SVM algorithm report

Group members: Hu Zhenwei (3035533719) & Chang Liyan (3035534880)

1. Introduction:

SVM is an algorithm designed to classify samples into positive and negative class. The idea is to maximize the margin between positive and negative classes. However, it sometimes fails to classify linearly inseparable test samples. Therefore, we introduce a slack variable to allow misclassification. Then this problem can be converted into a quadratic programming problem which can be optimized by CVXOPT package. Advantages, limitations and further implementation are to be discussed in later sections.

2. Analysis of Algorithm:

The optimization problem with slack variables in its primal form is given by

$$\begin{aligned} \text{Min} \quad & \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\sum_{i=1}^{m}\xi_i \\ \text{s.t.} \quad & y_i(\boldsymbol{w}^T\boldsymbol{x_i} - b) - 1 + \xi_i \geq 0 \quad \forall i \\ & \xi_i = \max\{0, 1 - y_i(\boldsymbol{w}^T\boldsymbol{x_i} - b)\} \geq 0 \quad \forall i \end{aligned}$$

Then, by definition, we can write its dual function

$$\begin{aligned} \phi(\boldsymbol{\alpha}, \boldsymbol{\mu}) &= \text{Min}\,\{\frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} - \sum_{i=1}^{m}\alpha_i[y_i(\boldsymbol{w}^T\boldsymbol{x_i} - b) - 1] - \sum_{i=1}^{m}\mu_i\xi_i\} \\ &\triangleq \text{Min}\, L(\boldsymbol{w}, b, \boldsymbol{\xi}; \boldsymbol{\alpha}, \boldsymbol{\mu}), \end{aligned}$$

where $\boldsymbol{\alpha}, \boldsymbol{\mu} \geq \mathbf{0}$. Now, setting $\frac{\partial L}{\partial \boldsymbol{w}} = \mathbf{0}$, $\frac{\partial L}{\partial b} = 0$ and $\frac{\partial L}{\partial \xi_i} = 0$ gives

$$\begin{aligned} &\frac{\partial L}{\partial \boldsymbol{w}} = \boldsymbol{w} - \sum_{i=1}^{m}\alpha_i y_i \boldsymbol{x_i} = 0 \Rightarrow \boldsymbol{w}_* = \sum_{i=1}^{m}\alpha_i y_i \boldsymbol{x_i} \\ &\frac{\partial L}{\partial b} = \sum_{i=1}^{m}\alpha_i y_i = 0 \\ &\frac{\partial L}{\partial \xi_i} = C - \alpha_i - \mu_i = 0 \Rightarrow \alpha_i = C - \mu_i \leq C \end{aligned}$$

Since $L(\boldsymbol{w}, b, \boldsymbol{\xi}; \boldsymbol{\alpha}, \boldsymbol{\mu})$ is a convex function for any fixed $\boldsymbol{\alpha}$ and $\boldsymbol{\mu}$, we have

$$\phi(\boldsymbol{\alpha}, \boldsymbol{\mu}) = \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{i=1}^{m}\alpha_i\alpha_j y_i y_j \boldsymbol{x_i}^{\mathrm{T}}\boldsymbol{x_j} \triangleq J(\boldsymbol{\alpha})$$

Hence, we get the dual form of original problem

$$\begin{aligned} \text{Max} \quad & J(\boldsymbol{\alpha}) \\ \text{s.t.} \quad & \sum_{i=1}^{m}\alpha_i y_i = 0 \quad \forall i \\ & \alpha_i \leq C \quad \forall i \end{aligned}$$

Note that if $J(\boldsymbol{\alpha})$ is multiplied by -1, this a traditional Quadratic Programming problem.

$$\begin{aligned} \text{Min} \quad & \tfrac{1}{2}\boldsymbol{x}^T P\boldsymbol{x} + \boldsymbol{q}^T\boldsymbol{x} \\ \text{s.t.} \quad & G\boldsymbol{x} \le \boldsymbol{h} \\ & A\boldsymbol{x} = \boldsymbol{b} \end{aligned}$$

Hence, we can construct this problem into

$$P = \boldsymbol{y}\boldsymbol{y}^{\boldsymbol{T}}\boldsymbol{x}_{\boldsymbol{i}}^T\boldsymbol{x}_{\boldsymbol{j}},\ \boldsymbol{q} = -\mathbf{1}$$

$$0 \le \alpha_i \le C, \forall i \ \equiv \ G = [-I, I]^T,\ \boldsymbol{h} = [0,0\ldots,0, C, C, \ldots, C]^T$$

$$\sum_{i=1}^{m} \alpha_i y_i = 0 \ \ \forall i \ \equiv \ A = trace\{y_1, y_2, \ldots, y_m\},\ \boldsymbol{x} = [\alpha_1, \alpha_2, \ldots, \alpha_m]^T,\ \boldsymbol{b} = \mathbf{0}.$$

Now, this problem can be solved using CVXOPT package and we get

$$\boldsymbol{w}_* = \sum_{i=1}^{m} \alpha_i y_i \boldsymbol{x}_{\boldsymbol{i}}$$

$$b = average\{\frac{1}{y_i} - \boldsymbol{w}_*^T\boldsymbol{x}_{\boldsymbol{i}}\}_{i\in m}$$

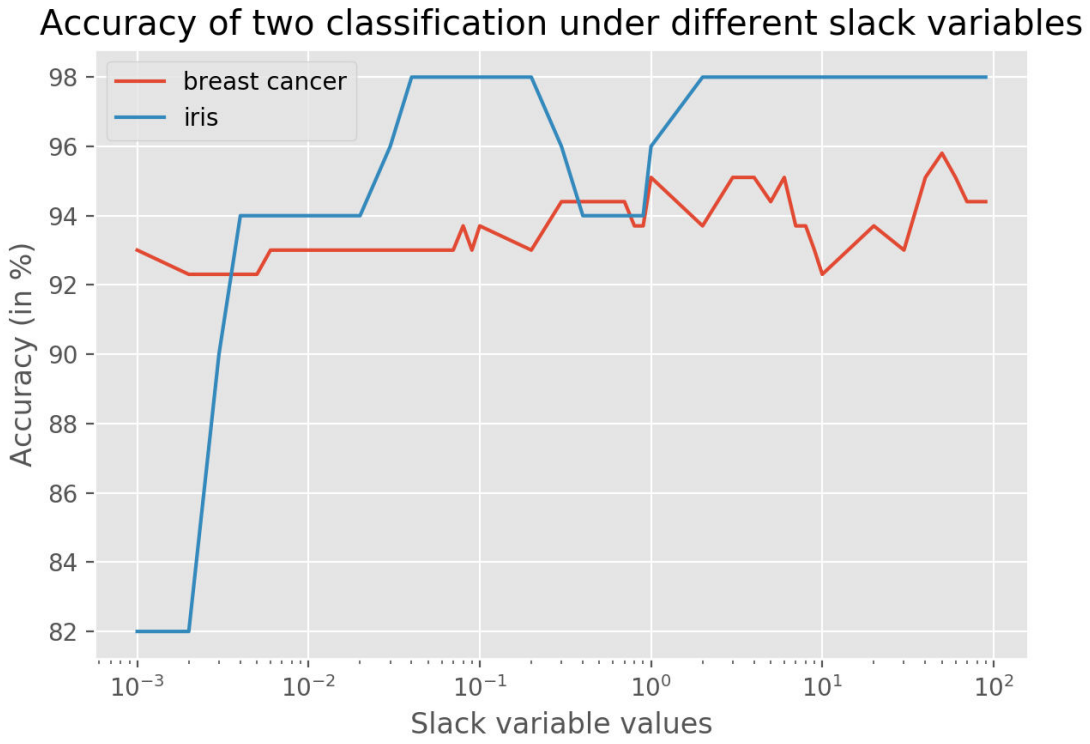
The first equation is obtained directly from the primal problem and the second is solved from the decision boundary equation with a support vector $\boldsymbol{x}_{\boldsymbol{i}}$. Finally, we can predict a class of sample $\boldsymbol{x}$ by $sgn(\boldsymbol{w}^T\boldsymbol{x} + b)$.

3. OVA (One-versus-all classifier):

We can classify binary class using SVM. For multiple classes, we need to apply OVA. For example, we have three classes in breast cancer dataset. The idea is to create three SVM classifiers, each of which considers class 0,1,2 as positive class respectively and the rest as negative class. For each test sample we predict it using the aforementioned three classifiers and take the one with the highest score.

4. Effect of C (slack variable):

When C is bigger, it is prone to a hard-margin SVM. We draw below a graph showing the accuracy of two classifications under different value of slack variables. It shows a trend of increasing accuracy when C is increasing. This denotes the characteristic of two datasets, where we might infer that Iris dataset is more "linearly separable" than breast cancer dataset, as its accuracy reaches 98% when C is around 0.1 or 10, which is higher than that of breast cancer dataset.



5.Advantages

- The kernel trick is real strength of SVM. With an appropriate kernel function, we can solve any complex problem. Further discussion about kernel trick is written in section 7.
- The regularization term(slack variables) is implicitly able to capture important dimensions from the high dimensional data. Also, it allows more flexibility and works for linearly inseparable datasets.
- It scales relatively well to high dimensional data. The increase of the dimension is not such a burden for SVM algorithm as SVM does not use all data point to generate decision boundary, compared to other algorithms. It only uses a few of these vectors that support (help) it to separate the points, i.e. the closest data points to the hyperplane.

6. Limitations

- When C is large, larger slacks penalize the objective function of SVM's more than when C is small. Hard-margin SVM is extremely sensitive to outliers. It is more likely to overfit. Also, for training data that is not linearly separable, it will probably fail to find a margin and hence no decision boundary. Conversely, we essentially have the opposite issue if the value of C is quite small. Our slack variables for all data points are free to be as large as possible to maximize the margin and it's easy for our model to underfit the training data. In short, it is hard to adjust the value of slack variable to an appropriate level.
- Besides, SVM faces with a problem that it is not suitable for large data sets. Although the size of breast cancer dataset is not that big, it can still be reflected that SVM perform less effectively for it than for iris dataset. If the dataset exceeds ten thousand or billion rows, the performance of SVM is going to drastically fall.

7. Summary and further implementations:
In conclusion, linear SVM shows a good performance on simple datasets. For more complicated datasets, we might consider using kernel trick. Then the formula will become

$$J(\boldsymbol{\alpha}) = \frac{1}{2}\sum_{i=1}^{m}\sum_{i=1}^{m}\alpha_i\alpha_j y_i y_j K(\boldsymbol{x_i}, \boldsymbol{x_j}) - \sum_{i=1}^{m}\alpha_i$$

Some other kernel tricks are polynomial, radial basis function (RBF), sigmoid function and so on. The effects of these kernel functions are not discussed as they perform much worse than linear kernel function with the given datasets(halve the accuracy), and it is not within the scope of this assignment.

8. Reference:
https://sandipanweb.wordpress.com/2018/04/23/implementing-a-soft-margin-kernelized-support-vector-machine-binary-classifier-with-quadratic-programming-in-r-and-python/

https://courses.csail.mit.edu/6.867/wiki/images/a/a7/Qp-cvxopt.pdf