Demystifying Maths of SVM — Part 1

Deriving the optimization objective of the Support Vector Machine for a linearly separable dataset with a detailed discourse on each step

Krishna Kumar Mahto
Jan 6, 2019 · 10 min read

So, three days into SVM, I was 40% frustrated, 30% restless, 20% irritated and 100% inefficient in terms of getting my work done. I was stuck with the Maths part of Support Vector Machine. I went through a number of YouTube videos, a number of documents, PPTs and PDFs of lecture notes, but everything seemed too indistinct for me. Out of all these, I found Andrew Ng's Stanford lectures the most useful. Although he falls a little short in his ability to convey everything he intends to, his notes and derivations flow down very smoothly. Whatever I am going to discuss was inspired 50% by Andrew Ng's lectures and his notes, 20% by one of the ML courses I am taking, and 29% by everything else and the rest of 1% comes from the little work which I put together into building this up. At the end, it turns out that it is not at all difficult to understand how SVM came up as all it takes is high school coordinate and vector geometry. For the most part, finding the right dots to make a sensible map was what I found difficult. With this article, I have tried to lay down the mathematical derivation which I came up with by affixing ideas from different sources, along with the thought process.

Let's begin …

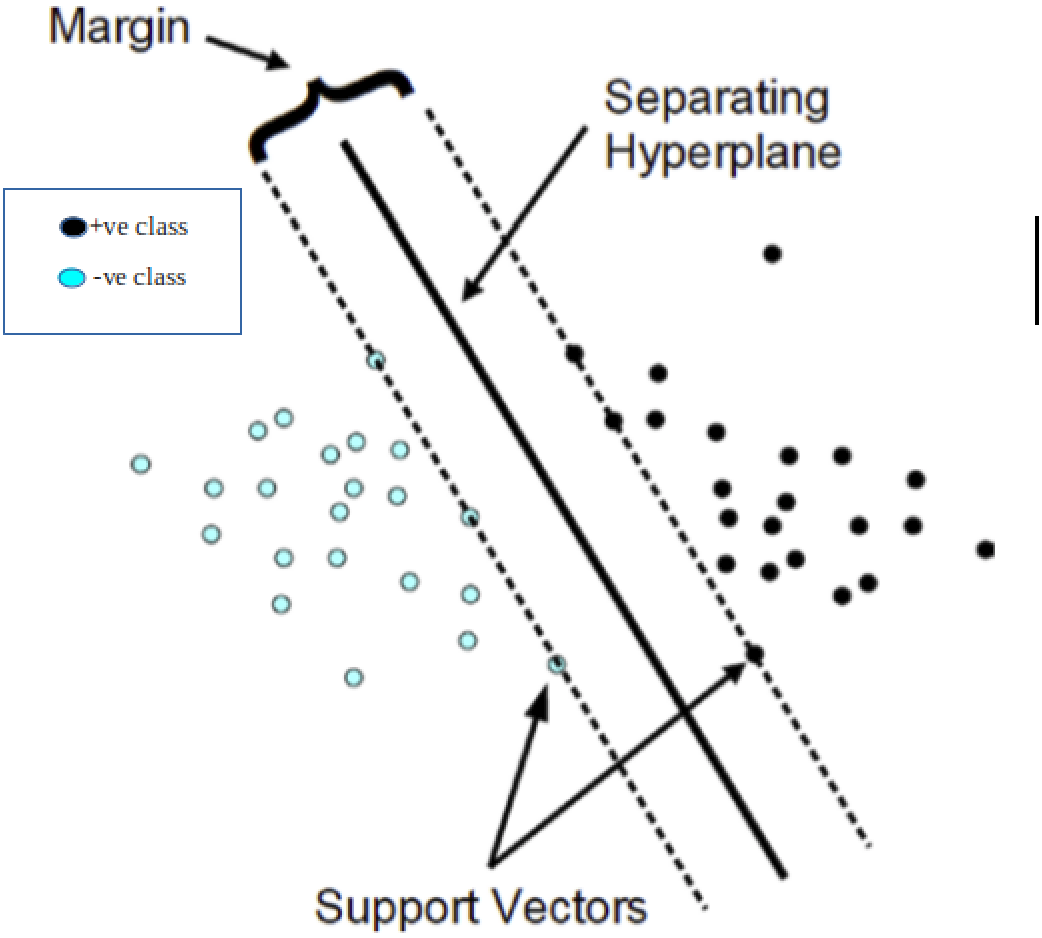


Fig 1. Diagrammatic representation of SVM for linearly separable dataset (Source: https://www.researchgate.net/figure/Classification-of-data-by-support-vector-machine-SVM_fig8_304611323)

The diagram does not look to be too worrying if you know SVM at a high conceptual level (the Optimal Margin Classifier stuff). Although the cases of linearly separable datasets are not seen in real life, discussion throughout this article on SVM will be for this context only. I might do a separate post for a more general version of SVM.

> ***SVM hypothesis***

Hypothesis, w.r.t. a machine learning model is the model itself, which is nothing but our classifier (which, is a function).

$$h_{w,b}(x) = g(w^T x + b)$$

g(z) = 1 if z ≥ 0, -1 otherwise

Class labels

Class labels are denoted as -1 for negative class and +1 for positive class in SVM.

$$y \in \{-1, 1\}$$

The final optimization problem that we shall have derived at the end of this article and what SVM solves to fit the best parameters is:

$$\min \frac{1}{2}\|w\|^2$$
$$s.t.\ y_i(w \cdot x_i + b) \geq 1,\ \forall x_i$$

Optimization problem that the SVM algorithm solves

This is a *convex optimization* problem, with a *convex optimization objective function* and a set of constraints that define a *convex set* as the *feasible region.* Convex functions look like a bowl placed right-side-up. Convex set is a set of points in which a line joining any two points lies entirely within the set. I would have loved to talk on these in more detail, but it would be more convenient to just google the terms in italics.

Before delving into the actual part, we should be familiar with two terms- *Functional margin and Geometric margin*.

Functional margin and Geometric margin

Following is how we are going to notate the hyperplane that separates the positive and negative examples throughout this article:

$$\pi:\ w^T x^{(i)} + b = 0$$

Equation of separating hyperplane; w is the normal to the hyperplane

Each training example is denoted as *x*, and superscript *(i)* denotes ***i***th training example. In the following section *y* superscripted with (i) represents label corresponding to the ***i****th training example.*

> *Functional margin* **of a hyperplane** *w.r.t.* ***i***th taining example *is* defined *as:*

$$\hat{\gamma}^{(i)} = y^{(i)}(w^T x^{(i)} + b)$$

Functional margin of a hyperplane w.r.t. ith example (denoting as gamma-hat superscripted with (i))

Functional margin ***of a hyperplane*** w.r.t. the entire dataset is defined as:

$$\hat{\gamma} = \min_{i=1,\ldots,m} \hat{\gamma}^{(i)}$$

Functional margin of a hyperplane w.r.t. the entire dataset

> *Geometric margin* **of a hyperplane** *w.r.t.* ***i****th training example is defined as functional margin normalized by norm(**w**):*

$$\gamma^{(i)} = y^{(i)}(\frac{w^T x^{(i)} + b}{||w||})$$

Geometric margin w.r.t. ith training example (denoted as gamma superscripted with (i)).

Geometric margin for a hyperplane w.r.t. the entire dataset is defined as:

$$\gamma = \min_{i=1,\ldots,m} \gamma^{(i)}$$

Geometric margin of hyperplane w.r.t. the entire dataset

> ***Note****: In the following discussion, if it is not specified whether the functional/geometric margin of a hyperplane is mentioned w.r.t. the entire dataset or some example, then it should be assumed to be in reference to the entire dataset, and not a single example.*

Brief on how SVM algorithm works, what it wants to achieve (interpreting SVM conceptually)

Just to make sure we are on the same page, lets discuss how SVM works. I have come across two interpretations of SVM (or more precisely, the goal that SVM aims at achieving). Both the interpretations as quoted below are just different ways of conveying the same thing as we shall see when we derive the optimization objective.

First,

> *SVM maximizes the margin (as drawn in fig. 1) by learning a suitable decision boundary/decision surface/separating hyperplane.*

Second,

> *SVM maximizes the geometric margin (as already defined, and shown below in figure 2) by learning a suitable decision boundary/decision surface/separating hyperplane.*

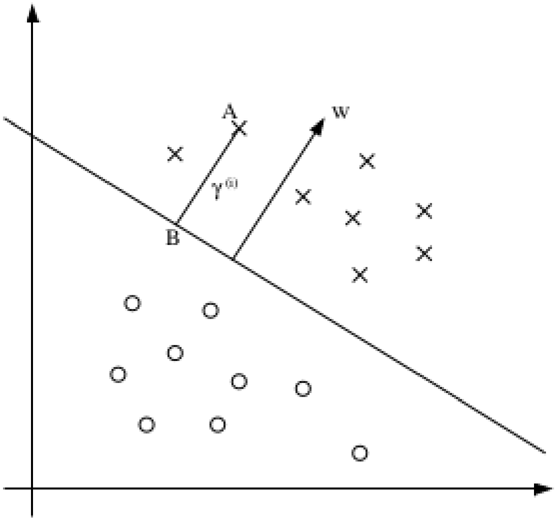


Fig. 2. A is ith training example, AB is the geometric margin of hyperplane w.r.t. A

The way I have derived the optimization objective starts with using the concepts of functional and geometric margin; and after establishing that the two interpretations of SVM coexist with each other, the final optimization objective is derived.

The derivation

As said, we shall start with functional and geometric margin interpretation, and then establish the coexistence of the two interpretations of SVM.

> *Could we have gone the other way round?*

I tried doing that, but it turned out to be not a better way to proceed by starting with the first interpretation. We shall discuss why that would not have worked out this well once we have derived the formulation of the optimization objective.

As already discussed, SVM aims at maximizing the geometric margin and returns the corresponding hyperplane. What it means is that out of all possible hyperplanes (each hyperplane has a geometric margin w.r.t. the point closest to it which is the least of all other geometric margins defined w.r.t. all other points), SVM chooses that hyperplane which has the maximum geometric margin. In fig. 3, the red hyperplane is the best separating hyperplane.

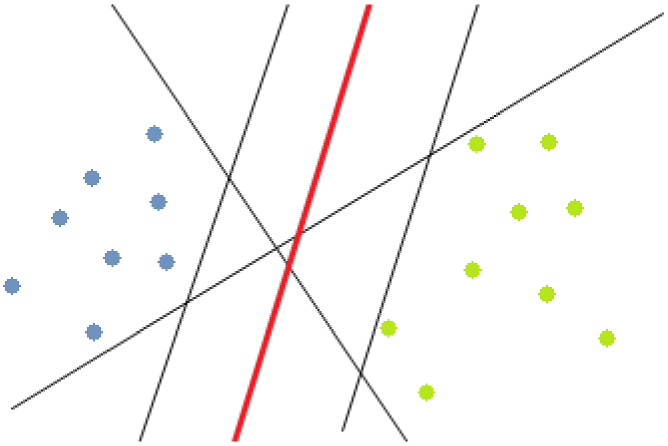


Fig. 3. Which hyperplane is the best? — the one in red

This can be mathematically written as,

$$(w^*, b^*) = \underset{w,b}{\arg\max} \ \frac{(w^T x + b) y^{(i)}}{||w||}$$

$$\text{s.t.} \quad y^{(i)} (w^T x^{(i)} + b) \geq w^T x + b \quad \forall i$$

$$\text{or,} \quad (w^*, b^*) = \underset{w,b}{\arg\max} \ \gamma$$

$$\text{s.t.} \quad y^{(i)} (w^T x^{(i)} + b) \geq \hat{\gamma} \quad \forall i$$

Initial optimization problem

Apparently, the objective function is the geometric margin of the hyperplane ***(w**, b)*. The constraints represent the fact that the objective function is the minimum of the set of geometric margins of the hyperplane ***(w**, b)* w.r.t. all the training examples.

We can observe that the formulation of geometric margin ensures that for the optimal hyperplane, the value of the smallest of all geometric margins (computed w.r.t. all examples) is shared by atleast one pair of examples (one pair from +ve class and the other from -ve class). Such points are called as ***support vectors*** (fig.- 1).

Therefore, the optimization problem as defined above is equivalent to the problem of maximizing the *margin* value (not geometric/functional margin values). Margin is defined as the distance between two hyperplanes, each of which is parallel to the separating hyperplane and passes through support vectors of each class (i.e., one hyperplane passes through support vectors of +ve class, while the other hyperplane passes through support vectors of -ve class, and both are parallel to the separating hyperplanes).

> *At this point, therefore, we can establish that both the interpretations of SVM actually coexist, although we started with the second interpretation to come to this conclusion. Towards the end, I shall discuss that while going the other way round, we carry some redundancy in our formulation and therefore, start off with less cleaner ideation of the problem.*

Let us denote the respective hyperplanes as:

$$\pi_+ : w^T x^{(i)} + b_+ = 0$$

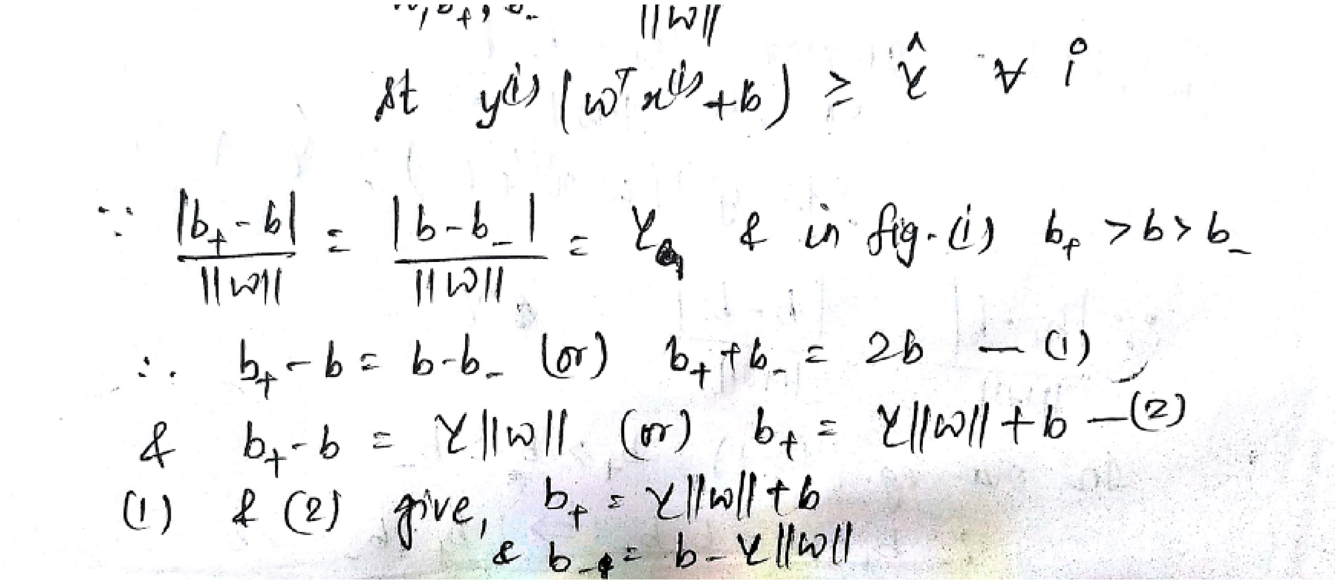
Hyperplane passing through positive support vectors

$$\pi_- : w^T x^{(i)} + b_- = 0$$

Hyperplane passing through negative support vectors

Therefore, the optimization problem can be reformulated with the following objective function:

$$(w^*, b^*) = \arg\max \frac{|b_+ - b_-|}{}$$



$$w, b_+, b_- \quad \|w\|$$

$$\text{s.t.} \quad y^{(i)}(w^T x^{(i)} + b) \geq \hat{\gamma} \quad \forall i$$

$$\because \frac{|b_+ - b|}{\|w\|} = \frac{|b - b_-|}{\|w\|} = \gamma \quad \& \text{ in fig-(i) } b_+ > b > b_-$$

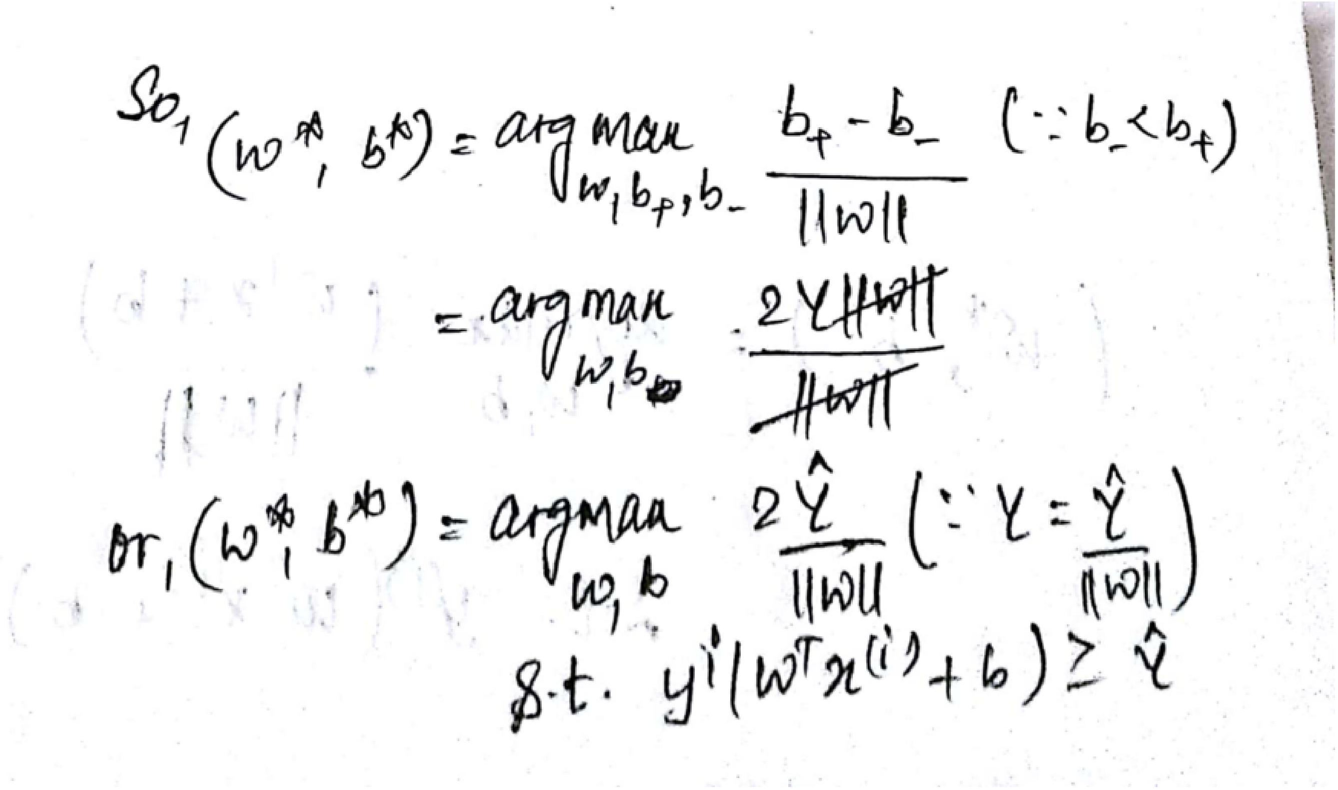
$$\therefore \quad b_+ - b = b - b_- \text{ (or) } b_+ + b_- = 2b \quad \text{— (1)}$$

$$\& \quad b_+ - b = \gamma\|w\| \text{ (or) } b_+ = \gamma\|w\| + b \quad \text{—(2)}$$

$$\text{(1) \& (2) give, } b_+ = \gamma\|w\| + b$$

$$\& \; b_- = b - \gamma\|w\|$$

Reformulated optimization objective (i)



$$\text{So, } (w^*, b^*) = \underset{w, b_+, b_-}{\arg\max} \frac{b_+ - b_-}{\|w\|} \quad (\because b_- < b_+)$$

$$= \underset{w, b}{\arg\max} \frac{2\gamma\cancel{\|w\|}}{\cancel{\|w\|}}$$

$$\text{or, } (w^*, b^*) = \underset{w, b}{\arg\max} \frac{2\hat{\gamma}}{\|w\|} \quad \left(\because \gamma = \frac{\hat{\gamma}}{\|w\|}\right)$$

$$\text{s.t. } y^i(w^T x^{(i)} + b) \geq \hat{\gamma}$$

Reformulated optimization objective (ii)

The simplification till now has been done only in terms of writing smaller notations and smaller expressions. We have not cut down anything in terms of the computations to be done in comparison to where we had started (actually, it would be nice to verify this on your own).

> *The idea that will be executed as the next step in developing the final form of the optimization problem of the SVM, does not appear to be as straightforward to ideate.*

We have been doing vector geometry since the beginning of this article. It turns out that SVM is actually geometrically motivated algorithm at its core. What is done to further simplify the objective function is not difficult apply or understand, but out of many operations that you could possibly do on an expression, zeroing down on the following idea feels non-trivial.

> *Here it is …*

One of the things about equation of a hyperplane is that scaling it does not change the hyperplane. It gives a new equation *(**w'**, b')*, where ***w'*** = *k* . ***w*** *and b'* = *k* . *b* (*k* is the scaling factor) but the hyperplane remains the same in space and the geometric margin therefore, also does not change (but functional margin does change). Leveraging this factor, what we can do is we can scale the ***w*** and *b* in our optimization objective function ***such that functional margin becomes 1.***

This makes our optimization problem as:

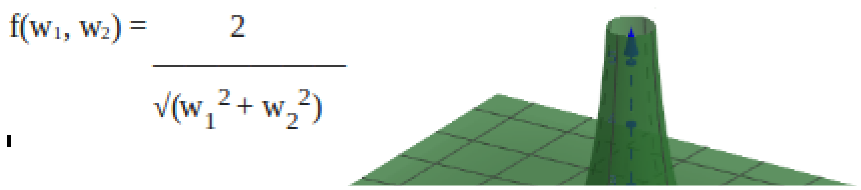
$$(w^*, b^*) = \arg\max_{w,b} \frac{2}{\|w\|}$$
$$\text{s.t. } y^{(i)}(w^T x^{(i)} + b) \geq 1$$

w and b have been scaled such that functional margin has become 1

This, in true sense, has reduced the optimization problem by reducing the number of computations to be performed!

> *We have gone through almost the entire derivation, with the last part we just saw being the most crucial one.*

However, there is a problem with this formulation. The objective function is a non-convex function, which is preferred to be avoided (fig. 4).



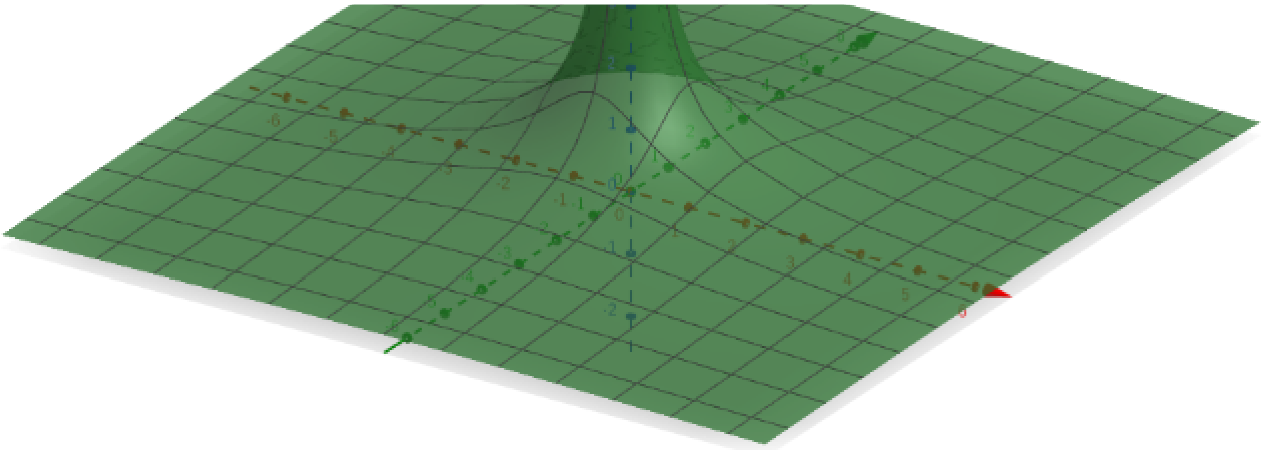


Fig 4. Geometic margin of hyperplane w = (w1, w2), b = 1; x(i) = (1, 1) (Plotter used: https://www.geogebra.org/)

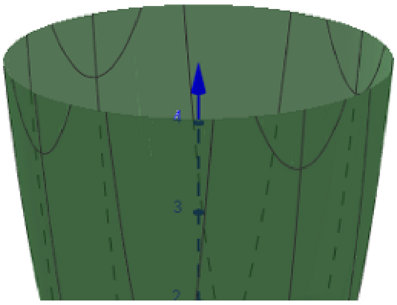
Since we apply gradient descent to optimize the function, non-convex functions may cause the algorithm to get stuck at a local minimum. But it turns out that squaring the *norm(w)* in the multiplicative inverse of the objective function gives a convex, differentiable function (inverse of the objective function is also convex, but is non-differentiable at $\boldsymbol{w} = \boldsymbol{0}$, you can verify this by using an online plotting tool). We can use this as our objective function, with the optimization problem being a little modified now:

$$(w^*, b^*) = \underset{w,b}{\operatorname{argmin}} \ \frac{\|w\|^2}{2}$$

$$\text{s.t. } y^{(i)}(w^T x^{(i)} + b) \geq 1 \quad \forall i$$

Making the objective function convex; this is also the final optimization problem of SVM

In fact, this is the final objective function which is perfectly convex also (fig. 5)! We love convex optimization problems.



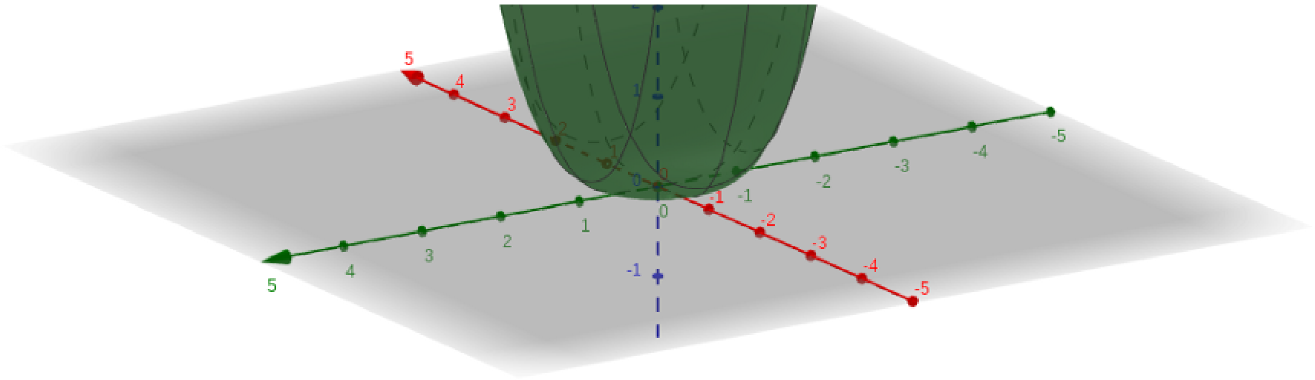


Fig. 5. The final objective function is perfectly convex (Plotter used: https://www.geogebra.org)

Note that maximizing a function *f* is same as minimizing a function $g = 1/f$. That is why the optimization problem is now a minimization problem opposed to being a maximization problem as before.

> *The end result feels satisfying.*

A few more points to conclude:

1. We started with the problem of optimizing geometric margin, and not functional margin. This is because functional margin can be easily increased (or decreased) by simply scaling $\boldsymbol{w}$ and b. Recall that doing this does not change the hyperplane. So, this either would not converge to an optimal separating hyperplane or there would be too many iterations which essentially would be useless. It might be argued that constraining the magnitude of $\boldsymbol{w}$ may help the issue. But this is not at all a good idea, since $||\boldsymbol{w}|| = c$ (c is some constant) is a non-convex constraint (as the feasible set is the surface of a hypersphere), making the optimization problem non-convex. We always want to avoid non-convex optimization problems.

2. We started with the second interpretation, and then established that both interpretations coexist. But we inferred the first interpretation from the second one. It turns out that starting out with the first interpretation might not be as convenient because then, we would be starting with two parallel hyperplanes, initially separated by some $|b1\text{-}\ b2|/||\boldsymbol{w}||$ where b1 and b2 are initial constants of their respective equations. It is not difficult to see that the distance between the two hyperplanes may not change at all even if their $\boldsymbol{w}$ is changed (rotating the hyperplanes by same angle). Constraining the feasibility region would be important for this case as well because learning hyperplanes at max distance may simply mean learning those hyperplanes that are as far as possible. That does not

serve the purpose. The constrained optimization problem would therefore be something like: “maximize the distance s.t. all -ve points are on one side of -ve hyperplane, and all +ve points are on one side of +ve hyperplane”. We could use functional margin for this, but then we can observe that all +ve points are on the same side of both +ve as well as -ve hyperplanes, the same holds for -ve points as well. So using two hyperplanes is a bit redundant. Also, without the definition of functional and geometric margin (which are defined in reference to a separating hyperplane) it would not be as easy, if not impossible, to derive the results we just did.

Footnotes

Hopefully I have written something sensible with correct Maths. I used Andrew Ng’s lecture notes and his lecture series available on YouTube (both from Stanford offline class notes, not Coursera), some ideas from a paid course that I am doing and a number of visits to stackoverflow.com, and then compiled them all into this. Hope you liked it.

Also, kindly pardon my handwriting. If what is written in those images are not clear, do let me know. I shall replace them with new pictures.

Thank you for reading.

Update

I have also written an article deriving the Soft-margin SVM. You can find it here. Thank you.

Thanks to Wendy Wong.

Machine Learning Classifcation Models Svm Supervised Learning Cost Function