Assignment 1:Linear Supported Vector Machines

TAK YIN PANG
University of Adelaide
a1796036@adelaide.edu.au

Abstract

The Supported Vector Machines Algorithm is one of the binary linear classifiers for supervised learning. It constructs a hyperplane to seperate two classes such that the margin between the hyperplane and the datapoint is maximized. In this assignment, we will go into the details of primal form and dual form of SVMs as well as other important properties.

1. Introduction

The linear SVMs algorithm is a linear classification algorithm by predicting using the linear predictor function. SVMs uses a linear function $w^t x_i + b$ to create the hyperplane and classify y as below.

$$y = 1 \text{ if } \vec{w} \cdot \vec{x} + b >= 1$$
$$y = -1 \text{ if } \vec{w} \cdot \vec{x} + b <= -1.$$

It is very similar to perceptron algorithm but perceptron classifier which uses 0 in the inequality instead of 1 and -1 does not care about the margin between hyperplane and the datapoint. However, SVMs does care the margin and wants to maximize it by setting a minimum margin distance.

2. Linear SVMs

In this section, I will go into details of linear SVMs. Here I will cover the primal and dual forms of hard and soft margin SVMs. Before that I will define some variables and notation in the below.

For a dataset with n number of datapoint, $\vec{x_i}$ is the i th datapoint with d number of features(or dimensions) and y_i is the i th label, where $y_i = 1$ denotes positive label and $y_i = -1$ denotes negative label

$(\vec{x_i}, y_i)$ where $i = 1, 2, 3, \ldots, n$, $y_i \in [-1, 1], x \in \Re^d$

For the hyperplane H that seperates $x_1, x_2, \ldots, x_n$,
$H : \vec{w} \cdot \vec{x} + b = 0$, where
w , the weight vector that is normal to the hyperplane
b , the bias term

3. Hard Margin

If the dataset is linearly seperable, hard margin SVMs would be able to seperate the two classes completely with a margin between the hyperplane and the datapoints of two classes. As mentioned in the introduction, the SVMs predicts y as below.

$$y_i = 1 \text{ if } \vec{w} \cdot \vec{x_i} + b >= 1$$
$$y_i = -1 \text{ if } \vec{w} \cdot \vec{x_i} + b <= -1.$$

These two inequalities can be combined into:

$$y_i * (\vec{w} \cdot \vec{x_i} + b) >= 1$$

Let H_1 and H_2 be $H_1 : \vec{w} \cdot \vec{x_i} + b = 1$ and $H_2 : \vec{w} \cdot \vec{x_i} + b = -1$ respectively. Here, SVMs wants to maximize the distance between H and $H1$ as well as H and $H2$, which is $\frac{1}{\|\vec{w}\|}$. Therefore, the total margin of both side is $\frac{1}{\|\vec{w}\|} + \frac{1}{\|\vec{w}\|} = \frac{2}{\|\vec{w}\|}$,which is what SVMs want to maximize.

3.1. Primal form

Hence, The Primal form of SVMs is as below.

$$max \frac{2}{\|\vec{w}\|}$$
$$\Rightarrow min \frac{\|\vec{w}\|}{2}$$
$$\Rightarrow min \frac{1}{2} * \|\vec{w}\|^2$$
$$\text{such that } y_i * (\vec{w} \cdot \vec{x_i} + b) - 1 >= 0$$

3.2. Dual form

By solving the above with Lagrangian dual, we allocate the lagrange multiplier $\alpha, \forall \alpha_i >= 0$ to the constrict function.

$$L_h(x) = F(x) - \alpha G(x)$$

where L is Lagrangian function and G is constrict function Here,

$$L_h = \frac{1}{2} * \|\vec{w}\|^2 - \alpha(y_i * (\vec{w} \cdot \vec{x_i} + b) - 1) \, \forall i$$
$$= \frac{1}{2} * \|\vec{w}\|^2 - \sum_{i=1}^{n} \alpha_i (y_i * (\vec{w} \cdot \vec{x_i} + b) - 1)$$

By solving the derivative of L with respect to $\vec{w}$ and b equal to zero,

$$\frac{\partial L_h}{\partial w} = 0 \Rightarrow w - \sum_{i=1}^{n} \alpha_i (y_i * \vec{x_i}) = 0 \dots (1)$$
$$\frac{\partial L_h}{\partial b} = 0 \Rightarrow \sum_{i=1}^{n} \alpha_i y_i = 0 \dots (2)$$

By substitute (1) and (2) into L, and instead of minimizing the objective in the primal form, we want to maximize the dual form.

$$max\, L_h \Rightarrow$$
$$max \sum_{i=1}^{n} \alpha_i - \frac{1}{2} * \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j * (\vec{x_i} \cdot \vec{x_j})$$
$$\text{such that } \alpha_i >= 0, \forall i = 1, 2, \dots, n \text{ and } \sum_{i=1}^{n} \alpha_i y_i = 0$$

This is the dual form of hard margin SVMs.

4. Soft Margin

If the dataset is not linearly seperable, soft margin SVMs should be used to allow the incorrect classes lie on both side of the hyperplane. The slack variable ξ_i is introduced here, $\xi_i >= 0\ \forall i = 1, 2, \dots, n$.
Hence,

$$y_i = 1 \text{ if } \vec{w} \cdot \vec{x_i} + b >= 1 - \xi_i$$
$$y_i = -1 \text{ if } \vec{w} \cdot \vec{x_i} + b <= -1 + \xi_i.$$

These two inequalities can be combined into:

$$y_i * (\vec{w} \cdot \vec{x_i} + b) >= 1 - \xi \text{ where } \xi_i >= 0\ \forall_i.$$

This slack variable tolerates the data point lies with in the margin and even in the incorrect side of the hyperplane.

4.1. Primal form

Below, Hinge loss is also introduced because we are not only want to maximize the margin but also minimize the number of misclassification.

$$min \frac{1}{2} * \|\vec{w}\|^2 + \frac{C}{n} \sum_{i=1}^{n} max(0, y_i * (\vec{w} \cdot \vec{x_i} + b) - 1)$$
$$\Rightarrow min \frac{1}{2} * \|\vec{w}\|^2 + \frac{C}{n} \sum_{i=1}^{n} \xi_i$$
$$\text{such that } y_i * (\vec{w} \cdot \vec{x_i} + b) >= 1 - \xi \text{ and } \xi_i >= 0\ \forall_i.$$

where C is the parameter that control the trade-off between maximizing the margin and the slack variable

4.2. Dual form

As well as the hard margin SVMs, soft margin can also be written as dual form. By solving the above with Lagrangian dual, we allocate the lagrange multiplier $\alpha, \beta\, \forall \alpha_i, \beta_i >= 0$ andto the constrict functions.

$$L_s(x) = F(x) - \alpha G(x) - \beta H(x)$$

where L is Lagrangian function and G and H are the constrict functions Here,

$$L_s = \frac{1}{2} * \|\vec{w}\|^2 + \frac{C}{n} \sum_{i=1}^{n} \xi_i - \alpha(y_i * (\vec{w} \cdot \vec{x_i} + b) - 1 + \xi) - \beta * \xi_i$$
$$\Rightarrow L_s = \frac{1}{2} * \|\vec{w}\|^2 + \frac{C}{n} \sum_{i=1}^{n} \xi_i$$
$$- \sum_{i=1}^{n} \alpha_i (y_i * (\vec{w} \cdot \vec{x_i} + b) - 1 + \xi) - \sum_{i=1}^{n} \beta_i * \xi_i\,, \forall_i$$

By solving the derivative of L with respect to $\vec{w}$ and b and ξ equal to zero,

$$\frac{\partial L}{\partial w} = 0 \Rightarrow w - \sum_{i=1}^{n} \alpha_i (y_i * \vec{x_i}) = 0 \dots (1)$$
$$\frac{\partial L}{\partial b} = 0 \Rightarrow \sum_{i=1}^{n} \alpha_i y_i = 0 \dots (2)$$
$$\frac{\partial L}{\partial \xi_i} = 0 \Rightarrow \frac{C}{n} - \alpha_i - \beta_i = 0 \dots (3)$$

From (3), we know that $\beta_i >= 0$, hence, we have

$$\frac{C}{n} >= \alpha_i$$

Substitute $(1), (2), (3)$ into L_s and maximize over it,

$$max L$$
$$\Rightarrow max \sum_{i=1}^{n} \alpha_i - \frac{1}{2} * \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j * (\vec{x_i} \cdot \vec{x_j})$$
$$\text{such that } 0 <= \alpha_i <= \frac{C}{n}, \forall i = 1, 2, \dots, n$$
$$\text{and } \sum_{i=1}^{n} \alpha_i y_i = 0$$

This is the dual form of soft margin SVMs.

5. Support vectors

The support vectors are those datapoints that have influence on the position of the hyperplane. Mathematically saying, the corresponding datapoints with $\alpha_i > 0$.

From Karush-Kuhn-Tucker condition,with a Lagrangain multipler α, $L = F(x) - \alpha G(x)$ satisfy the complementary slackness,

$$\sum_{i=1}^{n} \alpha_i G_i(x) = 0 \Rightarrow \alpha_i G_i(x) = 0, \forall_i$$

For hard margin SVMs, the support vectors are those datapoints that lie on the border. we have

$$\alpha_i (y_i * (\vec{w} \cdot \vec{x_i} + b) - 1) = 0, \forall_i$$

Either $alpha_i = 0$ or $(y_i * (\vec{w} \cdot \vec{x_i} + b) - 1) = 0$.
For $alpha_i = 0$, those datapoints lie outside the margin.
For $alpha_i > 0$,

$$(y_i * (\vec{w} \cdot \vec{x_i} + b) - 1) = 0$$
$$\Rightarrow y_i * (\vec{w} \cdot \vec{x_i} + b) = 1,\ \forall_i\, \alpha_i > 0$$

which means those datapoints lie on the margin, the hyperplane H_1 and H_2 we define in earlier section.

For soft margin SVMs, the support vectors are the datapoints that comprise the hyperplane and subject to ξ. we have

$$\alpha_i (y_i * (\vec{w} \cdot \vec{x_i} + b) - 1 + \xi) = 0, \forall_i \dots (I)$$

and

$$\beta_i (\xi_i) = 0, \forall_i \dots (II)$$

We know that from (3), $\frac{C}{n} - \alpha_i - \beta_i = 0$
Substitute (3) into (II),

$$(\frac{C}{n} - \alpha_i) * \xi_i = 0 \forall_i \ldots (III)$$

For $alpha_i = 0$, those datapoints lie outside the margin.
For $alpha_i > 0$, $(y_i * (\vec{w} \cdot \vec{x_i} + b) - 1 + \xi) = 0, \forall_i$ are the support vectors and comprise the hyperplane. Also, the bias b can be computed using those support vectors,that $\xi_i = 0, \alpha_i < \frac{C}{n}$ such that $y_i * (\vec{w} \cdot \vec{x_i} + b) = 1$, lies on the hyperplane H_1 and H_2, instead of using all the support vectors including those lying between the hyperplane.

6. Max Margin

The distance between the Hyperplane H and with respect to H_1 and H_2 is the margin. In SVMs, We want to maximize the distance between these hyperplane and the reason for that is we do not want to the hyperplane be too close to the datapoints which is not desirable on testing data.

For hard margin, maximizing the margin means maximizing the distance between the hyperplane H and the closest datapoints that lie on H_1 and H_2 for each class.

For soft margin, there is a parameter C controling the trade off between maximizing the margin and the number of misclassifications. For a large C, it penalizes the misclassifications heavily and thus it tries to reduce the number of misclassification as much as possible which lead to a narrower margin. For a small C, it has high tolerance level on misclassifications and allows more number of misclassifications lie between the hyperplane and results in larger margin.

7. Experimental Analysis

7.1. Introduction

The datasets are given as training set and testing set. The training data contains 8500 number of datapoints and each datapoint has 200 number of features and a target label with value 1 or 0. The testing data contains 1500 number of datapoints and the same number of features and label as training set.

7.2. Data preparation

Fristly, I will do data preparation. I change the label with value 0 to -1 to represent the negative class because it is more desirable or mathematically more convenient to use 1 and -1 as label in SVMs implementation.

Secondly, the traning data is split into training set with size 6800 and validation set with size 1700 as we want the size of validation set similar to the size of testing set.

K-fold cross validation is applied to training set for hyperparameter searching on parameter C. And we want a validation set to validate with the training error to make sure there is no over-fitting.

7.3. K-fold cross validation

Here, I use K-fold cross validation to do the hyperparameter grid search for the parameter C. I decided to use 5 fold cross validation which means 4 folds as training set and one fold as the validation set. I use the sklearn SVMs model and GridsearchCV to speed up the process because the time complexity of using cvxopt to optimize SVMs is high.

The K-fold cross validation gridsearchs the below C values and found that choosing 10 has the best accuracy score.
'C': [0.1,0.3,1,3,10,33,100,333,1000]

7.4. Model fitting

The best values of C with highest average accuracy score will then fit to the three models, sklearn 3rd party SVMs, Primal SVMs and Dual SVMs built by cvxopt optimization.

The weight vectors and bias obtained from those model are then compare to each other. The training, validation and test accuracy are also calculated with respect to all 3 models.

7.5. Weight and bias comparison

In this section, I will compare the weight vectors and the bias term obtained from sklearn, primal form and dual form.

In dual form, the weight and bias term are calculated by α. Weight vector is calculated as below $w = \sum_{i=1}^{n} \alpha_i * y_i * x_i$ Bias term uses those support vectors that lie on the hyperplane H_1 and H_2 which means $y_i * (\vec{w} \cdot \vec{x_i} + b) = 1$. Therefore, we need $\xi_i = 0$ and $\alpha_i < \frac{C}{n}$, hence,

$$b = \frac{1}{n} \sum_{i=1}^{n} (y_i - x_i * w)$$
$$\text{for } 0 < \alpha_i < \frac{C}{n}$$

Note: If using all the support vector to calculate bias, then the bias term of dual SVMs would have a larger difference when comparing with primal and sklearn bias.

I use the Euclidean distance to calculate the difference between the weight vector and absolute different between bias. The below table shows the result. The right upper triangle shows the weight different and the lower left one shows the bias different.

Bias, Weight	Primal	Dual	sklearn
Primal		4.59731e-14	0.000674278
Dual	0.00014195		0.000674278
sklearn	0.00016032	1.836919e-05	

The weight vector in Primal and Dual are the same but there is a very small different in bias term. The weight vector from sklearn model has a slight different then the one in cvxopt but it is almost negligible. The bias terms in dual and sklearn are more close to each other. The bias term different is very small among those models.

7.6. Accuracy on training,validation and testing data

The table are the accuracy training,validation and testing data with respect to Priaml, Dual and sklearn model.

Accuracy	Primal	Dual	sklearn
Training	0.978235	0.978235	0.978235
Validation	0.97	0.9694117	0.9694117
Testing	0.975333	0.975333	0.975333

The accuracy of among these 3 models in training, validation and testing are the same except the validation accuracy of primal is slightly better.

The training accuracy is higher than validation and testing accuracy as expected and the validation accuracy shows there is no overfitting.

The testing accuracy has 97.53% which is satisfying.

7.7. Runtime on Primal,Dual and sklearn model

The runtime performance of cvxopt is much slower than the sklearn model as shown in the below table.

	Primal	Dual	sklearn
Runtime(second)	178	166	1.5

8. Code

The code can be simply run by:
python3 SVM.py

9. Conclusion

In conclusion, the Primal and Dual models built by cvxopt are performing as well as the 3rd party SVMs model. But the time complexity on running cvxopt optimization is much higher the sklearn SVMs.

References

https://static1.squarespace.com/static/58851af9ebbd1a30e98fb283/t/58902fbae4fcb5398aeb7505/1485844411772/SVM+Explained.pdf

https://stats.stackexchange.com/questions/23391/how-does-a-support-vector-machine-svm-work/353605#353605