Support Vector Machine

Suppose there are N data points $\mathbf{x}_1, ..., \mathbf{x}_N$ and let $t_1, ..., t_N$ be their labels where $t_k \in \{-1, 1\}$ for all k. We want to find suitable $\phi(\mathbf{x}_k)$, weight $\mathbf{w}$ and bias b that satisfies

$$\mathbf{w}^T \phi(\mathbf{x}_k) + b = \{> 0 \text{if } t_k = 1 \\ < 0 \text{if } t_k = -1 \quad \text{for all } k$$

This is a linear classifier in feature space. The distance of each data point to this hyper plane is

$$\frac{\left|\mathbf{w}^T \phi(\mathbf{x}_k) + b\right|}{\parallel \mathbf{w} \parallel}$$

According to *statistical learning theory*(not actually understand), we want to maximize the minimal distance

$$\arg\ \max_{\mathbf{w}, b} \left\{ \frac{1}{\parallel \mathbf{w} \parallel} \min_k \left|\mathbf{w}^T \phi(\mathbf{x}_k) + b\right| \right\}$$

Since $t_k\left(\mathbf{w}^T \phi(\mathbf{x}_k) + b\right) > 0$ and $t_k \in \{-1, 1\}$, the problem is equivalent to

$$\arg\ \max_{\mathbf{w}, b} \left\{ \frac{1}{\parallel \mathbf{w} \parallel} \min_k t_k\left(\mathbf{w}^T \phi(\mathbf{x}_k) + b\right) \right\}$$

Observe that any rescaling of $\mathbf{w}$ and b won't change the ultimate value, hence we can set

$$t_k\left(\mathbf{w}^T \phi(\mathbf{x}_k) + b\right) = 1$$

for those $\mathbf{x}_k$ that are closest to the hyper plane. Then other points yield

$$t_k\left(\mathbf{w}^T \phi(\mathbf{x}_k) + b\right) \geq 1$$

Now we simplify the problem into

$$\arg\ \max_{\mathbf{w}, b} \frac{1}{\parallel \mathbf{w} \parallel} \quad \text{s.t.} \quad t_k\left(\mathbf{w}^T \phi(\mathbf{x}_k) + b\right) \geq 1 \text{ for all } k$$

It's equivalent to

$$\arg\ \min_{\mathbf{w}, b} \frac{1}{2} \parallel \mathbf{w} \parallel^2 \quad \text{s.t.} \quad t_k\left(\mathbf{w}^T \phi(\mathbf{x}_k) + b\right) \geq 1 \text{ for all } k$$

And gives the Lagrangian function

$$\mathcal{L}(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2} \parallel \mathbf{w} \parallel^2 + \sum_{k=1}^{N} a_k \left[1 - t_k\left(\mathbf{w}^T \phi(\mathbf{x}_k) + b\right) \right]$$

where $\mathbf{a} = (a_1, ..., a_N)^T \geq \mathbf{0}$. For

$$\max_{\mathbf{a}} \min_{\mathbf{w}, b} \mathcal{L}(\mathbf{w}, b, \mathbf{a}) \quad \text{s.t.} \quad \mathbf{a} \geq 0$$

let gradient vanish with respect to $\mathbf{w}$ and b

$$\nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w}, b, \mathbf{a}) = \mathbf{w} - \sum_{k=1}^{N} a_k t_k \phi(\mathbf{x}_k) = 0$$

$$\nabla_b \mathcal{L}(\mathbf{w}, b, \mathbf{a}) = \sum_{k=1}^{N} a_k t_k = 0$$

Substitute back then yield the dual form

$$\max_{\mathbf{a}} \sum_{k=1}^{N} a_k - \frac{1}{2}\sum_{k=1}^{N}\sum_{m=1}^{N} a_k a_m t_k t_m \phi(\mathbf{x}_k)^T \phi(\mathbf{x}_m)$$

subject to

$$a_k \geq 0, \quad k = 1, ..., N$$
$$\sum_{k=1}^{N} a_k t_k = 0.$$

Let $k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \phi(\mathbf{x}')$ stands for the kernel function and let $y(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) + b$. When $\mathbf{w}$ is the solution, $\mathbf{w} = \sum_k a_k t_k \phi(\mathbf{x}_k)$. Put this into $y(\mathbf{x})$ and have

$$y(\mathbf{x}) = \sum_{k=1}^{N} a_k t_k k(\mathbf{x}_k, \mathbf{x}) + b$$

here we express the classifier in terms of $\{a_k\}$ and the kernel function $k(\mathbf{x}, \mathbf{x}')$.

Apply KKT Condition

We have a primal problem

$$\text{arg min}_{\mathbf{w},b} \frac{1}{2} \parallel \mathbf{w} \parallel^2 \quad \text{s.t.} \quad t_k y(\mathbf{x}_k) - 1 \geq 0 \text{ for all } k$$

and a dual problem

$$\max_{\mathbf{a}} \sum_{k=1}^{N} a_k - \frac{1}{2}\sum_{k=1}^{N}\sum_{m=1}^{N} a_k a_m t_k t_m k(\mathbf{x}_k, \mathbf{x}_m) \quad \text{s.t.} \quad a_k \geq 0, \forall\, k \text{ and } \sum_{k=1}^{N} a_k t_k = 0$$

The KKT condition needs further constraint that

$$a_k\{t_k y(\mathbf{x}_k) - 1\} = 0$$

This implies $a_k = 0$ or $t_k y(\mathbf{x}_k) = 1$. Any data point has $a_k = 0$ will not contribute to the classifier. The rest data points have $t_k y(\mathbf{x}_k) = 1$ are called *support vector.*This tells us that we only need support vectors to predict new point though we need whole data to train. Mathematically, choose one support vector $\mathbf{x}'$, we can get b by

$$t' y(\mathbf{x}') = t' \left\{ \sum_m a_m t_m k(\mathbf{x}_m, \mathbf{x}') + b \right\} = 1$$

where m stands for the index of support vector. In practical, multiply each side by one label t_k of one support vector and have

$$b = t_k - \left\{ \sum_m a_m t_m k(\mathbf{x}_m, \mathbf{x}_k) \right\} \text{ for all } k$$

Take the average of all possible b as the final one

$$b = \frac{1}{M} \sum_k \left(t_k - \left\{ \sum_m a_m t_m k(\mathbf{x}_m, \mathbf{x}_k) \right\} \right)$$

where M is the number of support vectors and k and m are both the index of support vector.

Soft Margin Support Vector Machine

The original SVM is

$$\min_{\mathbf{w},b} \frac{1}{2} \parallel \mathbf{w} \parallel^2 \quad \text{s.t.} \quad t_k y(\mathbf{x}_k) \geq 1 \text{ for all } k$$

To allow some points can be misclassified, we introduce each point a *slack variable* ξ_k that is defined by

$$\xi_k = \{0 \text{if } t_k y(\mathbf{x}_k) \geq 1$$
$$|t_k - y(\mathbf{x}_k)| \text{otherwise}$$

Look deeper into this definition. If $0 < t_k y(\mathbf{x}_k) < 1$, we have $0 < y(\mathbf{x}_k) < 1$ or $-1 < y(\mathbf{x}_k) < 0$. Hence $0 < \xi_k = 1 - t_k y(\mathbf{x}_k) < 1$. If $t_k y(\mathbf{x}_k) = 0$, $\xi_k = 1 - t_k y(\mathbf{x}_k) = 1$. If $t_k y(\mathbf{x}_k) < 0$, $\xi_k = 1 - t_k y(\mathbf{x}_k) > 1$. In summary

$$\xi_k \begin{cases} = 0 & \text{if } t_k y(\mathbf{x}_k) \in [1 \\ \infty) & \\ = 1 - t_k y(\mathbf{x}_k) \begin{cases} \in (0,1) \text{if } t_k y(\mathbf{x}_k) \in (0,1) \\ = 1 \quad \text{if } t_k y(\mathbf{x}_k) = 0 \\ > 1 \quad \text{if } t_k y(\mathbf{x}_k) \in (-\infty, 0) \end{cases} & \end{cases}$$

Replace the constrain with

$$t_k y(\mathbf{x}_k) \geq 1 - \xi_k \text{ for all } k$$

This is so called *soft margin.*

When there exist outliers, they'll have extremely large ξ_k. To avoid this, here comes the soft SVM that also minimize slack vairable

$$\min_{\mathbf{w},b,\boldsymbol{\xi}} C \sum_{k=1}^{N} \xi_k + \frac{1}{2} \parallel \mathbf{w} \parallel^2 \quad \text{s.t.} \quad t_k y(\mathbf{x}_k) \geq 1 - \xi_k \text{ for all } k$$

where C is some constant. Briefly speaking, slack vairables relax some points' constarint, their $t_k y(\mathbf{x}_k)$ only needs to larger than some value smaller than 1. That's why we call this *soft margin.*

Apply KKT Condition

The Lagrangian of soft SVM is

$$\mathcal{L}(\mathbf{w}, b, \xi, \mathbf{a}, \boldsymbol{\mu}) = \frac{1}{2} \parallel \mathbf{w} \parallel^2 + C \sum_{k=1}^{N} \xi_k - \sum_{k=1}^{N} a_k \{ t_k y(\mathbf{x}_k) - 1 + \xi_k \} - \sum_{k=1}^{N} \mu_k \xi_k$$

where $a_k, \mu_k \geq 0$, $t_k y(\mathbf{x}_k) - 1 + \xi_k \geq 0$ and note that slack variables are non negative $\xi_k \geq 0$. The KKT conditions are

$$
\begin{aligned}
\textbf{Dual Feasibility } a_k \geq 0, \qquad & \mu_k \geq 0 \\
\textbf{Primal Feasibility } t_k y(\mathbf{x}_k) - 1 + \xi_k \geq 0, \qquad & \xi_k \geq 0 \\
\textbf{Complmentary Slackness } a_k(t_k y(\mathbf{x}_k) - 1 + \xi_k) = 0, & \, \mu_k \xi_k = 0
\end{aligned}
$$

Use $y(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) + b$ and compute gradients of $\mathcal{L}$ with respect to $\mathbf{w}$, b and ξ

$$
\begin{aligned}
\nabla_{\mathbf{w}} \mathcal{L} &= \mathbf{w} - \sum_{k=1}^{N} a_k t_k \phi(\mathbf{x}_n) = 0 \\
\nabla_b \mathcal{L} &= \sum_{k=1}^{N} a_k t_k = 0 \\
\nabla_{\xi_k} \mathcal{L} &= a_k - C - \mu_k = 0
\end{aligned}
$$

Substitute back and get the dual form

$$
\max_{\mathbf{a}} \sum_{k=1}^{N} a_k - \frac{1}{2} \sum_{k=1}^{N} \sum_{m=1}^{N} a_k a_m t_k t_m k(x_k, x_m) \quad \text{s.t.} \quad 0 \leq a_k \leq C, \forall\, k \text{ and} \sum_{k=1}^{N} a_k t_k = 0
$$

It’s the same as normal SVM, the only difference is the constraint of a_k, which is known as the *box constraint.*