Machine Learning Methods
Support Vector Machine

By Zhu Xuelin
Email: xuelin@u.nus.edu

September 28, 2024

This note focus on the classification method - *Support Vector Machine*. For more reference, check §4.5, 12 of *Elements of Statistical Learning*, §7 of *Pattern Recognition and Machine Learning*.

We focus on the derivation of binary case now, $\mathcal{Y} = \{-1, 1\}$ with $\mathbf{x} \in \mathbb{R}^p$. The key idea of SVM is to find the optimal separating hyperplane, which maximizes the distance (also called *margin*) to the closest point from either class. And the two notions, optimal and hyperplane, are worth further explanation.

Definition 1 (Hyperplane)**.** A separating (linear) hyperplane $\mathcal{L}$ in $\mathbb{R}^p$ is a collection of points in the following form

$$\mathcal{L} = \{\mathbf{x} \in \mathbb{R} : \beta_0 + \boldsymbol{\beta}^\top \mathbf{x} = 0\}, \quad \text{where } \beta_0 \in \mathbb{R} \text{ and } \boldsymbol{\beta} \in \mathbb{R}^p.$$

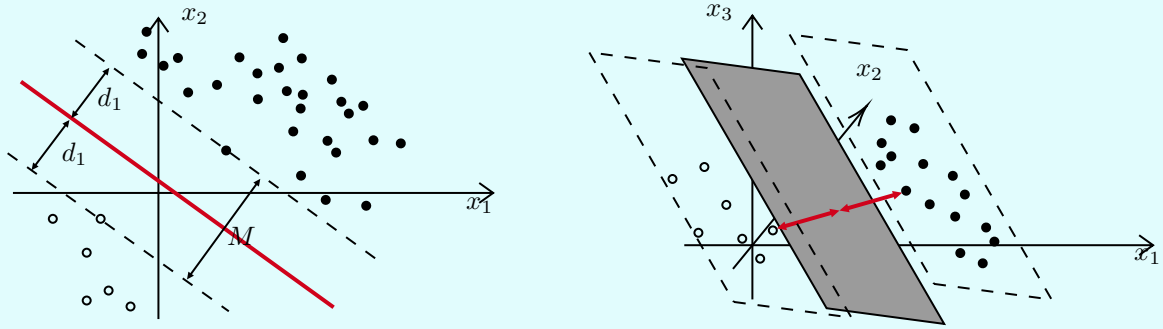
In mathematics. it is also referred as the *affine* hyperplane.

Definition 2 (Margin)**.** For a hyperplane $\mathcal{L}$, the its margin M is defined to be the sum of smallest distances to the closest point from either class, i.e.

$$M = d(\mathbf{x}_1, \mathcal{L}) + d(\mathbf{x}_2, \mathcal{L}),$$

where $\mathbf{x}_1, \mathbf{x}_2$ are two closest points from two classes.

Example. Here are two examples of given hyperplanes, sample, and margins in $\mathbb{R}^2$ and $\mathbb{R}^3$.



$\mathcal{L} = \{x : 1 + x_1 + x_2 = 0\}$ $\quad$ $\mathcal{L} = \{x : 1 + x_1 + x_2 + x_3 = 0\}$

Figure 1: Examples of hyperplane and margin

Remark. There are several things worth noticing in the examples.

- A hyperplane of $\mathbb{R}^2$ is a line, and of $\mathbb{R}^3$ is a plane. But it is usually not a subspace, since it does not contain the origin.
- Given $\boldsymbol{\beta} \in \mathbb{R}^p$, we can translate parallel the hyperplane by varying the intercept $\beta_0 \in \mathbb{R}$. But we choose the one that makes the distance to two classes the same as in the $\mathbb{R}^2$ example, i.e. $d(\mathbf{x}_1, \mathcal{L}) = d(\mathbf{x}_2, \mathcal{L})$.
- Once the hyperplane is given, denoted by $f(\mathbf{x}) = \beta_0 + \boldsymbol{\beta}^\top \mathbf{x}$ for convenience, the points on either side would have different sign when plugging in $\beta_0 + \boldsymbol{\beta}^\top \mathbf{x}$. So, the classification rule is

$$y = \begin{cases} 1 & \text{if } f(\mathbf{x}) > 0, \\ -1 & \text{if } f(\mathbf{x}) < 0. \end{cases}$$

With these preliminaries, we can introduce SVM formally now, starting with the ideal case, which means the 2 classes could be separated by a hyperplane. But note that, sometimes they cannot.

1 Separable Case SVM

1.1 Optimization Formation

For any hyperplanes $\mathcal{L}$, first we calculate what it the distance of any point to it. Suppose there are two point $\mathbf{x}_1, \mathbf{x}_2$ in it. Then by definition, we know

$$\begin{cases} \beta_0 + \boldsymbol{\beta}^\top \mathbf{x}_1 = 0 \\ \beta_0 + \boldsymbol{\beta}^\top \mathbf{x}_2 = 0 \end{cases} \quad \Rightarrow \quad \boldsymbol{\beta}^\top (\mathbf{x}_1 - \mathbf{x}_2) = 0 \quad \Rightarrow \quad \boldsymbol{\beta} \perp \mathcal{L},$$

i.e. $\boldsymbol{\beta} \in \mathbb{R}^p$ (not including β_0) is a vector in $\mathbb{R}^p$ perpendicular to the hyperplane $\mathcal{L}$, as shown below.

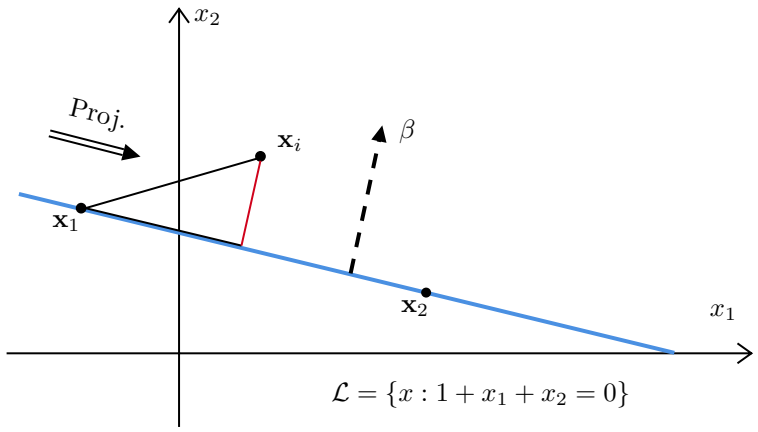


Figure 2: Illustration in a $\mathbb{R}^2$ hyperplane

Then the signed distance of a sample point $\mathbf{x}_i$ to $\mathcal{L}$ should be the length of the projection of $(\mathbf{x}_i - \mathbf{x}_1)$ on $\boldsymbol{\beta}$, i.e.

$$d_{\text{sign}}(\mathbf{x}_i, \mathcal{L}) = ||\text{Proj}_{\boldsymbol{\beta}}(\mathbf{x}_i - \mathbf{x}_1)|| = \left|\left| \frac{\boldsymbol{\beta}\boldsymbol{\beta}^\top (\mathbf{x}_i - \mathbf{x}_1)}{||\boldsymbol{\beta}||^2} \right|\right| = \frac{1}{||\boldsymbol{\beta}||} (\boldsymbol{\beta}^\top \mathbf{x}_i - \boldsymbol{\beta}^\top \mathbf{x}_1).$$

With $\mathbf{x}_1$ on $\mathcal{L}$, we can use $\beta_0 + \boldsymbol{\beta}^\top \mathbf{x}_1 = 0$ to replace $\boldsymbol{\beta}^\top \mathbf{x}_1$ in the signed distance, and it follows that

$$d_{\text{sign}}(\mathbf{x}_i, \mathcal{L}) = \frac{1}{||\boldsymbol{\beta}||}(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) = \frac{1}{||\boldsymbol{\beta}||} f(\mathbf{x}_i),$$

where the sign comes from $f(\mathbf{x}_i)$. If we use y_i to denote its classification result, i.e.

$$y = \begin{cases} 1 & \text{if } f(\mathbf{x}) > 0, \\ -1 & \text{if } f(\mathbf{x}) < 0, \end{cases}$$

we can derive an expression for the distance of any point $\mathbf{x}_i \in \mathbb{R}^p$ to $\mathcal{L}$:

$$d(\mathbf{x}_i, \mathcal{L}) = \frac{f(\mathbf{x}_i) y_i}{||\boldsymbol{\beta}||}.$$

Recall the notation of margin M, and we can define the SVM formally now.

Definition 3 (Support vector machine)**.** Suppose $(\mathbf{x}_i, Y_i)$, $i = 1, \ldots, N$ is a random sample in $\mathbb{R}^p \times \{-1, 1\}$. The support vector machine is a hyperplane in $\mathbb{R}^p$ solving the following optimization problem:

$$\max_{\beta_0, \boldsymbol{\beta}} M \quad \text{subject to} \quad \frac{y_i(\mathbf{x}_i^\top \boldsymbol{\beta} + \beta_0)}{||\boldsymbol{\beta}||} \geq M, \quad \forall i = 1, \ldots, N.$$

Intuitively, this means we are looking for a hyperplane such that maximizes the margin, and it corresponds to what we want in the beginning. Since $\boldsymbol{\beta}$ is a direction vector for the hyperplane, we do not care about its length. For convenience, we further set $||\boldsymbol{\beta}|| = 1/M$, and we want

$$\max_{\beta_0, \boldsymbol{\beta}} \frac{1}{||\boldsymbol{\beta}||} \quad \Longleftrightarrow \quad \min_{\beta_0, \boldsymbol{\beta}} ||\boldsymbol{\beta}|| \quad \Longleftrightarrow \quad \min_{\beta_0, \boldsymbol{\beta}} \frac{1}{2}||\boldsymbol{\beta}||^2,$$

which leads us to the **primal problem** of SVM:

$$\min_{\beta_0, \boldsymbol{\beta}} \frac{1}{2}||\boldsymbol{\beta}||^2 \quad \text{subject to} \quad y_i(\mathbf{x}_i^\top \boldsymbol{\beta} + \beta_0) \geq 1, \quad \forall i = 1, \ldots, N.$$

Remark. Though the function to be minimized $||\boldsymbol{\beta}||$ does not contain β_0, the constraints contain β_0, which plays a role in the optimization. And this is a standard convex problem (definition by Wikipedia). This is why we can optimize it easily.

1.2 Convex Optimization

1.2.1 Primal Problem

Using Lagrange function, we can simplify our primal problem to a conventional optimization form, which helps solve it using optimization methods (for more, refer *STATS - 606*).

Problem Setup (SVM Primal problem)**.** The primal problem of SVM,

$$\min_{\beta_0, \boldsymbol{\beta}} \frac{1}{2}||\boldsymbol{\beta}||^2 \quad \text{subject to} \quad y_i(\mathbf{x}_i^\top \boldsymbol{\beta} + \beta_0) \geq 1, \quad \forall i = 1, \ldots, N,$$

introduces the primal Lagrange function

$$L_P(\beta_0, \boldsymbol{\beta}, \boldsymbol{\alpha}) := \frac{1}{2}||\boldsymbol{\beta}||^2 - \sum_{i=1}^{N} \alpha_i \Big[y_i \left(\mathbf{x}_i^\top \boldsymbol{\beta} + \beta_0 \right) - 1 \Big],$$

where $\boldsymbol{\alpha} = \begin{pmatrix} \alpha_1 & \cdots & \alpha_N \end{pmatrix}^\top$ with all non-negative elements. Then solving the primal problem is equivalent to solve the following problem:

$$\min_{\beta_0, \boldsymbol{\beta}, \boldsymbol{\alpha}} L_P(\beta_0, \boldsymbol{\beta}, \boldsymbol{\alpha}) \quad \text{subject to} \quad \alpha_i \geq 0, \quad \forall i = 1, \ldots, N.$$

The reason to do the above way is a focus of §1.2.3 of this note, thereby omitted here. Now, we continue further simplify this primal problem to the dual problem. Taking derivative of L_P w.r.t. β_0 and $\boldsymbol{\beta}$ leads to

$$\frac{\partial L_P}{\partial \beta_0} = -\sum_{i=1}^{N} \alpha_i y_i \quad \Rightarrow \quad \sum_{i=1}^{N} \alpha_i^* y_i = 0, \tag{1}$$

$$\frac{\partial L_P}{\partial \boldsymbol{\beta}} = \boldsymbol{\beta} - \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i \quad \Rightarrow \quad \boldsymbol{\beta}^* = \sum_{i=1}^{N} \alpha_i^* y_i \mathbf{x}_i. \tag{2}$$

Plugging the two expression back into the L_P, leads to

$$\begin{aligned} L_P(\beta_0, \boldsymbol{\beta}, \boldsymbol{\alpha}) &= \frac{1}{2} \left|\left| \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i \right|\right|^2 - \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i^\top \left[\sum_{j=1}^{N} \alpha_j y_j \mathbf{x}_j \right] - \sum_{i=1}^{N} \alpha_i y_i \beta_0 + \sum_{i=1}^{N} \alpha_i \\ &= \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j \quad \text{subject to} \quad \alpha_i \geq 0 \ \text{ and } \ \sum_{i=1}^{N} \alpha_i y_i = 0. \end{aligned}$$

1.2.2 Dual Problem

And this is what we will handle in the end, the dual problem.

Problem Setup (SVM Dual problem)**.** The primal problem of SVM,

$$\min_{\beta_0, \boldsymbol{\beta}, \boldsymbol{\alpha}} L_P(\beta_0, \boldsymbol{\beta}, \boldsymbol{\alpha}) \quad \text{subject to} \quad \alpha_i \geq 0, \quad \forall i = 1, \ldots, N,$$

leads to the dual Lagrange function

$$L_D(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j \quad \text{subject to} \quad \alpha_i \geq 0 \ \text{ and } \ \sum_{i=1}^{N} \alpha_i y_i = 0.$$

Then solving the solving the primal problem is equivalent to solve

$$\max_{\boldsymbol{\alpha}} L_D(\boldsymbol{\alpha}).$$

We already can solve for the best hyperplane $\hat{f}(\mathbf{x})$ by the above, but we also want to know the reason of switch max and min, and better interpret the results. These lead us to the following discussion.

1.2.3 Constrained Optimization

In this subsection, we forget the notation of β_0 in the SVM, and suppose all parameters are contained in $\boldsymbol{\beta} \in \mathbb{R}^p$, which is a more general case. We consider a primal problem

$$p^* = \min_{\boldsymbol{\beta} \in \mathbb{R}^p} f(\boldsymbol{\beta}) \quad \text{subject to} \quad \mathbf{g}(\boldsymbol{\beta}) = \begin{cases} g_1(\boldsymbol{\beta}) \leq 0 \\ \vdots \\ g_m(\boldsymbol{\beta}) \leq 0 \end{cases} \tag{3}$$

i.e., minimizing the scalar $f(\boldsymbol{\beta})$ w.r.t. a p-dimensional vector subject to m inequality constraints. If we introduce the Lagrange function and define

$$L_P(\boldsymbol{\beta}, \boldsymbol{\lambda}) = f(\boldsymbol{\beta}) + \boldsymbol{\lambda}^\top \mathbf{g}(\boldsymbol{\beta}),$$

where $\boldsymbol{\lambda} = \begin{pmatrix} \lambda_1 & \cdots & \lambda_m \end{pmatrix}^\top$ with all non-negative elements, we can transfer the primal problem to be

$$p^* = \min_{\boldsymbol{\beta}: \mathbf{g}(\boldsymbol{\beta}) \leq 0} f(\boldsymbol{\beta}) = \min_{\boldsymbol{\beta} \in \mathbb{R}^p} \left[f(\boldsymbol{\beta}) + \max_{\boldsymbol{\lambda} \geq 0} \boldsymbol{\lambda}^\top \mathbf{g}(\boldsymbol{\beta}) \right] = \min_{\boldsymbol{\beta} \in \mathbb{R}^p} \max_{\boldsymbol{\lambda} \geq 0} \left[f(\boldsymbol{\beta}) + \boldsymbol{\lambda}^\top \mathbf{g}(\boldsymbol{\beta}) \right] = \min_{\boldsymbol{\beta} \in \mathbb{R}^p} \max_{\boldsymbol{\lambda} \geq 0} L_P(\boldsymbol{\beta}, \boldsymbol{\lambda}).$$

Remark. We can simply explain the reason for the second equality. Suppose some $g_i(\boldsymbol{\beta}) > 0$. Then this $\boldsymbol{\beta}$ could not be a optimizer in the LHS. But since $\lambda_i > 0$, in the RHS, the maximization of $\lambda_i g_i(\boldsymbol{\beta})$ can go to ∞. So, it is also not included in any possible solution of RHS.

As we shall see, under some regularity conditions, we can switch the order of min and max:

$$p^* = \min_{\boldsymbol{\beta} \in \mathbb{R}^p} \max_{\boldsymbol{\lambda} \geq 0} L_P(\boldsymbol{\beta}, \boldsymbol{\lambda}) = \max_{\boldsymbol{\lambda} \geq 0} \min_{\boldsymbol{\beta} \in \mathbb{R}^p} L_P(\boldsymbol{\beta}, \boldsymbol{\lambda}).$$

If we define

$$L_D(\boldsymbol{\lambda}) = \min_{\boldsymbol{\beta} \in \mathbb{R}^p} L_P(\boldsymbol{\beta}, \boldsymbol{\lambda}),$$

under the regularity conditions, the primal problem is equivalent to the dual problem:

$$d^* = \max_{\boldsymbol{\lambda} \geq 0} L_D(\boldsymbol{\lambda}) = \max_{\boldsymbol{\lambda} \geq 0} \min_{\boldsymbol{\beta} \in \mathbb{R}^p} L_P(\boldsymbol{\beta}, \boldsymbol{\lambda}) = p^*.$$

It remains to consider the order of optimizations. In general, $p^* \geq d^*$, and we call $(p^* - d^*)$, the *duality gap*. For simplicity, in the following discussion, we write $\mathbf{g}(\boldsymbol{\beta}) < 0$ or $\boldsymbol{\lambda} \geq 0$ to mean the inequalities for all elements in this vector.

Theorem 4. *In the above setting and notations, if $f(\cdot)$ and $\mathbf{g}(\cdot)$ are both convex, and there exists $\boldsymbol{\beta} \in \mathbb{R}^p$ such that $\mathbf{g}(\boldsymbol{\beta}) < 0$, then $p^* = d^*$.*

The above theorem guarantees the order switch of max and min in our SVM problem. Furthermore, we have a stronger theorem to help the optimization.

Theorem 5 (Karush-Kuhn-Tucker condition)**.** *If f and g are both differentiable and convex, and there exists $\boldsymbol{\beta} \in \mathbb{R}^p$ such that $\mathbf{g}(\boldsymbol{\beta}) < 0$, then the solution to the optimization $(\boldsymbol{\beta}^*, \boldsymbol{\lambda}^*)$ satisfies*

(a) $\mathbf{g}(\boldsymbol{\beta}^*) \leq 0$ *and* $\boldsymbol{\lambda}^* \geq 0$,

(b) for all $j = 1, \ldots, m$, *we have* $\lambda_j^* g_j(\boldsymbol{\beta}^*) = 0$,

(c) $\nabla_{\boldsymbol{\beta}} f(\boldsymbol{\beta}^*) + (\boldsymbol{\lambda}^*)^\top \nabla_{\boldsymbol{\beta}} \mathbf{g}(\boldsymbol{\beta}^*) = \mathbf{0}$, *in other words,* $\nabla_{\boldsymbol{\beta}} L_P(\boldsymbol{\beta}^*, \boldsymbol{\lambda}^*) = \mathbf{0}$.

1.2.4 Solution Properties to SVM

Now in this section, we focus on the properties of the solution. Recall the SVM primal problem:

$$\min_{\beta_0, \boldsymbol{\beta}} \frac{1}{2}\|\boldsymbol{\beta}\|^2 \quad \text{subject to} \quad y_i\left(\mathbf{x}_i^\top \boldsymbol{\beta} + \beta_0\right) \geq 1, \quad \forall i = 1, \ldots, N,$$

and comparing to (3), we can find

$$f(\beta_0, \boldsymbol{\beta}) = \frac{1}{2}||\boldsymbol{\beta}||^2 \text{ and } g_i(\beta_0, \boldsymbol{\beta}) = 1 - y_i\left(\mathbf{x}_i^\top \boldsymbol{\beta} + \beta_0\right) \leq 0.$$

Both f and $\mathbf{g}$ are convex and differentiable, satisfying the KKT condition. Therefore, we can apply the solution properties to the SVM problem. By Theorem 5,

(a) for all $i = 1, \ldots, N$, we have $y_i\left(\mathbf{x}_i^\top \boldsymbol{\beta}^* + \beta_0^*\right) \geq 1$ and $\alpha_i^* \geq 0$,

(b) for all $i = 1, \ldots, N$, we have $\alpha_i^*\left\{y_i\left[(\boldsymbol{\beta}^*)^\top \mathbf{x}_i + \beta_0^*\right] - 1\right\} = 0$,

(c) recalling the derivatives (1) and (2), we have

$$\sum_{i=1}^N \alpha_i^* y_i = 0 \text{ and } \boldsymbol{\beta}^* = \sum_{i=1}^N \alpha_i^* y_i \mathbf{x}_i.$$

Now we can have some discussions concerning these results. Fix the observation i. From (b), if $\alpha_i^* > 0$, then

$$y_i\left[(\boldsymbol{\beta}^*)^\top \mathbf{x}_i + \beta_0^*\right] = 1$$

And recall that the original optimization is

$$\min_{\beta_0, \boldsymbol{\beta}} \frac{1}{2}||\boldsymbol{\beta}||^2 \quad \text{subject to} \quad y_i(\mathbf{x}_i^\top \boldsymbol{\beta} + \beta_0) \geq 1, \quad \forall i = 1, \ldots, N,$$

with the equality holding when $\mathbf{x}_i$ is on the boundary of classification.

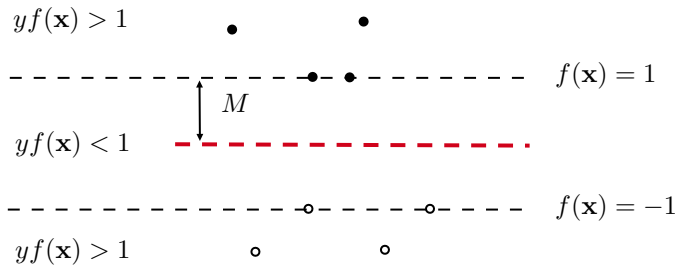


Figure 3: SVM Boundary

Similarly, by (c), if $y_i\left[(\boldsymbol{\beta}^*)^\top \mathbf{x}_i + \beta_0^*\right] > 1$, then $\mathbf{x}_i$ is not on the boundary and at the same time $\alpha_i = 0$. This means whether an observation $\mathbf{x}_i$ is on the boundary or not can be judged by its corresponding α_i^*, i.e. for all observations $i = 1, \ldots, N$,

$$\alpha_i = \begin{cases} 0 & \text{if } \mathbf{x}_i \text{ is not on the boundary,} \\ 1 & \text{if } \mathbf{x}_i \text{ is on the boundary.} \end{cases}$$

Then combining the (c), it follows that

$$\boldsymbol{\beta}^* = \sum_{i=1}^{N} \alpha_i^* y_i \mathbf{x}_i = \sum_{i:\alpha_i^*>0} \alpha_i^* y_i \mathbf{x}_i = \sum_{i \in \text{Boundary}} \alpha_i^* y_i \mathbf{x}_i.$$

Formally speaking, SVM classification decision is decided by a linear combination of observations on the boundary. And we call those points on the boundary, the *support points/vectors*, denoted by a set $\mathbb{S}$.

1.2.5 Estimation of the Intercept

We can understand the hyperplane $f(\mathbf{x}) = \beta_0 + \boldsymbol{\beta}^\top \mathbf{x}$ as a direction vector $\boldsymbol{\beta} \in \mathbb{R}$ and a location parameter β_0. By previous sections, we have estimated the direction $\boldsymbol{\beta}^*$, and now it remains the location. Since

$$\forall i \in \mathbb{S}: \quad y_i\left(\beta_0 + \boldsymbol{\beta}^\top \mathbf{x}_i\right) = 1 \;\Rightarrow\; y_i^2\left(\beta_0 + \boldsymbol{\beta}^\top \mathbf{x}_i\right) = y_i \;\Rightarrow\; \sum_{i\in\mathbb{S}}\left(\beta_0 + \boldsymbol{\beta}^\top \mathbf{x}_i\right) = \sum_{i\in\mathbb{S}} y_i,$$

which leads to a natural estimator of β_0^* by plugging in $\boldsymbol{\beta}^*$ and the following algorithm:

$$\beta_0^* = \frac{\sum_{i\in\mathbb{S}}\left(y_i - (\boldsymbol{\beta}^*)^\top \mathbf{x}_i\right)}{|\mathbb{S}|}.$$

Algorithm 6. The procedure to do SVM is:

(a) Optimize the dual problem for $\boldsymbol{\alpha}^*$:

$$\max_{\boldsymbol{\alpha}} L_D(\boldsymbol{\alpha}) = \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j \quad \text{subject to } \alpha_i \geq 0 \text{ and } \sum_{i=1}^{N}\alpha_i y_i = 0.$$

(b) Calculate the estimated hyperplane by

$$\beta^* = \sum_{i=1}^{N}\alpha_i^* y_i \mathbf{x}_i \text{ and } \beta_0^* = \frac{\sum_{i\in\mathbb{S}}\left[y_i - (\boldsymbol{\beta}^*)^\top \mathbf{x}_i\right]}{|\mathbb{S}|}$$

(c) The final classification rule is

$$f^*(\mathbf{x}) = \beta_0^* + (\boldsymbol{\beta}^*)^\top \mathbf{x} \begin{cases} > 0 & \Rightarrow \quad \hat{y} = 1 \\ < 0 & \Rightarrow \quad \hat{y} = -1 \end{cases}$$

1.3 Kernelization of SVM

You may wonder, if we only care about the solution but not the properties or interpretations, why don't we directly solve the primal problem by minimizing

$$L_P\left(\beta_0, \boldsymbol{\beta}, \boldsymbol{\alpha}\right) := \frac{1}{2}\|\boldsymbol{\beta}\|^2 - \sum_{i=1}^{N} \alpha_i \left[y_i \left(\mathbf{x}_i^\top \boldsymbol{\beta} + \beta_0\right) - 1\right], \tag{4}$$

but rather transfer to another dual problem by maximizing

$$L_D(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j \quad \text{subject to} \quad \alpha_i \geq 0 \quad \text{and} \sum_{i=1}^{N} \alpha_i y_i = 0. \tag{5}$$

Are we making things more complicated? The answer is, if we only care about the solution, either optimization is fine. But the dual problem naturally leads us to the kernelization of SVM with $\mathbf{x}_i^\top \mathbf{x}_j$.

Specifying a kernel function $k(\cdot, \cdot) : \mathbb{R}^p \times \mathbb{R}^p \mapsto \mathbb{R}$, similarly to the kernel PCA/Ridge, we project

$$\forall i = 1, \ldots, N: \quad \mathbf{x}_i \mapsto \phi(\mathbf{x}) := k_{\mathbf{x}_i} := k(\mathbf{x}_i, \cdot), \text{ where } k_{\mathbf{x}_i}(\cdot) : \mathbb{R}^p \mapsto \mathbb{R},$$

and find a linear hyperplane on $\mathscr{H}_k$, which can be understood as a non-linear hyperplane in $\mathbb{R}^p$. So, in the primal problem, if we replace

$$\mathbf{x}_i \Rightarrow \phi(\mathbf{x}_i) \text{ and } \mathbf{x}_i^\top \mathbf{x}_j \Rightarrow \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle = k(\mathbf{x}_i, \mathbf{x}_j),$$

the primal problem (4) needs modification to

$$L_P(\beta_0, \boldsymbol{\beta}, \boldsymbol{\alpha}) := \frac{1}{2}||\boldsymbol{\beta}||^2 - \sum_{i=1}^{N} \alpha_i \Big[y_i \Big(\langle \phi(\mathbf{x}_i), \boldsymbol{\beta} \rangle + \beta_0 \Big) - 1 \Big]$$

Treating all $\phi(\mathbf{x}_i)$ as constants, by the same derivation, the dual problem (5) needs modification to

$$L_D(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) \quad \text{subject to} \quad \alpha_i \geq 0 \quad \text{and} \sum_{i=1}^{N} \alpha_i y_i = 0.$$

Similar to the Algorithm 6, once $\boldsymbol{\alpha}$ is estimated, everything else is done. But one needs to note that though

$$\boldsymbol{\beta}^* = \sum_{i=1}^{N} \alpha_i^* y_i \phi(\mathbf{x}_i), \text{ where } \phi \text{ is an element in } \mathscr{H}_k,$$

and we don't know its true form, this would not bring any trouble since the estimation of the intercept is

$$\begin{aligned}
\beta_0^* &= \frac{\sum_{i \in \mathbb{S}} \left[y_i - (\boldsymbol{\beta}^*)^\top \phi(\mathbf{x}_i) \right]}{|\mathbb{S}|} = \frac{\sum_{i \in \mathbb{S}} \Big(y_i - \langle \boldsymbol{\beta}^*, \phi(\mathbf{x}_i) \rangle \Big)}{|\mathbb{S}|} \\
&= \frac{\sum_{i \in \mathbb{S}} \Big(y_i - \Big\langle \sum_{i=1}^{N} \alpha_j^* y_j \phi(\mathbf{x}_j), \phi(\mathbf{x}_i) \Big\rangle \Big)}{|\mathbb{S}|} \\
&= \frac{\sum_{i \in \mathbb{S}} \Big(y_i - \sum_{i=1}^{N} \alpha_j^* y_j k(\mathbf{x}_i, \mathbf{x}_j) \Big)}{|\mathbb{S}|},
\end{aligned}$$

and for any new observation $\mathbf{x} \in \mathbb{R}^p$, the classification rule is

$$f(\mathbf{x}) = \beta_0^* + (\boldsymbol{\beta}^*)^\top \phi(\mathbf{x}) = \beta_0^* + \langle \boldsymbol{\beta}^*, \phi(\mathbf{x}) \rangle = \beta_0^* + \left\langle \sum_{i=1}^N \alpha_i^* y_i \phi(\mathbf{x}_i), \phi(\mathbf{x}) \right\rangle$$
$$= \beta_0^* + \sum_{i=1}^N \alpha_i^* y_i \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}) \rangle = \beta_0^* + \sum_{i=1}^N \alpha_i^* y_i k(\mathbf{x}_i, \mathbf{x}).$$

2 Non-separable Case SVM

The SVM searches for a linear hyperplane in $\mathbb{R}^p$, but it is possible that there does not exist any hyperplane that could perfectly split two classes (do not consider kernel SVM). In the algorithm, it behaves no solution to the optimization, both L_P and L_D. In this case, we search a hyperplane which maximizes the margin by allowing some points on the wrong side of the margin, as shown in Figure 4.

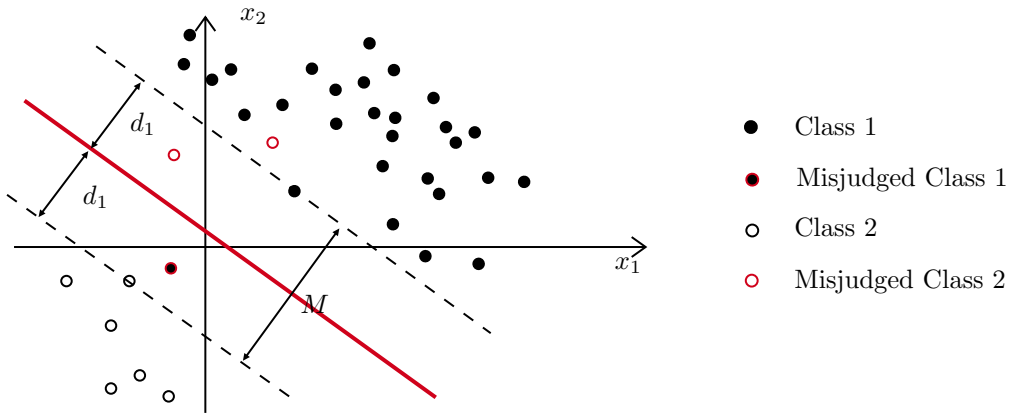


Figure 4: Non-separable SVM

In practical, we do the following primal optimization.

Definition 7 (Non-separable SVM)**.** Suppose $(\mathbf{x}_i, Y_i)$, $i = 1, \ldots, N$ is a random sample in $\mathbb{R}^p \times \{-1, 1\}$. The non-separable SVM,

$$f^*(\mathbf{x}) = \beta_0^* + (\boldsymbol{\beta}^*)^\top \mathbf{x},$$

is a hyperplane in $\mathbb{R}^p$ solving the following optimization problem:

$$\min_{\beta_0, \boldsymbol{\beta}} \frac{1}{2} ||\boldsymbol{\beta}||^2 \quad \text{subject to} \quad y_i(\mathbf{x}_i^\top \boldsymbol{\beta} + \beta_0) \geq 1 - \xi_i, \quad \forall i = 1, \ldots, N,$$

where the slack variable is defined to be

$$\boldsymbol{\xi}_{N \times 1} = \begin{pmatrix} \xi_1 \\ \vdots \\ \xi_N \end{pmatrix} \quad \text{with} \quad \forall i = 1, \ldots, N \quad \xi_i \geq 0 \quad \text{and} \quad \sum_{i=1}^N \xi_i \leq K.$$

Recall that in the separable SVM, we know the two cases

$$y_i f(\mathbf{x}_i) = y_i \left(\mathbf{x}_i^\top \boldsymbol{\beta} + \beta_0\right) \geq 1 \text{ with } \begin{cases} = 1 & \mathbf{x}_i \text{ in on the boundary,} \\ > 1 & \mathbf{x}_i \text{ in not on it and correctly classified.} \end{cases}$$

Now in the non-separable case, we allow $y_i f(\mathbf{x}_i) \geq 1 - \xi_i$, it allows three cases

$$y_i f(\mathbf{x}_i) = y_i \left(\mathbf{x}_i^\top \boldsymbol{\beta} + \beta_0\right) \begin{cases} = 1 & \mathbf{x}_i \text{ is on the boundary,} \\ > 1 & \mathbf{x}_i \text{ is not on it and correctly classified,} \\ < 1 & \mathbf{x}_i \text{ is inside the margin of its class.} \end{cases}$$

And ξ_i can be viewed as a tolerance of how much $\mathbf{x}_i$ is allowed to be wrong, and $\sum_{i=1}^N \xi_i$ is a total wrong flexibility.

2.1 Optimization and Intuition behind

By convex optimization, we can equivalently do

$$\min_{\beta_0, \boldsymbol{\beta}} \frac{1}{2}\|\boldsymbol{\beta}\|^2 + C\sum_{i=1}^N \xi_i \quad \text{subject to} \quad \xi_i \geq 0 \text{ and } y_i \left(\mathbf{x}_i^\top \boldsymbol{\beta} + \beta_0\right) \geq 1 - \xi_i, \quad \forall i = 1, \ldots, N. \tag{6}$$

And this is exactly what we do in the programming.

Remark. There are few things we can notice from this form.

(a) The constant $C \geq 0$ is a cost/tuning parameter to be chosen, which controls the trade-off between maximizing margin and minimizing error.

(b) The separable case SVM corresponds to $C = \infty$, and leading to $\xi_i = 0$.

(c) Large C tends to minimize the training error, but the margin may be narrow. But note that, if C is too large, it may has no solution as the separable case.

(d) Small C tends to maximize the margin distance, but the error may be more.

In fact, the above optimization has one equivalence, which helps us understand the intuition behind the SVM. But remember, what we do in the programming is (6).

Proposition 8. *The optimization (6) is equivalent to*

$$\min_{\beta_0, \boldsymbol{\beta}} \frac{1}{2}||\boldsymbol{\beta}||^2 + C\sum_{i=1}^N L\Big(y_i, f(\mathbf{x}_i)\Big), \quad \textit{where } L\Big(y_i, f(\mathbf{x}_i)\Big) := \max\Big(1 - y_i f(\mathbf{x}_i), 0\Big)$$

is the Hinge loss function.

Proof. We consider two cases. If $1 - y_i f(\mathbf{x}_i) \leq 0$, then constraints

$$\begin{cases} \xi_i \geq 0 \\ \xi_i \geq 1 - y_i f(\mathbf{x}_i) \end{cases} \quad \Rightarrow \quad \xi_i \geq 0 \quad \Rightarrow \quad \text{to make } C\sum_{i=1}^N \xi_i \text{ min,} \quad \text{take } \xi_i = 0.$$

If $1 - y_i f(\mathbf{x}_i) > 0$, then the above argument leads to $\xi_i \geq 1 - y_i f(\mathbf{x}_i)$. To attain the min, we take $\xi_i = 1 - y_i f(\mathbf{x}_i)$. The two cases leads to

$$\xi_i = \max\Big(1 - y_i f(\mathbf{x}_i), 0\Big) = L\Big(y_i, f(\mathbf{x}_i)\Big).$$

□

Changing some constants, the SVM optimization we do in essence is equivalent to the following:

$$\hat{f}(\mathbf{x}) \leftarrow \arg\min_{\beta_0, \boldsymbol{\beta}} \sum_{i=1}^{N} L\Big(y_i, f(\mathbf{x}_i)\Big) + \lambda ||\boldsymbol{\beta}||^2,$$

where

$$L\Big(y_i, f(\mathbf{x}_i)\Big) = \max\Big(1 - y_i f(\mathbf{x}_i), 0\Big) \text{ and } f(x) = \mathbb{P}(Y = 1 \mid \mathbf{x}) - \frac{1}{2}.$$

Recall that the (kernel) logistic regression is doing

$$\hat{f}(\mathbf{x}) \leftarrow \arg\min_{f \in \mathscr{H}_k} \sum_{i=1}^{N} L\Big(y_i, f(\mathbf{x}_i)\Big) + \lambda ||f||^2_{\mathscr{H}_k},$$

where

$$L\Big(y_i, f(\mathbf{x}_i)\Big) = \frac{\log\left(1 + e^{-\tilde{y}_i f(\mathbf{x}_i)}\right)}{\log 2} \text{ and } \hat{f}(x) = \log \frac{\mathbb{P}(Y = 1 \mid \mathbf{x})}{\mathbb{P}(Y = -1 \mid \mathbf{x})}.$$

What we are trying to understand is, different classification methods are doing the same thing: modeling the probability by some function f, and minimizing w.r.t. this function f by some loss function L and the data.

2.2 Estimation

In the previous subsection, we mentioned the optimization is based on (6). Introducing Lagrange multipliers, we reach the primal problem.

Problem Setup (Non-separable SVM Primal problem)**.** The primal Lagrange function of non-separable SVM is

$$L_P(\beta_0, \boldsymbol{\beta}, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\mu}) := \frac{1}{2}||\boldsymbol{\beta}||^2 + C\sum_{i=1}^{N} \xi_i - \sum_{i=1}^{N} \alpha_i \Big[y_i\left(\beta_0 + \boldsymbol{\beta}^\top \mathbf{x}_i\right) - (1 - \xi_i)\Big] - \sum_{i=1}^{N} \mu_i \xi_i,$$

where $\boldsymbol{\alpha}_{N\times 1}$ and $\boldsymbol{\mu}_{N\times 1}$ are Lagrange multipliers with $\alpha_i \geq 0$ and $\mu_i \geq 0$ for all $i = 1, \ldots, N$.

The derivation is almost the same as in the separable SVM. Taking partial derivatives, we get

$$\frac{\partial L_P}{\partial \beta_0} = -\sum_{i=1}^{N} \alpha_i y_i \;\Rightarrow\; \sum_{i=1}^{N} \alpha_i^* y_i = 0 \tag{7}$$

$$\frac{\partial L_P}{\partial \boldsymbol{\beta}} = \boldsymbol{\beta} - \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i \quad \Rightarrow \quad \boldsymbol{\beta}^* = \sum_{i=1}^{N} \alpha_i^* y_i \mathbf{x}_i \tag{8}$$

$$\frac{\partial L_P}{\partial \xi_i} = C - \alpha_i - \mu_i \;\Rightarrow\; C = \alpha_i^* + \mu_i^* \quad \forall i = 1, \ldots, N.$$

Plugging them back leads to the dual problem.

Problem Setup (Non-separable SVM Dual problem)**.** The dual Lagrange function of non-separable SVM is

$$L_D(\boldsymbol{\alpha}) := \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j \quad \text{subject to} \quad \sum_{i=1}^{N} \alpha_i y_i = 0 \;\text{ and }\; 0 \le \alpha_i \le C, \;\; \forall i.$$

Using the same procedure in separable SVM, we estimate $\boldsymbol{\alpha}^*$ first, then calculate $\boldsymbol{\beta}^*$ and β_0^*.

2.3 Solution Properties

The above optimization is also a convex one with KKT conditions. So, the following properties follows:

(a) for all $i = 1, \ldots, N$, we have

$$\begin{cases} y_i \left(\beta_0^* + \mathbf{x}_i^\top \boldsymbol{\beta}^*\right) \ge 1 - \xi_i^*, \\ \xi_i \ge 0, \end{cases} \quad \text{and} \quad \begin{cases} \alpha_i^* \ge 0, \\ \mu_i^* \ge 0. \end{cases}$$

(b) for all $i = 1, \ldots, N$, we have

$$\alpha_i^* \Big[y_i(\beta_0^* + \mathbf{x}_i^\top \boldsymbol{\beta}^*) - (1 - \xi_i^*) \Big] = 0 \;\text{ and }\; \mu_i \xi_i = 0.$$

(c) recalling the derivatives (7) and (8), we have for all $i = 1, \ldots, N$

$$\sum_{i=1}^{N} \alpha_i^* y_i = 0, \quad \boldsymbol{\beta}^* = \sum_{i=1}^{N} \alpha_i^* y_i \mathbf{x}_i, \;\text{ and }\; \mu_i^* + \alpha_i^* = C.$$

Now we will explore the relation between α_i and boundary conditions as we did in separable SVM. Starting the discussion of α_i.

(α-i) If $\alpha_i = 0$, meaning that this observation $\mathbf{x}_i$ has no contribution to $\boldsymbol{\beta}^*$, then

$$\alpha_i = 0 \overset{\text{(c3)}}{\Longrightarrow} \mu_i = C \overset{\text{(b2)}}{\Longrightarrow} \xi_i = 0 \overset{\text{(a1)}}{\Longrightarrow} y_i \left(\beta_0^* + \mathbf{x}_i^\top \boldsymbol{\beta}^*\right) \ge 1,$$

we conclude that $\mathbf{x}_i$ is either on the margin, or on the right side.

(α-ii) If $0 < \alpha_i < C$, then

$$0 < \alpha_i < C \overset{\text{(c3)}}{\Longrightarrow} \mu_i > 0 \overset{\text{(c2)}}{\Longrightarrow} \xi_i = 0 \overset{\text{(a1)}}{\Longrightarrow} y_i\left(\beta_0^* + \mathbf{x}_i^\top \boldsymbol{\beta}^*\right) = 1,$$

we conclude that $\mathbf{x}_i$ is on the margin.

(α-iii) If $\alpha_i = C$, then

$$\alpha_i = C \overset{\text{(c3)}}{\Longrightarrow} \mu_i = 0 \overset{\text{(b2)}}{\Longrightarrow} \xi_i \geq 0 \begin{cases} > 0 \overset{\text{(b1)}}{\Longrightarrow} y_i\left(\beta_0^* + \mathbf{x}_i^\top \boldsymbol{\beta}^*\right) < 1 \Rightarrow \text{ wrong side,} \\ = 0 \overset{\text{(b1)}}{\Longrightarrow} y_i\left(\beta_0^* + \mathbf{x}_i^\top \boldsymbol{\beta}^*\right) = 1 \Rightarrow \text{ on the margin.} \end{cases}$$

Conversely, let's discuss by $y_i f(\mathbf{x}_i)$.

($yf(\mathbf{x})$−i) If $y_i f(\mathbf{x}_i) > 1$, then

$$y_i f(\mathbf{x}_i) > 1 \overset{\text{(b1)}}{\Longrightarrow} \alpha_i = 0 \Rightarrow \text{ no contribution to } \boldsymbol{\beta}^*.$$

($yf(\mathbf{x})$−ii) If $y_i f(\mathbf{x}_i) = 1$, then $\mathbf{x}_i$ is on the margin, but we know nothing about α_i:

$$y_i f(\mathbf{x}_i) = 1 \Rightarrow 0 \leq \alpha_i \leq C.$$

There might exist $\mathbf{x}_i$ on the margin, but $\alpha_i = 0$, i.e. it still has no contribution to $\boldsymbol{\beta}^*$.

($yf(\mathbf{x})$−iii) If $y_i f(\mathbf{x}_i) < 1$, then

$$y_i f(\mathbf{x}_i) < 1 \overset{\text{(a1)}}{\Longrightarrow} \xi_i > 0 \overset{\text{(b2)}}{\Longrightarrow} \mu_i = 0 \overset{\text{(c3)}}{\Longrightarrow} \alpha_i^* = C,$$

we conclude $\mathbf{x}_i$ contributes to the SVM.

By the discussion of $yf(\mathbf{x})$, to sum up, partial points on the margin ($yf(\mathbf{x}) - ii$) and all points inside the margin ($yf(\mathbf{x}) - iii$) are supports points, i.e. $\alpha_i^* > 0$.

2.4 Kernalization of SVM

Using the same idea, we can generalized the non-separable SVM by kernel methods, assuming

$$\log \frac{\mathbb{P}_x(Y=1)}{\mathbb{P}_x(Y=-1)} = f(\mathbf{x}), \text{ where } f \in \mathcal{H}_k.$$

If we further use Hinge loss function on the sample, the primal problem is

$$f^* \leftarrow \arg\min_{f \in \mathcal{H}_k} \sum_{i=1}^N L\Big(y_i, f(\mathbf{x}_i)\Big) + \lambda ||f||_{\mathcal{H}_k},$$

and the dual problem is

$$\boldsymbol{\alpha}^* \leftarrow \arg\max_{\boldsymbol{\alpha}} \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{ij} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j), \quad \text{subject to} \quad \begin{cases} 0 \leq \boldsymbol{\alpha} \leq C, \\ \sum_{i=1}^N \alpha_i y_i = 0. \end{cases}$$

3 Generalization of Hinge Loss

Since the Hinge loss is introduced, we can utilize it in the regression setting. Given the sample $(\mathbf{x}_i, y_i)$ for $i = 1, \ldots, N$, the model assumption is

$$\mathbb{E}_{\mathbf{x}_i} Y_i = f(\mathbf{x}_i) = \beta_0 + \boldsymbol{\beta}^\top \mathbf{x}_i,$$

and we estimate the parameters by the Hinge loss

$$\min_{\beta_0, \boldsymbol{\beta}} \sum_{i=1}^{N} L_\epsilon\Big(y_i, f(\mathbf{x}_i)\Big) + \lambda ||\boldsymbol{\beta}||^2, \text{ where } L_\epsilon\Big(y_i, f(\mathbf{x}_i)\Big) = \begin{cases} 0 & \text{if } |y_i - f(\mathbf{x}_i)| \leq \epsilon \\ |y_i - f(\mathbf{x}_i)| - \epsilon & \text{o.w.} \end{cases}$$

When $\epsilon \to 0$, the loss function becomes L_1 loss, which gives the median regression.

Remark. Note the difference between L_1 loss(median reg) and L_1 penalty(lasso). For more, refer §7 of *Pattern Recognition and Machine Learning*.