Concept of SVM

Support vector machine is a type of linear classification model for classifying datasets into their individual classes based on their distinctive features. It does so by creating a line or a hyperplane (depending on the dimensionality of the corresponding datasets) that separates them into classes. But there is a catch here, there are infinitely many lines/hyperplanes that can be created to simply separate the datasets.

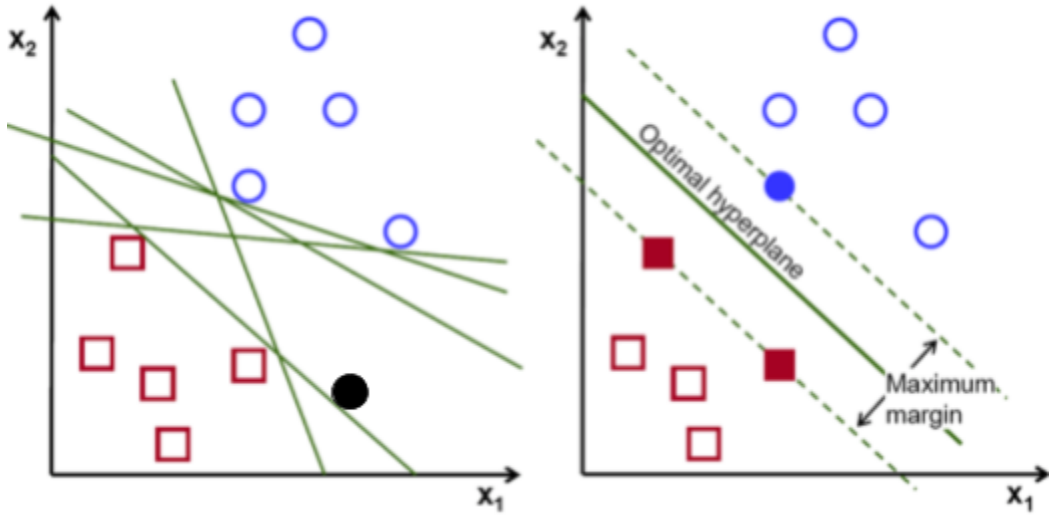

Figure 1 *(Source =https://towardsdatascience.com/support-vector-machine-vs-logistic-regression-94cc2975433f)*

For example, the plot on the left in figure 1 above shows multiple linear classifiers that can be used to easily separate the two datasets. However, choosing a hyperplane that is close to either one of the two clusters of datasets will result in a high test error rate because it cannot generalise well for new test data. Imagine if a new test data were to lie very close to this hyperplane but appear on the other side of the hyperplane relative to the red datasets, it will be classified as a blue dataset even though it makes more sense to classify it as red dataset as shown by the black dot. SVM is set out to counter this problem by the following :

Concept of Margin and Support Vectors

The margin is referring to the distance from the hyperplane to the closest data from each of the classes. SVM is about maximising this margin such that the classification of data becomes more certain and reliable such that the black dot will either be treated as an error or correctly classified as part of the red datasets depending on the type SVM margin. The support vectors are the data that lie the closest to the hyperplane such that moving the support vectors would also move the decision hyperplane. The support vectors are the one that will dictate how the decision boundary will be like, meaning adding more training data off of the boundaries of the margin will not do anything to affect the hyperplane.

The formulation of the Maximum Margin

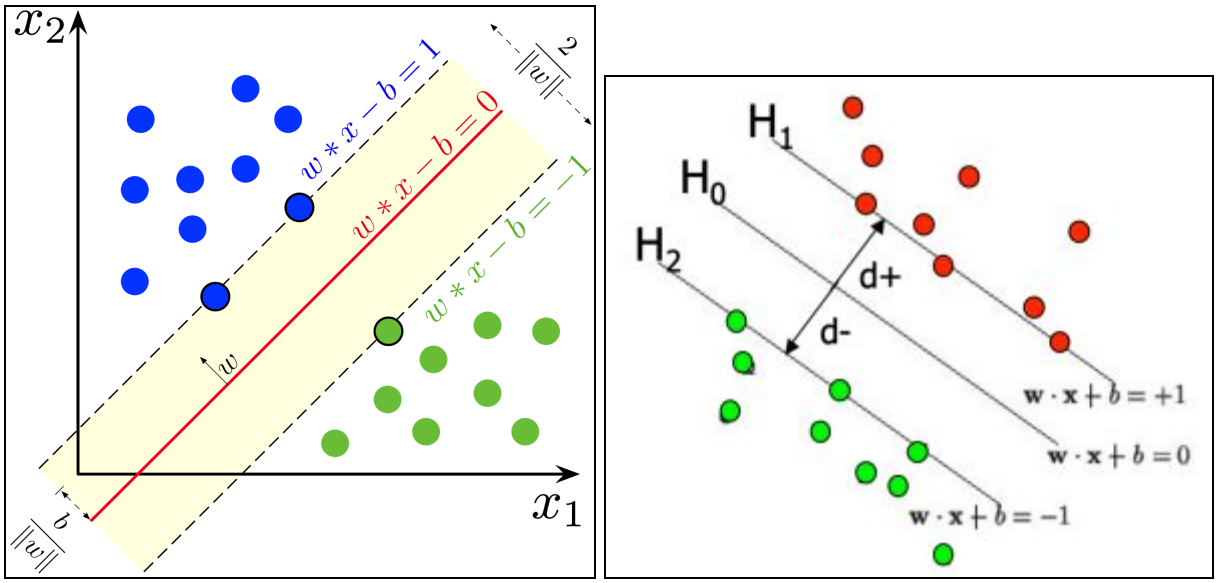

Figure 2 : Left plot = (Source = n.wikipedia.org/wiki/Support_vector_machine#/media/File:SVM_margin.png)
Right plot = (Source = http://web.mit.edu/6.034/wwwbob/svm-notes-long-08.pdf)

Given that the hyperplane is in the form of $w^t * x + b = 0$ where w^t is the weight vector, x is the input vector (features) and b is the bias term. This allows us to formulate $w^t * x + b = 1$ referring to the boundary for the positively classified data and $w^t * x + b = -1$ referring to the boundary for the negatively classified data assuming that the distance between any support vectors and the hyperplane is 1. These boundaries are where the support vectors lie and they will dictate how the shape of the hyperplane. Given that the distance from a point (x_1, x_2) to a line $ax + by + c = 0$ is $\frac{|ax+by+c|}{\sqrt{a^2+b^2}}$ then the distance from H_0 *to* H_1 is $\frac{|w^t*x+b|}{\|w\|} = \frac{1}{\|w\|}$, similarly the distance from H_0 *to* H_2 $\frac{|w^t*x+b|}{\|w\|} = \frac{|-1|}{\|w\|} = \frac{1}{\|w\|}$. Given the distance from H_0 *to* H_1 and H_0 *to* H_2, the margin is equal to $\frac{2}{\|w\|}$ as shown in the figure above. In order to maximise the margin $\frac{2}{\|w\|}$, we need to minimise $\frac{1}{2}\|w\|^2$ which forms the quadratic optimisation problem. For convenience sake, we can formulate the constraint as $y * (w^t * x + b) \geq 1$ by introducing a variable y such that when $w^t * x + b$ is $+ve$ the y will be 1 and when $w^t * x + b$ is $-ve$ the y will be -1. This quadratic optimisation problem with this particular constraint is also known as the SVM

Hard-margin Primal problem:

$$min_{w,b} \ \frac{1}{2}\|w\|^2 \text{ s.t } y * (w^t * x + b) \geq 1 \ \forall i$$

This means that there will be no misclassification allowed such that all test data must be above or below the support vector boundaries in order to be correctly classified.

Hard-margin Dual Problem:

We can also solve the hard-margin primal constrained optimization problem by transforming it into a **dual problem** using the Lagrange multipliers method. This method will transform the constrained optimization problem into an unconstrained one better known as **generalized Lagrangian** by subtracting each constraint from the original objective function and multiplied by a new variable, ' α ' called Karush–Kuhn–Tucker (KKT) multipliers :

$$L\ (w,\ b,\ \alpha) = \tfrac{1}{2}||w||^2\ - \sum_{i=1}^{m} a^{(i)}(y^{(i)}(w^t * x^{(i)} +\ b) - 1)\ with\ a^{(i)} \geq 0\ for\ i = 1, 2, ..., m$$

To find the solution to this problem, we need to find the stationary points of $(w,\ b,\ \alpha)$ by computing their partial derivatives such that the solution meets the KKT conditions. The partial derivatives of the generalized Lagrangian in terms of w and b is as follows:

$$\text{In terms of w} = [\ \nabla_w L\ (w,\ b,\ \alpha) = w\ -\ \sum_{i=1}^{m} a^{(i)}y^{(i)}x^{(i)}\ ] \text{ and } [\ \tfrac{d}{db} L\ (w,\ b,\ \alpha) = -\sum_{i=1}^{m} a^{(i)}y^{(i)}\ ]$$

When these partial derivatives are equal to 0:

$w\ = \sum_{i=1}^{m} a^{(i)}y^{(i)}x^{(i)}$ and $\sum_{i=1}^{m} a^{(i)}y^{(i)} = 0$, If we plug these back into the generalized Lagrangian then we would get rid of the dependence on w and b and get the following:

Final Form:

$$\text{maximise } L\ (w,\ b,\ \alpha) = \sum_{i=1}^{m} a^{(i)} - \tfrac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} a^{(i)}a^{(j)}y^{(i)}y^{(j)}x^{(i)} \text{ s.t } a^{(i)} > 0 \text{ and } \sum_{i=1}^{n} a^{(i)}y^{(i)} = 0$$

The goal is to find the optimal α that maximises this function, once the α is found we can compute the $w\ = \sum_{i=1}^{m} a^{(i)}y^{(i)}x^{(i)}$ by plugging the α back into this equation. (Note that most of the α will have values of zero and the α that is not zero are support vectors) To get b we need to compute this equation $b = \tfrac{1}{n} \sum_{i=1}^{m} y^{(i)}\ - w^t * x^{(i)}$, this is derived from the fact that a support vector must verify $y^{(i)}(w^t * x^{(i)} +\ b\ ) = 1$ such that $a^{(i)} \geq 0$. The solution that we find here should be the same as that of the primal problem.

Soft-margin Primal problem:

$$min_{w,b,\xi}\ \tfrac{1}{2}||w||^2 + C * \tfrac{1}{n} \sum_{i=1}^{n} \xi^{(i)} \text{ s.t } y^{(i)} * (w^t * x^{(i)} + b)\ \geq 1 - \xi^{(i)} \text{ and } \xi^{(i)} \geq 0$$

In order to counter the Hard-margin primal problem sensitiveness to outliers, we introduce a slack variable, ξ to allow constraints to be violated such that a data that is closer to the hyperplane or on the wrong side can still be classified but a loss is incurred for every misclassification on the objective function. The C is known as the hyperparameter when C is small, the SVM becomes very loose and allows for a lot of margin violations. **When C is large, the SVM becomes very strict and tries to prevent margin violations. The goal of a soft-margin problem is to strike a good balance between keeping the margin as large as**

possible while also limiting the margin violations because they are for the objective function. However, in the case where C is small, it tends to generalize better than when C is large.

Soft-margin Dual Problem:

To transform the soft-margin primal constrained optimisation problem into a dual problem, we need to employ the Lagrange multipliers method and it will look like the following:

$$L(w, b, \xi, a, \beta) = \frac{1}{2}||w||^2 + C * \frac{1}{n}\sum_{i=1}^{n}\xi^{(i)} + \sum_{i=1}^{n} a^{(i)}[(1 - \xi^{(i)}) - y^{(i)}(w^T x^{(i)} + b)] + \sum_{i=1}^{m}\beta^{(i)}(-\xi^{(i)})$$

Where a *and* β are the Lagrange multipliers variables each must be ≥ 0. To find the dual problem, we need to find the stationary point of (w, b, ξ, a, β) where the Lagrangian is minimized. To do this, we need to set the **partial derivatives to zero with respect to** w, b *and* ξ :

With respect to $w = \sum_{i=1}^{n} a^{(i)}y^{(i)}x^{(i)}$, With respect to b $(\sum_{i=1}^{n} a^{(i)}y^{(i)} = 0)$ and With respect to $\xi^{(i)}$ $(\beta^{(i)} = C - a^{(i)})$ the fact that $\beta^{(i)} \geq 0$ leads to the constraint of $(0 \leq a^{(i)} \leq C)$. If we plug all of these results back into the Lagrangian function, we would get a reduced Lagrangian that depends only on a *and* β :

maximise $L(w, b, \alpha) = \sum_{i=1}^{m} a^{(i)} - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} a^{(i)}a^{(j)}y^{(i)}y^{(j)}x^{(i)}$ s.t $0 \geq a^{(i)} \geq C$ and $\sum_{i=1}^{n} a^{(i)}y^{(i)} = 0$

Notes about C: It allows data points to be on the boundary as well as data points on the wrong side of the boundary to be included. The degree in how much error or points is tolerated depends on the magnitude of C.

Variance and Bias Tradeoffs

Bias is about how well a model can accurately replicate the "true" relationship between two datasets or in other words how well it can fit the training set and Variance is about how well a model can fit the testing set. The hard-margin problem has low bias and high variance because it is very sensitive to the training data but performs poorly on the testing data. The soft-margin, on the other hand, has high bias and low variance because it is less sensitive to the training data but performs better on the testing data. In order to find the best SVM model for classification, we need to run cross-validation on the SVM model with different parameters and choose the best-performing ones.

Experiments

Note: The following data are obtained from the cross Validation of both the primal and the dual (Follows the Primal_Dual_run(3).mat -- saved workspace for the third run of the main)

Hyperparameter, C value	Primal, b	Dual, b	Difference, b	Difference, w ***(Average)***
0.01	0.6185	0.0337	0.5848	$9.9501 * 10^{-5}$
0.1	0.3836	0.3137	0.0699	$1.0837 * 10^{-4}$
1	0.6894	0.7053	0.0159	$2.3615 * 10^{-5}$
10	1.1606	1.1836	0.0230	$1.9661 * 10^{-4}$
20	1.3446	1.4576	0.1130	0.0013
30	1.4488	1.4505	0.0018	0.0044
40	1.5341	1.7856	0.2515	0.0025
50	1.5895	2.2567	0.6672	0.0079
60	1.6242	1.9401	0.3159	0.0042
70	1.6684	2.0368	0.3683	0.0036
80	1.7035	2.0929	0.3894	0.0039

(Comparison between Primal b and w with Dual b and w)

The difference between the soft-margin primal b values with that of the soft-margin dual b value for each of the hyperparameter, C value that was used is fairly small which indicates that the optimal hyperplane for both the primal and the dual problems are more or less in the same position (the y-intercept == bias term, b). The same can be said for the difference in the mean of the w terms between both the primal and the dual as the difference is fairly small which would indicate that the optimal hyperplane for both of the problems has the same slope. The main reason for this small difference or perhaps similarity is due to the fact that the soft-margin dual problem formulates the lower bound to the soft-margin primal problem.

Hyperparameter, C value	Validation_accuracy_Fold (Primal)	Validation_accuracy_Fold (Dual)	Test_accuracy	Test_accuracy_d	test_accuracy(Libsvm)
0.01	58.61	92.13	49.40	92.60	96.20
0.1	94.79	95.42	94.67	95.20	99.33
1	97.15	97.15	97.00	96.93	99.40

10	97.21	96.96	97.33	97.27	99.33
20	97.22	97.01	97.40	97.13	99.33
30	97.19	95.02	97.07	96.33	99.33
40	97.08	97.05	97.13	97.07	99.33
50	97.02	96.02	97.07	96.73	99.33
60	96.93	96.58	97.13	97.20	99.33
70	96.91	97.06	97.13	97.20	99.33
80	96.89	96.94	97.00	97.00	99.33

(Comparison between Primal and Dual validation accuracy, test accuracy and test_accuracy with LibSVM)

In both the primal and dual cases, the validation accuracies are very similar to the test accuracies which suggest that the svm training model is not overfitting. The accuracy rate for both the Primal and the Dual are consistent with the svm mathematical formulation as the hyperparameter, C value increases, the accuracy for both the validation and test increases as well. This is due to the fact as the C becomes bigger the optimisation algorithm will try to minimize the w as much as possible resulting in a hyperplane that tries to classify each training data correctly which also leads to a margin that is smaller which also means that it doesn't allow as many misclassifications as that of a small hyperparameter, C value. The accuracy rate for both the primal and the dual for each corresponding hyperparameter, C value should in theory be the same since the dual lower bounds the primal and it looks like it is somewhat consistent although there are small differences across the board excluding the case with the 0.01 hyperparameter, C value. However, the test accuracy from my own svm model differs by around 2% from the test accuracy obtained in the Libsvm. I am not sure why this is the case because I do not have a full understanding of how the libsvm internal mechanisms work for this calculation. It could be due to the fact that a kernel method was used in Libsvm.

(Cross Validation for the best Hyperparameter)

From the 5-fold cross validation runs, it was found that the best performing hyperparameter for the primal is 20 and the best performing hyperparameter for the dual is 1. In the case of the test accuracy, the best performing hyperparameter for the primal is 20 which is the same as that obtained from the cross validation but the best performing hyperparameter for the dual is 10 which is different. And in the case of the Libsvm test accuracy, it was found that the best performing hyperparameter is 1. Based on my findings, the best hyperparameter lies in between

1 to 20. However, in theory the best hyperparameter should be the same for both the primal and dual cases, the difference here could be due to the constraint on the support vectors used to formulate the w in the dual svm model. ($1 * 10^{-7} < al < (C/n - 1 * 10^{-7})$)

References

1. *Aurelien Geron.(2019) Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems (pg 220-231, pg 1016-1019). O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.*
2. *Nathaniel Hobbs.(May 5, 2018). Kernel SVM for Image Classification. Retrieved from http://www.nathanielhobbs.com/documents/cvx_opt/cvx_opt_final_report.pdf*
3.