Super vector machine

Ziyan Wang
University of Adelaide
a1713103@student.adelaide.edu.au

Abstract

This paper is to introduce binary class linear SVMs and comparing the result of the implemented a methods and the library results from libsvm.

1. Introduction

In this assignment, I use Matlab as the main tool. In this paper, I will briefly introduce concept of SVMs and then the hard margin and soft margin of both primal and dual form of the SVM. In the final section, I will provide the experimental results of comparing the result in libsvm, soft margin dual and primal implementation.

2. Related work

Tom Mitchell defines machine learning as : A learning program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E [2]. The main process of machine learning is to prepare a training data set, then feed the data set to a learning algorithm. By calculating the data set, the algorithm will produce a function, which is the hypothesis. The hypothesis could then be used to process the inputed data and give the estimated results. The hypothesis that produced by learning algorithm could been considered as a kind of machine.

3. Support vector

Support vectors are a set of data that lie closest to the hyperplane and coming from the training samples. With the help of support vectors, the devision function might be modified it the support vector is changed. Support vector will be used to sequentially optimize the error function, depending on which, there are two types of the support vector. For type 1, if the set is linear separable, support vector will only contains the difficult classified data points that closest to the decision boundary. For type 2, if the set is not linear separable, support vector will be subject to with a slack, that will ignore limited difficult classified data points.

4. understanding of binary class linear SVMs

SVMs, which is the support vector machines, is an algorithm that could be used for classification or regression problems [1]. In one particular case, for separating a set of linear separable data, in 2 dimensions case we could use a line to classify the data points (in 3d use a hyperplane). However, the parameters that defines a available line for separating the data is infinite.

To getting the best hyperplane (line), there two intuitions. One is the functional margin. which write as $\vec{\omega}\vec{x} + \omega_0 = 0$. The other intuition is the geometric margin, given by $\frac{z_i(\vec{\omega}^t x_i + \omega_0)}{||\vec{\omega}||}$.$(z_i \in -1, 1)$

To sum up, SVMs for linearly binary class is to get a line(hyperplane) that classifies all training vectors into two classes as better as possible. For getting the best hyperplane, the best choice is the hyperplane that leaves the maximum margin from both classes.

We define the separating hyperplane as equation as following.

$$g(\vec{x}) = \vec{\omega}^T\vec{x} + \omega_0 = 0$$

The distance between hyperplane and closest data point:

$$\frac{|\vec{\omega}^T\vec{x} + \omega_0|}{||\vec{\omega}||}$$

$\vec{\omega}$ is a weight vector, $\vec{x}$ is the data set while ω_0 is the bias. We set the points on the boundary as a support vector and others could be ignored, those for which $|\vec{\omega}^T\vec{x_i} + \omega_0| = 1$ (x_i is a example closest to boundary).

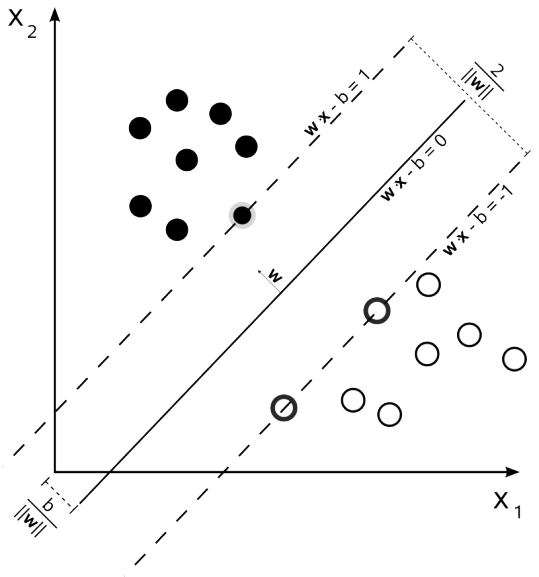

Figure 1. SVM

As you can see in the figure 1 [3].

5. Soft margin and hard margin for Primal form

For the closest point x_i, we set g(x_i)=1, then we get the distance:

$$\frac{1}{||\vec{\omega}||}$$

The total margin will become $\frac{2}{||\vec{\omega}||}$ which is the maximize margin. Based on this, to maximize the margin could be seen as find the $\vec{\omega}$ and ω_0 that minimize the $||\vec{\omega}||$.

The constrains are

$$g(\vec{x}) \geq 1(\forall \vec{x} \in \mathbb{R}^+)$$
$$g(\vec{x}) \leq -1(\forall \vec{x} \in \mathbb{R}^-)$$

This could be understand as separating data into two classes, when $g(\vec{x}) \geq 1$ the $\vec{x}$ belongs to class 1, when $g(\vec{x}) \leq -1$ the $\vec{x}$ belongs to class 2. In order to maximize the margin, we will minimize $||\vec{\omega}||$, subject to there are no data points in the margin which is the hard margin case.

The problem is convert to

$$max\frac{2}{||\vec{\omega}||}$$

subject to

$$z_i(\vec{w^t}x_i) \geq 1, \forall i$$

$$z_i = 1, \forall x_i \in \mathbb{R}^+$$
$$z_i = -1, \forall x_i \in \mathbb{R}^-$$

Based on former, the primal formula can become a constrained optimization problem and write as:

$$minf(\vec{w}) = \frac{1}{2}||\vec{w}||^2$$

and subject to

$$z_i(\vec{w^t}x_i + \vec{w_0}) \geq 1, \forall i$$

However, when the data points, in some case, are not linear separable, we introduce the slack variables to allow some errors or misclassify. And use the penalty function to control the errors. This is the soft margin for primal.

$$minf(\vec{w}) = \frac{1}{2}||\vec{w}||^2 + C\sum_{i=1}^{n}\xi_i$$

and subject to

$$z_i(\vec{w^t}x_i + \vec{w_0}) \geq 1 - \xi_i, \forall i \in \{1, n\}$$

6. Soft margin and hard margin for Dual form

If use quadratic programming to solve this problem, the complexity will relate to the number of samples and dimensions. If the dimension of the sample is infinite, then efficiency of solving this problem will be low. For make the complexity of the problem only relate to the number of samples, we use Langrange multipliers α to convert the primal problem into dual form. In our case minimizing $\vec{w}$ is a nonlinear optimization task, we introduce the Langrange multipliers α and convert the problem as:

$$L(\alpha, \omega) = \frac{1}{2}||\vec{\omega}||^2 + \sum_{i=1}^{n} \alpha_i [z_i(\omega^t x_i + \omega_0) - 1]$$

The dual problem can be write as

$$\min_{\omega_0, \vec{\omega}} (\max_{\alpha} Ł(\omega_0, \vec{\omega}, \alpha))$$

By setting

$$\frac{\partial L(\alpha, \omega)}{\partial \vec{\omega}} = 0$$

we can get the relationship:

$$\vec{\omega} = \sum_{i=0}^{n} \alpha_i z_i \vec{x_i}$$

By partial the ω_0 to the function

$$\frac{\partial L(\alpha, \omega)}{\partial \vec{\omega}_0} = 0$$

we can get

$$\sum_{i=0}^{n} \alpha_i z_i = 0$$

For soft margin, we introduce the slack variables $\xi = \xi_1, \xi_2, ..., \xi_n$

$$f(\vec{\omega}, \xi) = \frac{1}{2}||\omega||^2 + C \sum_{i=1}^{n} \xi_i$$

, subject to

$$\forall i, z_i(\vec{\omega}^t x_i + \omega_0) \geq 1 - \xi_i, \xi_i \geq 0$$

By using Karush-Kuhn-Tucker (KKT) confitions the problem could become:
maximize:

$$Ł(\omega_0, \vec{\omega}, \alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j}^{n} \alpha_i \alpha_j z_i z_j x_i^t x_j$$

that constrains to

$$\alpha_i \geq 0, \forall i, \sum_{i=1}^{n} \alpha_i z_i = 0$$

Which to sum up is:

$$\max_{\alpha}(\sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{m} z_i z_j \alpha_i \alpha_j x_i x_j)$$

subject to

$$\alpha_i \geq 0, i \in \{1, 2, ..., m\}$$

$$\sum_{i=1}^{m} \alpha_i z_i = 0$$

We could find the optimal α, which when $\alpha_i = 0$, sample x_i is not a support vector, when $\alpha_i \neq 0$ and $z_i(\vec{\omega}^t x_i + \omega_0 - 1) = 0$ x_i is support vector.

Then by use the α in the

$$\vec{\omega} = \sum_{i=0}^{n} \alpha_i z_i \vec{x_i}$$

, we can get the $\vec{\omega}$.

After get the $\vec{\omega}$, the b could then be calculated by using

$$\forall \alpha_i \leq C \wedge \alpha_i \geq 0, \alpha_i[z_i(\vec{\omega}^t x_i + \omega_0) - 1] = 0$$

because $\alpha_i \geq 0$, the formula could become

$$\omega_0 = \frac{1}{z_i} - \vec{\omega}^t x_i$$

$$\omega_0 = -\frac{1}{2}(\max_{i, z_i = -1} \omega^T x_i + \min_{i: z_i = 1} \omega^T x_i)$$

7. Concepts of generalization/test error

Generalization error is an indicator that we use to measure the accuracy of the predicting result that produced by machine learning algorithm. The model is trained by limited data set, we want the model have good performance or have good estimating ability in different data set. Overall, after we train a model, we got the parameters of the hypothesis. The error that appear in the test set with the model is the generalization error.

8. Why max margin is good

There are infinite candidates of the hyperplane (shown in figure 2) , but the one with the maximum-margin could separate two data sets is at the middle position. This means, the margin between the hyperplane to the closest points of both data sets are maximums. In this condition, if there are new data points that appear around or in the margin will have greater possibility to be classified into the correct dataset.

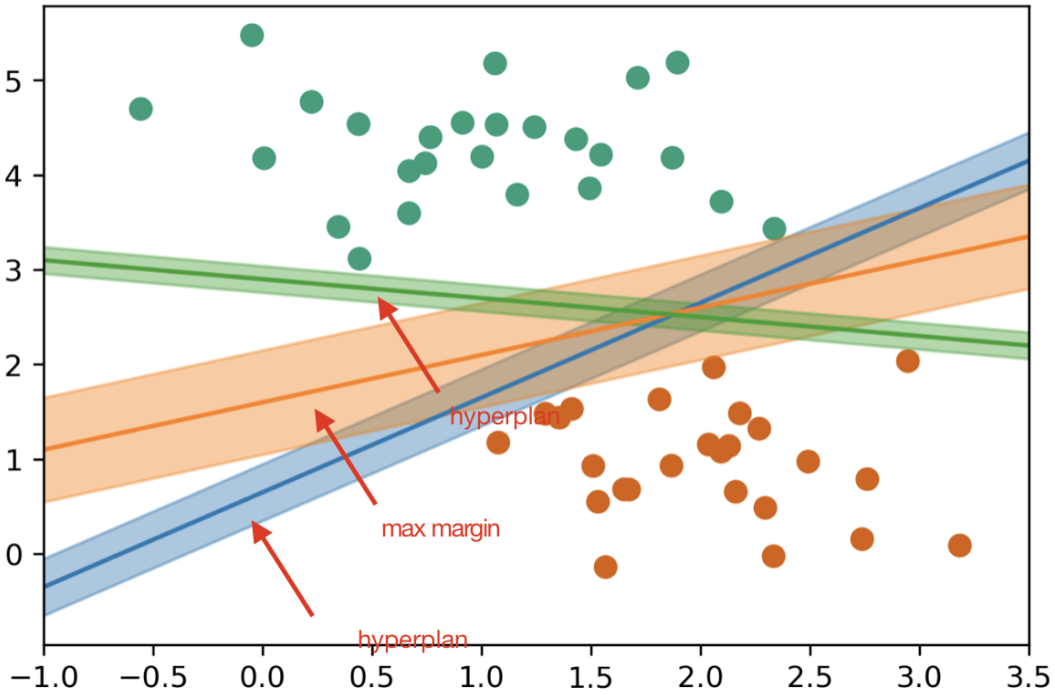

Figure 2. hyperplane candidates

9. Experimental results comparing with libsvm

9.1. Experiment 1: compare ω and b

Through solving the primal problem we get the result of ω and b. By solving the dual problem, we can get the α. And based on the formula

$$\vec{\omega} = \sum_{i=1}^{n} \alpha_i z_i x_i$$

We can reconstruct $\vec{\omega}$. Then the ω_0 can be solved by use the

$$\alpha_i[z_i(\omega^t x_i + \omega_0) - 1] = 0$$

.

The w and b from primal and dual in soft margin has been shown in the following table. The b value produced by primal is larger then b of dual by using the above method. However the accuracy is both over 90 %. The ω value when C=1, from libsvm, dual and primal have been shown in frigure 3. The primal and dual $\vec{\omega}$ are relative similar comparing with value from libsvm. The accuracy and value when C=10, has also been listed as follow. When C=10, the daul is almost hard margin, confider this the accuracy should be higher then C=1, but actually, it is worse then C=0 i dual case.

9.2. Experiment2

check the duality gap of both the results and the formulations

Through implementing the both dual and primal problem, the gap could be calculated by comparing the optimal values. In C=10 case, the objective value from bual is over 150 larger then primal.

Name	Accuracy(C=1)	b
Primal	0.9713	3.1171
Dual	0.9167	1.0092

	1
1	2.5827
2	-2.7272
3	14.1321
4	-9.1684
5	7.2465
6	-15.5411
7	7.1617
8	-3.6951
9	-1.2309
10	3.3762
11	8.3421
12	-2.3585
13	-2.3540
14	2.7971
15	6.4780
16	-5.8078
17	2.7847
18	-1.7257

	1
1	0.0036
2	0.0032
3	7.2462e-04
4	-0.0056
5	0.0057
6	-0.0045
7	0.0020
8	0.0043
9	0.0022
10	0.0068
11	0.0042
12	-4.3053e-04
13	0.0016
14	0.0022
15	-3.4639e-04
16	-8.3078e-04
17	-2.7994e-04
18	-0.0065

	1
1	-0.0559
2	-0.1230
3	0.0675
4	-0.0576
5	0.1123
6	-0.0234
7	0.1198
8	-0.0181
9	-0.0167
10	0.0151
11	0.1127
12	-0.0364
13	0.0843
14	0.0902
15	0.0311
16	-0.0287
17	-0.1016
18	-0.0703

Figure 3. w of libsvm and dual and primal

Name	Accuracy(C=10)	b
Primal	0.9760	3.1989e+03
Dual	0.8653	1.1605e +06

Name	objective function value	gap
Primal	+699.6442	152.556
Dual	+547.088	

9.3. Experiment3

compare the w b α with the libsvm

The result of w and b and the accuracy value has been listed in the following tables. The accuracy rate is $\frac{\sum testy \in (sign(testdata * w + \omega_0)}{sizeofy}$. The plot result of w in libsvm, dula and primal form has been shown in figure4 and figure 5

Name	Accuracy(C=1)	b
Primal	0.9713	3.1171
Dual	0.9107	1.0092
Libsvm	0.994	0.0243

Name	Accuracy(C=10)	b
Primal	0.9760	3.1989e+03
Dual	0.8653	1.1605e +06
Libsvm	0.994	0.0243

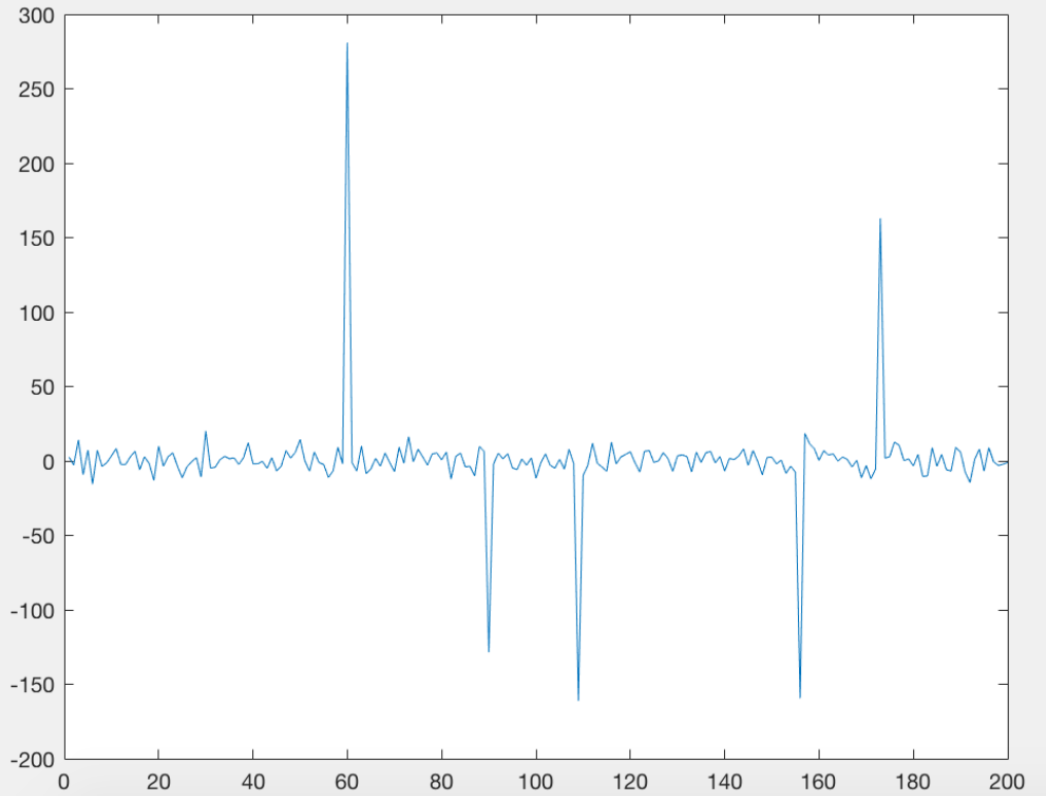

Figure 4. w from libsvm

9.4. Experiment4

The accuracy of training and testing errors of primal and dual forms and libsvm has been listed as follow. For primal and dual case, the value of C has different impact.

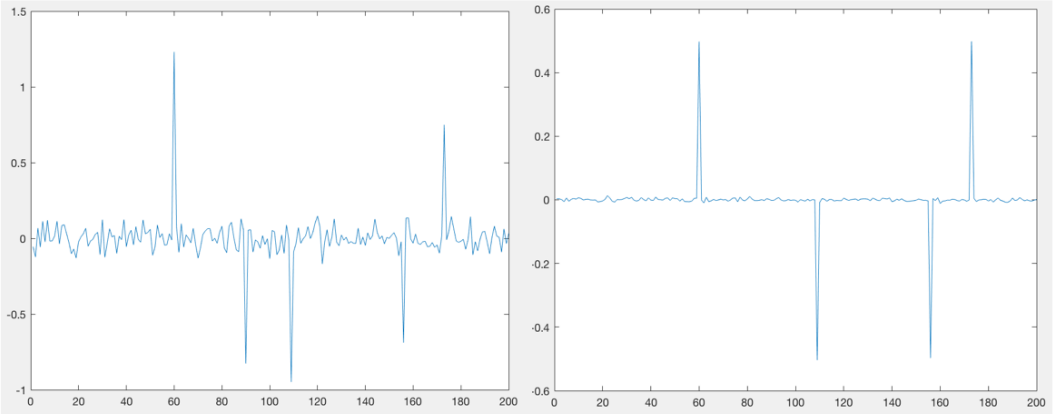

Figure 5. w(from left to right: prim dual)

Name	C=1	C=10
Primal	0.9713	0.9760
Dual	0.9167	0.8653
Libsvm	0.994	0.994

References

[1] Advantage of svm.

[2] Machine learning, 2017.

[3] Support vector machine, August 2018.