of features:
[0.7565093167701865, 0.8331614906832298, 0.8950527950310558, 0.898444099378882, 0.9164130434782609, 0.9351086956521738]
The standard deviation of accuracy on training data with different number of features:
[0.004734149919066124, 0.004642120675297999, 0.003730653966916669, 0.0044168205847390715, 0.0035325660361904486, 0.003099178163505262]
The average accuracy on test data with different number of features:
[0.7558508327299057, 0.8288993482983347, 0.8907385952208544, 0.89112961622013, 0.9110354815351194, 0.929971035481535]
The standard deviation of accuracy on test data with different number of features:
[0.009852756306479487, 0.009320712173161614, 0.006696298968098945, 0.008662991019655338, 0.006435258336550281, 0.0065485559135709925]



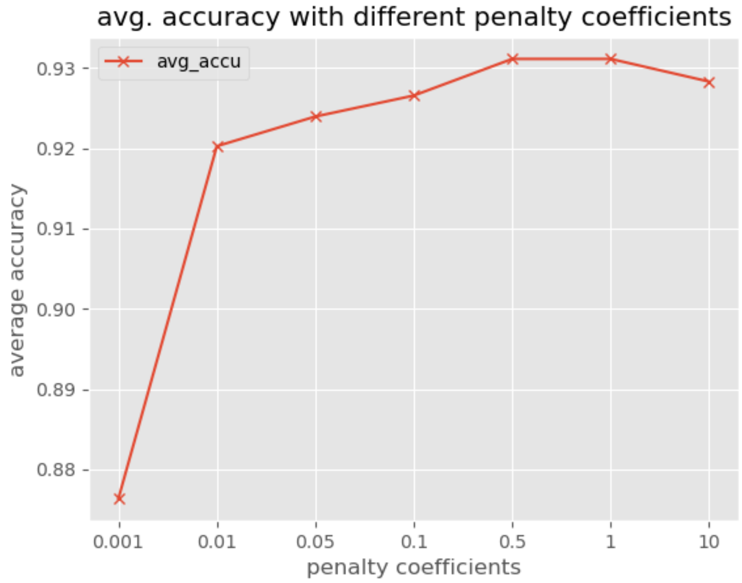
I select the first W features of the dataset, and record the accuracy of both training set and test set. The average accuracy of both datasets increase with W. With an increasing number of features, more critical information is provided for the classifier, which improves the accuracy.

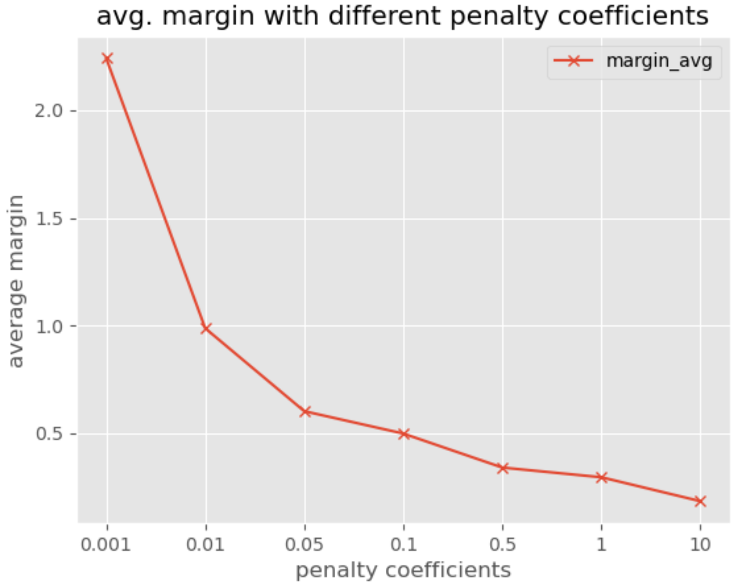


As for the standard deviation of the accuracy with first W features, there's no explicit interpretation for these two curves. The value of standard deviation depends mainly on the data distribution, not the algorithm and the select of features.

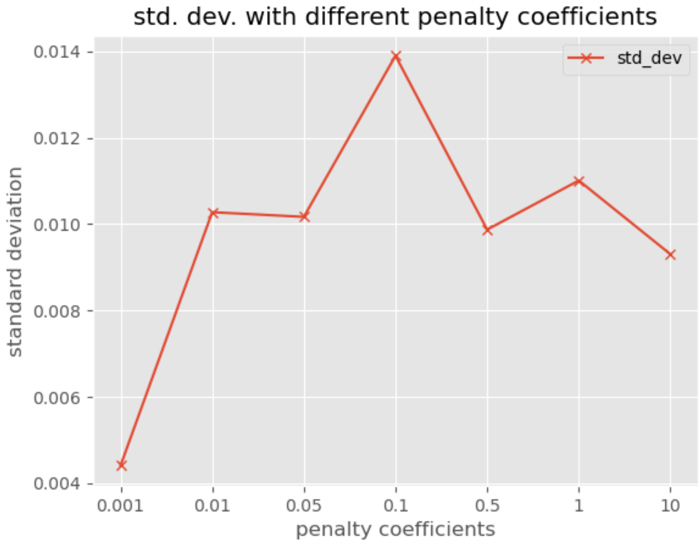
Problem 2:

```
The average accuracy with different penalty coefficients:
[0.8763307554465717, 0.9202362538904083, 0.9239262472885033,
0.9265396585871924, 0.9311011034612845, 0.9311001603319815,
0.9282787890219748]
The std. dev. of accuracy with different penalty coefficients:
[0.004408915164092 58, 0.010278129084166 32, 0.010169955694542 85,
0.013901382806319057, 0.009874569443906116, 0.011011095491976545,
0.00930429246604665]
The average margin with different penalty coefficients:
[2.241297377244207, 0.9878516196128857, 0.6030412745101472,
0.4992609934894783, 0.3414930943365735, 0.2964295703686542,
0.18533053455422302]
```





The average accuracy first increases with penalty coefficients, and reaches its peak when C=0.5. After that, it goes down a little bit. The average margin continues to go down with the increase of penalty coefficients. A lower penalty coefficient allows a large number of mistakes because the penalties for misclassification are small. Therefore, the algorithm will try its best to maximize the margin according to the given optimization problem at the cost of accuracy. By contract, a large coefficient means few cases of misclassification are allowed. The algorithm assures the best performance on training data at the cost of margin, but that does not necessarily assure a high accuracy on test data because of over-fitting.



Again, the curve of standard deviation does not have an exact meaning. The value is small when C=0.001, because the algorithm is underfitting and the results are all very poor.

Therefore, I will choose C=0.5 to avoid both underfitting and overfitting because the corresponding average accuracy is the highest.

```
The accuracy on test data with the chosen penalty coefficient is 0.9152787834902245
```

Problem 3:

```
The average and the standard deviation of the accuracy of SVM:
0.9302353107611052 0.010824546927472726
The average and the standard deviation of the accuracy of Logisitc Regression:
0.9302348391964539 0.013117890036932985
The average precision and recall of SVM:
0.921736362337497 0.8988045869367658
The average precision and recall of Logistic Regression:
0.9256059371337336 0.8945995937432961
t-statistic value=0.7145922968351659
```

To show whether one classifier is better than another, I'll use paired t-test to illustrate that.

H_0: SVM_accu=LR_accu

$t \sim t_9$, $\alpha=0.05$

$t_{0.025,9}=2.26>0.71$

Therefore, the null hypothesis should be rejected and the one with higher average accuracy is better. (It's SVM in this case)

I run this code for several times, and find that although the average accuracy remains almost the same, the standard deviation varies dramatically. For most of the cases, the calculated t value is smaller than 2.26, but the opposite cases may also happen. However, the opposite case is a small probability event, so we can still come to the conclusion that SVM is better than Logistic Regression in this case.