SVM Report

Hangyu Wang
518030910283

I. Methods

A. SVM

SVM's objective is to maximize

$$\begin{aligned}\max_{\alpha} \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j x_i^T x_j \\ s.t. \sum_{i=1}^{m}\alpha_i y_i = 0,\ \alpha_i \geq 0, i = 1,2,..,m\end{aligned} \tag{1}$$

SMO algorithm is always used to solve this problem, which I will introduce later. Here, we suppose that we have get the solution α^*. Then we can compute $w^* = \sum_{i=1}^{m}\alpha_i^* y_i x_i$. The decision function is

$$G(x_i) = sign[x_i^T w^* + b] \tag{2}$$

However, sometimes the data set can't be linear classification as it contains some outliers. SVM use soft margin to overcome this type of problem. Induce $\xi_i \geq 0$ for any (x_i, y_i), s.t. $y_i(w^T x_i + b) \geq 1 - \xi_i$, then the new formulation is

$$\begin{aligned}\min \frac{1}{2}||w||_2^2 + C\sum_{i=1}^{m}\xi_i \\ s.t.\ y_i(w^T x_i + b) \geq 1 - \xi_i(i = 1,2,..,m),\ \xi_i \geq 0(i = 1,..m)\end{aligned} \tag{3}$$

Where $C > 0$ is a regularization parameter.Same with former formulation, We can get the new objective function is

$$\begin{aligned}\min_{\alpha} \frac{1}{2}\sum_{i=1,j=1}^{m}\alpha_i\alpha_j y_i y_j x_i^T x_j - \sum_{i=1}^{m}\alpha_i \\ s.t. \sum_{i=1}^{m}\alpha_i y_i = 0,\ 0 \leq \alpha_i \leq C\end{aligned} \tag{4}$$

We induce kernel function to overcome the problem of linearly inseparable low-dimensional feature data. Suppose ϕ is a mapping from low dimension space χ to high dimension space $\mathcal{H}$, if exists $K(x,z)$, for any $x, z \in \chi$, $K(x,z) =< \phi(x), \phi(z) >$, then $K(x,z)$ is a kernel function. In order to introduce kernel function, we just need to replace x_i with $\phi(x_i)$ and the use of $K(x_i, x_j)$ can remove the need to compute $\phi(x_i)$ specifically. Next part, I will introduce SMO algorithm.

B. SMO algorithm

1) Basic idea: First, let's review SVM's objective function

$$\begin{aligned}\min_{\alpha} \frac{1}{2}\sum_{i=1,j=1}^{m}\alpha_i\alpha_j y_i y_j K(x_i, x_j) - \sum_{i=1}^{m}\alpha_i \\ s.t. \sum_{i=1}^{m}\alpha_i y_i = 0,\ 0 \leq \alpha_i \leq C\end{aligned} \tag{5}$$

And our solution has to satisfy KKT condition

$$\alpha^*(y_i(w^T x_i + b) - 1 + \xi_i^*) = 0 \tag{6}$$

Define $g(x) =< w^*, \phi(x) > +b = \sum_{j=1}^{m}\alpha_j^* y_j K(x, x_j) + b^*$, we have

$$\begin{aligned}\alpha^* = 0 \rightarrow y_i g(x_i) \geq 1 \\ 0 < \alpha^* < C \rightarrow y_i g(x_i) = 1 \\ \alpha^* = C \rightarrow y_i g(x_i) \leq 1\end{aligned} \tag{7}$$

SMO only optimizes two variables at a time, and treats the other variables as constants. If we optimize α_1 and α_2, the new objective function is

$$\begin{aligned}\min_{\alpha_1,\alpha_2} \frac{1}{2}K_{11}\alpha_1^2 + \frac{1}{2}K_{22}\alpha_2^2 + y_1 y_2 K_{12}\alpha_1\alpha_2 - (\alpha_1 + \alpha_2) + \\ y_1\alpha_1\sum_{i=3}^{m}y_i\alpha_i K_{i1} + y_2\alpha_2\sum_{i=3}^{m}y_i\alpha_i K_{i2} \\ s.t.\ \alpha_1 y_1 + \alpha_2 y_2 = -\sum_{i=3}^{m}y_i\alpha_i = \zeta,\ 0 \leq \alpha_i \leq C(i = 1,2)\end{aligned} \tag{8}$$

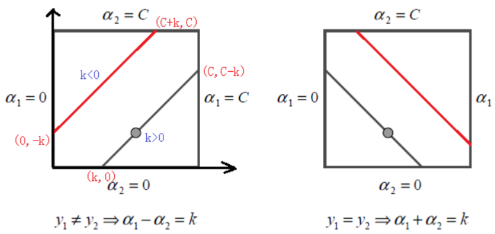

Fig. 1.

Look the Fig.1. Suppose we get the solution $\alpha_1^{old}, \alpha_2^{old}$ in the last iteration, and get $\alpha_1^{new}, \alpha_2^{new}$ in this iteration. Then we have

$$\alpha_2^{new} = \begin{cases} H, & \alpha_2^{new,unc} > H \\ \alpha_2^{new,unc}, & L \leq \alpha_2^{new,unc} \leq H \\ L, & \alpha_2^{new,unc} < L \end{cases} \tag{9}$$

If it's the condition in the left plot of Fig.1:

$$L = \max(0, \alpha_2^{old} - \alpha_1^{old}) \ H = \min(C, C + \alpha_2^{old} - \alpha_1^{old}) \quad (10)$$

If it's the right condition:

$$L = \max(0, \alpha_2^{old} + \alpha_1^{old} - C) \ H = \min(C, \alpha_2^{old} - \alpha_1^{old}) \quad (11)$$

Moreover, $\alpha_2^{new,unc}$ can be computed by taking the partial derivative of the objective function to α_2. Here I omit the detailed process of derivation. We define

$$E_i = g(x_i, y_i) == \sum_{j=1}^{m} \alpha_j^* y_j K(x_i, x_j) + b - y_i \quad (12)$$

Then $\alpha_2^{new,unc}$ is

$$\alpha_2^{new,unc} = \alpha_2^{old} + \frac{y_2(E_1 - E_2)}{K_{11} + K_{22} - x - 2K_{12}} \quad (13)$$

2) Choose two variables: For the first variable, we choose the points that violate the KKT condition(Eq.7) most severely. Generally, we first choose the point that violates $0 < \alpha^* < C \to y_i g(x_i) = 1$.

For the second variable, we choose the variable that maximizes the change of $|E_1 - E_2|$. We can record all E_i to accelerate iterations.

3) Compute b and E_i: After the optimization of two variables, we need to recalculate b. If $0 < \alpha_1^{new} < C$, we have

$$b_1^{new} = -E_1 - y_1 K_{11}(\alpha_1^{new} - \alpha_1^{old}) - y_2 K_{21}(\alpha_2^{new} - \alpha_2^{old}) + b^{old} \quad (14)$$

Samely, if $0 < \alpha_2^{new} < C$, we have

$$b_2^{new} = -E_2 - y_1 K_{12}(\alpha_1^{new} - \alpha_1^{old}) - y_2 K_{22}(\alpha_2^{new} - \alpha_2^{old}) + b^{old} \quad (15)$$

And $b^{new} = \frac{b_1^{new} + b_2^{new}}{2}$. Then update E_i:

$$E_i = \sum_S y_j \alpha_j K(x_i, x_j) + b^{new} - y_i \quad (16)$$

where S is the set of all support vectors. SMO algorithm will continue the iteration until all α_i satisfy the KKT condition.

II. Experimental Settings

My training set is gpl96.csv and testing set is gpl97.csv. The environment is Python 3.7 with numpy, pandas, matplotlib and sklearn library.

III. Result& Discussion

I implement SMO algorithm with two kernel functions. In detail, I use linear kernel and RBF kernel:

$$\begin{aligned} Linear &: K(x, y) =< x, y > \\ RBF &: K(x, y) = exp(-\frac{||x - y||^2}{2\sigma^2}) \end{aligned} \quad (17)$$

The following is the specific meaning of the different parameters: Moreover, besides the accuracy, I also use AUC curve

TABLE I
The meaning of the different parameters

Names in code	Meaning
C	the regularization parameter
toler	fault tolerance
maxIter	the maximum number of iterations

to evaluate the results by my SVM and the sklearn SVM.

$$\begin{aligned} Sensitivity = TPR &= \frac{TP}{TP + FN} \\ Specificity = 1 - FPR &= \frac{TN}{FP + TN} \\ Recall = FPR &= \frac{FP}{FP + TN} \\ Precision &= \frac{TP}{TP + FP} \\ F1 &= \frac{2Precision * Recall}{Precision + Recall} \end{aligned} \quad (18)$$

A. Linear Kernel Result

In this part, I fix the value of $C = 0.8$. In my SVM, I set $toler = 0.001$, $maxIter = 15$. The AUC curve is in Fig.2:

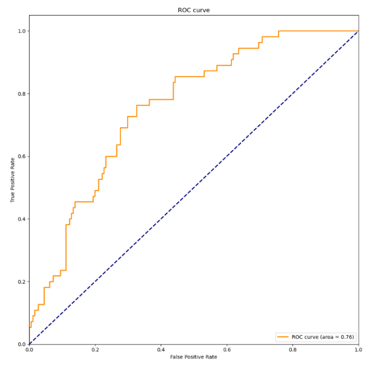

Fig. 2. My SVM ROC curve using linear kernel.

In sklearn SVM, I set $C = 0.8$. The AUC curve is in Fig.3:

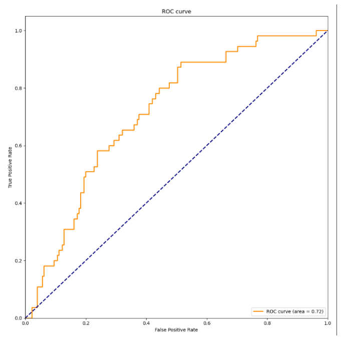

Fig. 3. Sklearn SVM ROC curve using linear kernel.

Table.II contains the different evaluation results of My SVM andd Sklearn SVM with linear kernel(F1 score and AUC are both tested on the testing dataset). We can see that using

TABLE II
THE EVALUATION OF MY SVM AND SKLEARN SVM WITH LINEAR KERNEL

	My SVM	Sklearn SVM
training acc	0.8347	0.9322
testing acc	0.7331	0.7542
F1 score	0.2703	0.0
AUC	0.76	0.72

the linear kernel, My SVM's training accuracy and testing accuracy are a little bit lower than Sklearn SVM's, My SVM's AUC surpasses Sklearn SVM's. But My SVM's F1 score is just 0.27 and Sklearn's F1 score is 0. These results show that there is an imbalance between positive and negative samples in the data, and our SVM may only learn the information of negative samples or positive samples, which will cause the model accuracy rate to be high, but the F1 score are relatively low.

B. RBF Kernel Result

In this part, I also fix $C = 0.8$. In MySVM, I set $toler = 0.001$, $maxIter = 15$ and $\sigma = 0.5$(the paramater for RBF). The AUC curve is in Fig.4.

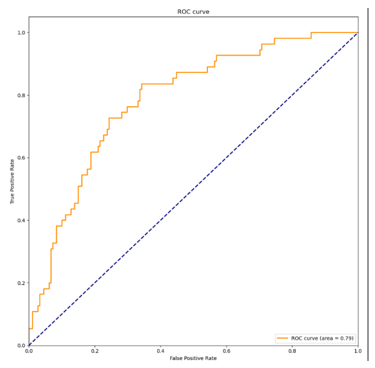

Fig. 4. My SVM ROC curve using RBF kernel

In Sklearn SVM, I set $C = 0.8$ and $gamma = 0.2$(the parameter for Sklearn SVM). The AUC curve is in Fig.5

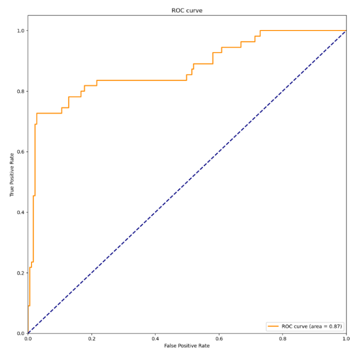

Fig. 5. Sklearn SVM ROC curve using RBF kernel.

Table.III contains the different evaluation results of My SVM andd Sklearn SVM with RBF kernel(F1 score and AUC are both tested on the testing dataset). We can see that using the RBF kernel, both SVMs' performances are improved. But there still exists the same condition as the linear kernel: F1 score is very small. After I check the testing dataset, I find the main reason: there are too many negative samples and too few positive samples in the training set and training set. SVM judges all samples as negative samples, so that although the accuracy and AUC are high, the precision is 0, resulting in a F1 score of zero.

TABLE III
THE EVALUATION OF MY SVM AND SKLEARN SVM WITH RBF KERNEL

	My SVM	Sklearn SVM
training acc	0.7670	0.9958
testing acc	0.7754	0.7669
F1 score	0.2535	0.0
AUC	0.79	0.87

C. Cross validation

In this part, I will use 5-fold cross-validation to assess logistic regression and SVM on the gpl96.csv dataset using metrics like accuracy, AUC and F1 score. Here, I use SVM and Logistic Regression both from sklearn library. In SVM, I use linear kernel and set $C = 0.8$. The results are as follows:

TABLE IV
THE AVERAGE EVALUATION RESULTS OF LOGISTIC REGRESSION AND SVM USING CROSS-VALIDATION

	Logistic Regression	SVM
accuracy	0.8520	0.8354
F1 score	0.6953	0.6941
AUC	0.9221	0.9186

Table.IV shows the average evaluation results about logistic regression and SVM using cross-validation(all tested on the gpl96.csv). We can see that the performance of SVM is similar to logistic, and logistic is slightly better. This may be because the dataset is linearly separable, so the performances of these two methods are similar.