Notes on SVM

Scribe: Jiajia Xie April 10, 2023

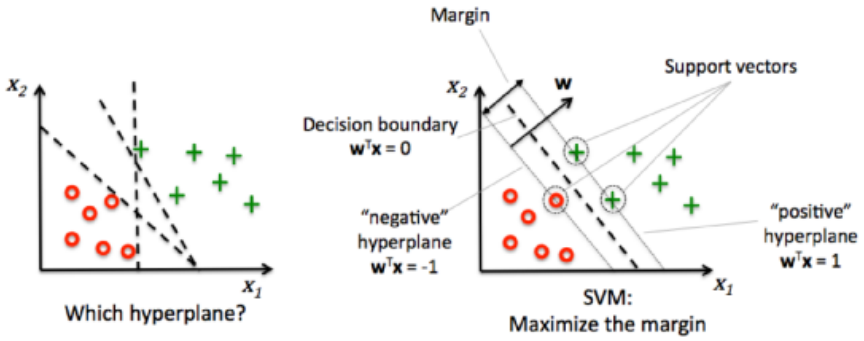

Figure 1: An Illustrative Example of Maximization of Margin

1 What is the Margin of a Linear Classifier

In machine learning, a typical type of problem is to build a binary classifier f, given $(x, y)_{i=1}^{N}$, where each $x_i \in \mathbb{R}^d$ is a d-dimensional vector and each $y_i \in \{1, -1\}$ is a binary scalar. If the classifier f is restricted to be linear, i.e. let $w \in \mathbb{R}^d$ and $b \in \mathbb{R}$ so that $f(x) = w^\top x + b$. The classifier will predict 1 if $f(x) > b$ and -1 otherwise. Note that we can always add an additional dimension with a value -1 to x, and let $w = (w, b)^\top$, this will make the classifier to be $f(x) = w^\top x$. The decision rule now becomes $f(x) > 0$ for 1 and $f(x) \leq 0$ for -1.

Note that we can always assume $\|w\| = 1$ and $\|x\| \leq 1$ without loss of generality. This can be done by re-scaling the vectors x_i. The purpose of these assumptions is to simplify our analysis.

Let us begin with some geometry analysis. What is $w^\top x$? It is a hyperplane of $d-1$. The left figure of Figure 1 shows three examples when $d = 2$, so each $w^\top x$ is just a line splitting the whole space into two such that all the data with a 1 label are on the right-hand side, i.e. $w^\top x > 0$, and all the data with a -1 label are on the left-hand side, i.e. $w^\top x \leq 0$.

What is the margin for each x_i? It is just $w^\top x$.

Theorem. *w is a vector orthogonal to the decision boundary $f(x) = 0$.*

Let us prove this theorem.

Proof. Consider any two points x_1 and x_2 has been determined by f lying on the decision boundary, i.e. $w^\top x_1 = 0$ and $w^\top x_2 = 0$. We can immediately show $w^\top(x_1 - x_2) = 0$, and we know when $d = 2$, $x_1 - x_2$ is the vector parallel to the decision boundary (this is called the triangular law of vector addition learned in high school). For a general d, $x_1 - x_2$ is just a vector on the decision boundary hyperplane. We can pick arbitrary x_1 and x_2 to make $x_1 - x_2$ any vector on the decision boundary. Therefore, w is orthogonal to the decision boundary. □

For an x_i, what is the margin? The margin of x_i is the smallest distance from all x_i to the decision boundary. In general, the distance for any x_i with $y_i = 1$ is $\frac{w^\top x}{w^\top w}$ (remember the dot product between a vector x and a unit vector y is the distance of x projected onto the direction of y), and since we assume $\|w\| = 1$, the distance is just $w^\top x$. The margin γ is defined as the lower bound of the distances:

$$\gamma := \min_{i \in \{1,2 \ldots N\}} y_i w^\top x_i \tag{1}$$

where the product between y_i and $w^\top x_i$ ensures that the distance is non-negative (remember $w^\top x_i \leq 0$ when $y_i = -1$). By defining γ, let us look again at the left figure of Figure 1 one more time. Don't the three lines all have very small γ, i.e. the smallest distance between data points and the decision boundary is close to zero?

2 Why Maximizing the Margin?

Like the three examples above, all linear classifiers with small γ are bad classifiers in terms of error bounds. In other words, a linear classifier with small γ will make a lot of wrong predictions on the unseen data $(x_i, y_i)_{i=1}^M$. Consider the following online algorithm, known as the perceptron algorithm:

Algorithm 1 Perceptron

```
procedure PERCEPTRON(w = 0)
    for A new x_i arrives do
        Make a prediction ŷ_i based on the sign of w^⊤ x_i
        if ŷ_i ≠ y_i then
            w = w + y_i x
        end if
    end for
end procedure
```

Theorem. *The maximum mistakes of a perceptron algorithm are bounded by $\frac{1}{\gamma^2}$. In other words, let C be the number of mistakes the perceptron algorithm can make, $\hat{w}$ and $\hat{x}$ achieve the margin, $\gamma = |\hat{w}^\top \hat{x}|$:*

$$C \leq \frac{\|\hat{w}\|\|\hat{x}\|}{\gamma^2} \tag{2}$$

with $\|\hat{w}\| = 1$ *and* $\|\hat{x}\| \leq 1$*, the bound is:*

$$C \leq \frac{1}{\gamma^2} \tag{3}$$

We omit the proof for the readers. The point of presenting the above theorem is: if γ is small, C the maximum number of mistakes can be large. This gives, perhaps, the most straightforward motivation for us to maximize γ, which is exactly the motivation for using a support vector machine (SVM) model: we would like to bound the mistake of the model on the unseen datasets.

3 SVM

We drop the assumptions of $\|w\| = 1$ and $\|x\| \leq 1$ and use $f(x) = w^\top x + b$ again. Let us derive the formulation of the support vector machine model step by step. As said, the objective of SVM is to select w and b such that γ is maximized. This idea is equivalent to the following mathematical formulation:

$$\begin{aligned} \max_{w,b} \quad & \gamma \\ \text{s.t.} \quad & \frac{y_i(w^\top x_i + b)}{\|w\|} \geq \gamma, \ \forall i \in \{1, 2...N\} \end{aligned}$$

Since $\frac{1}{\|w\|^2}$ is independent on $i \in \{1, 2...N\}$, we can just factor it out to the objective, so the formulation becomes:

$$\begin{aligned} \max_{w,b} \quad & \frac{\gamma}{\|w\|} \\ \text{s.t.} \quad & y_i(w^\top x_i + b) \geq \gamma, \ \forall i \in \{1, 2...N\} \end{aligned}$$

There are tricks that can enable a more simplified formulation of the above problem. One of them is: we can always re-scale kw and kb by a positive factor k and for any $i \in \{1, 2...N\}$, the distance is unchanged.

Proof.

$$\frac{y_i(kw^\top x_i + kb)}{\|kw\|} = \frac{ky_i(w^\top x_i + b)}{|k|\|w\|} = \frac{y_i(w^\top x_i + b)}{\|w\|}$$

□

The above re-scaling enables us to set $\gamma = y_i(w^\top x_i + b) = 1$ for the point with the smallest distance, i.e. the margin is equal to 1 with all distance of the data to the decision boundary remaining the same. The formulation becomes:

$$\begin{aligned} \max_{w,b} \quad & \frac{1}{\|w\|} \\ \text{s.t.} \quad & y_i(w^\top x_i + b) \geq 1, \ \forall i \in \{1, 2...N\} \end{aligned}$$

Another trick that enables us to transform the above problem into quadratic programming is changing the objective into $\|w\|^2$ and flipping the problem into a minimization problem. This is because $\max \frac{1}{\|w\|}$ is equivalent to $\min \|w\|^2$. The final problem for the SVM model is:

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2}\|w\|^2 \\ \text{s.t.} \quad & y_i(w^\top x_i + b) \geq 1, \ \forall i \in \{1, 2...N\} \end{aligned}$$

This problem belongs to the category of quadratic programming. It has a $d+1$ number of variables and N constraints. Before moving to the next section, one should note that you can always solve Problem 3 using any solver, e.g. gurobi. The challenge is, if either N or d is large, solving the problem can be computationally challenging.
The next step is to derive the dual problem of SVM, which enjoys two advantages: (1) the dual formulation enables us to apply nonlinear kernel functions, (2) the dual problem can be solved efficiently using SMO algorithm [1].

4 Dual Formulation

4.1 Does dual equal primal?

In order the derive the dual formulation, let us first form the Lagrange dual problem. Let us define the Lagrangian function as:

$$\mathcal{L}(w, b, \lambda) = \frac{1}{2}\|w\|^2 + \sum_{i=1}^{N} \lambda_i \{1 - y_i(w^\top x_i + b)\} \tag{4}$$

All of the following analyses assume the reader's familiarity with duality and K.K.T. conditions. The primal problem is the same as:

$$\min_{w,b} \max_{\lambda, \lambda \geq 0} \mathcal{L}(w, b, \lambda)$$

The dual problem is:

$$\max_{\lambda, \lambda \geq 0} \min_{w,b} \mathcal{L}(w, b, \lambda)$$

The following inequality, which is known as the weak duality theorem, must hold:

$$\max_{\lambda, \lambda \geq 0} \min_{w,b} \mathcal{L}(w, b, \lambda) \leq \min_{w,b} \max_{\lambda, \lambda \geq 0} \mathcal{L}(w, b, \lambda)$$

Proof. First, we know:

$$\mathcal{L}(w, b, \lambda) \leq \max_{\lambda} \mathcal{L}(w, b, \lambda)$$

Therefore, $\min_{w,b} \mathcal{L}(w, b, \lambda)$ is just a special case of the above inequality, so:

$$\min_{w,b} \mathcal{L}(w, b, \lambda) \leq \max_{\lambda} \min_{w,b} \mathcal{L}(w, b, \lambda)$$

And $\max_\lambda \min_{w,b} \mathcal{L}(w, b, \lambda)$ is just another special case for the left-hand side, so:

$$\max_{\lambda} \min_{w,b} \mathcal{L}(w, b, \lambda) \leq \min_{w,b} \max_{\lambda} \mathcal{L}(w, b, \lambda)$$

□

This inequality says that the dual problem yields a lower bound for the primal problem. However, for the SVM problem, we are interested in, the inequality is in fact equality, i.e.

$$\max_{\lambda, \lambda \geq 0} \min_{w,b} \mathcal{L}(w, b, \lambda) = \min_{w,b} \max_{\lambda, \lambda \geq 0} \mathcal{L}(w, b, \lambda)$$

which literally tells us that solving the dual problem is equivalent to solving the primal problem. We omit the corresponding theorem, Slaster's condition, and its proof for simplicity.

4.2 What is the dual?

We now derive the dual formulation for the SVM model. The dual problem by definition is:

$$\max_{\lambda, \lambda \geq 0} \min_{w,b} \mathcal{L}(w, b, \lambda) \tag{5}$$

To solve the internal minimization problem, we can compute the gradient with respect to w and b and set them equal to zero. The two conditions attained are:

$$w = \sum_{i=1}^{N} \lambda_i y_i x_i$$

$$b = \sum_{i=1}^{N} \lambda_i y_i$$

Eliminating w and b by substitutions from Equation (4) gives us the dual problem to solve in the end:

$$\begin{aligned} \max_{\lambda} \quad & \sum_{i=1}^{N} \lambda_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \lambda_i \lambda_j y_i y_j x_i^\top x_j \\ \text{s.t. } & \lambda_i \geq 0, \ \forall i \in \{1, 2...N\} \\ & \sum_{i=1}^{N} \lambda_i y_i = 0 \end{aligned} \tag{6}$$

The dual formulation enables us to define arbitrary kernel function and replace $x_i^\top x_j$ by $k(x_i, x_j)$. The above problem remains to be quadratic programming with linear constraints. The number of variables is N and the number of constraints is $N + 1$. Let and $X \in \mathbb{R}^{N \times d}$ and $y \in \mathbb{R}^N$ be the N data's matrix representations, and let $Q = diag(y) X X^\top diag(y)$ where $diag(y)$ is a diagonal matrix with y as its diagonal elements. Also let A be the negative of an identity matrix, $A = -I$:

The above problem is the same as:

$$\begin{aligned} \max_{\lambda} \quad & \mathbf{1}^\top \lambda - \frac{1}{2}\lambda^\top Q \lambda \\ \text{s.t.} \quad & A\lambda \leq \mathbf{0} \\ & \lambda^\top y = 0 \end{aligned} \tag{7}$$

5 Numerical Experiment with Quadratic Program Solvers

We can try to solve the above problem using any quadratic program solvers, e.g. *gurobi*, *cvxopt*. See http://cvxopt.org/index.html for the descriptions and examples of *cvxopt*. In general, using these solvers for training an SVM model with Equation (7) is neither sufficient nor robust. For $N = 20010$, the solver takes more than 40 mins to complete. Considering the robustness, this formulation has no solution when the data points are not linearly-separable (i.e. there exists no w and b to split all $y_i = -1$ and $y_i = 1$ into two separated half-planes). Even more, Equation (7) is not robust with linearly separable cases but there exists a few data points that make the margin very small. This challenge will be overcome if we use a soft-margin SVM instead. We will also cover soft-margin SVM and a fast algorithm for training SVMs, called SMO [1], in the future.

References

[1] J. Platt, "Sequential minimal optimization: A fast algorithm for training support vector machines," 1998.