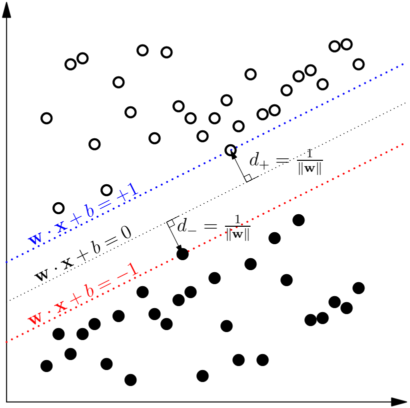
Support Vector Machines

September 22, 2018

Support Vector Machine: Linearly Separable Case

Binary Classification

Given the data $\{(\mathbf{x}_i, y_i)\}_{i=1}^n, y_i \in \{-1, +1\}, \mathbf{x}_i \in \mathbb{R}^d$, find the hyperplane with maximum "margin" — the gap between parallel hyperplanes seperating two classes where the vectors of neither class can lie.



Let $\mathbf{w} \cdot \mathbf{x} + b = 0$ be the seperating hyperplane and d_+, d_- be the shortest distance to the closest objects from the class $+1, -1$, respectively. Suppose all the training data satisfy the following constraints:

$$\begin{aligned} \mathbf{w} \cdot \mathbf{x}_i + b &\geqslant +1 \quad \text{for } y_i = +1 \\ \mathbf{w} \cdot \mathbf{x}_i + b &\leqslant -1 \quad \text{for } y_i = -1 \end{aligned}$$

These constraints can be combined as

$$y_i\left(\mathbf{w} \cdot \mathbf{x}_i + b\right) - 1 \geqslant 0 \quad \forall i.$$

Let $\mathbf{x}_0$ be a point on $\mathbf{w} \cdot \mathbf{x} + b = 0$, then $\mathbf{x}_0$ projected on $\mathbf{w} \cdot \mathbf{x} + b = 1$ is $\mathbf{x}_0 + t\,\mathbf{w}$ with t to be determined; we have

$$\mathbf{w} \cdot (\mathbf{x}_0 + t\,\mathbf{w}) + \mathbf{b} = 1 \Longrightarrow t = \frac{1}{\|\mathbf{w}\|^2} \Longrightarrow d_+ = \|t\,\mathbf{w}\| = \frac{1}{\|\mathbf{w}\|}.$$

By the same token, $d_- = \frac{1}{\|\mathbf{w}\|}$; the margin $= d_+ + d_- = \frac{2}{\|\mathbf{w}\|}$.

Convex Programming Problem

Given $f, g_1, g_2, \ldots, g_m$ convex functions defined on $\mathbb{R}^d$, minimize $f(\mathbf{x})$ subject to the constraints $\mathbf{x} \geqslant 0$, $g_1(\mathbf{x}) \leqslant 0, g_2(\mathbf{x}) \leqslant 0, \ldots, g_m(\mathbf{x}) \leqslant 0$.

Feasible set $\mathbf{X}$: $\{\mathbf{x}|\mathbf{x} \geqslant 0, g_1(\mathbf{x}) \leqslant 0, g_2(\mathbf{x}) \leqslant 0, \ldots, g_m(\mathbf{x}) \leqslant 0\}$; $\mathbf{X}$ is convex.

The Lagrangian function $F(\mathbf{x}, \mathbf{y})$ of Convex Programming Problem

$$F(\mathbf{x}, \mathbf{y}) = f(\mathbf{x}) + y_1 g_1(\mathbf{x}) + y_2 g_2(\mathbf{x}) + \cdots + y_m g_m(\mathbf{x}) \quad \mathbf{y} = (y_1, y_2, \ldots, y_m).$$

Saddle Point Problem

Determine a saddle point of F, i.e. a point $(\mathbf{x}_0, \mathbf{y}_0) \in \mathbb{R}^{d+m}$ such that

$$\mathbf{x}_0 \geqslant 0,\ \mathbf{y}_0 \geqslant 0,\ F(\mathbf{x}_0, \mathbf{y}) \leqslant F(\mathbf{x}_0, \mathbf{y}_0) \leqslant F(\mathbf{x}, \mathbf{y}_0) \quad \forall \mathbf{x} \geqslant 0, \mathbf{y} \geqslant 0.$$

Lemma

Let $f_1, f_2, \ldots, f_k$ be convex functions defined on a non-empty convex set $\mathbf{Y}$ in $\mathbb{R}^d$. Suppose that no $\mathbf{y} \in \mathbf{Y}$ such that $f_1(\mathbf{y}) < 0$, $f_2(\mathbf{y}) < 0$, …, $f_k(\mathbf{y}) < 0$. Then there exists $a_1, a_2, \ldots, a_k \geqslant 0$, not all zero, such that

$$a_1 f_1(\mathbf{y}) + a_2 f_2(\mathbf{y}) + \cdots + a_k f_k(\mathbf{y}) \geqslant 0 \quad \forall \mathbf{y} \in \mathbf{Y}.$$

Theorem

Let $(\mathbf{x}_0, \mathbf{y}_0)$ be a saddle point of the Lagrangian function F, then $\mathbf{x}_0$ is a solution to the convex programming problem and $F(\mathbf{x}_0, \mathbf{y}_0) = f(\mathbf{x}_0)$.

Proof.

Let $\mathbf{x}_0 = (x_1^0, x_2^0, \ldots, x_p^0) \geqslant 0$, $\mathbf{y}_0 = (y_1^0, y_2^0, \ldots, y_m^0) \geqslant 0$. For all $\mathbf{y} = (y_1, y_2, \ldots, y_m) \geqslant 0$, by def of saddle point, $F(\mathbf{x}_0, \mathbf{y}_0) \geqslant F(\mathbf{x}_0, \mathbf{y})$;

$$y_1^0 g_1(\mathbf{x}_0) + y_2^0 g_2(\mathbf{x}_0) + \cdots + y_m^0 g_m(\mathbf{x}_0) \geqslant y_1 g_1(\mathbf{x}_0) + y_2 g_2(\mathbf{x}_0) + \cdots + y_m g_m(\mathbf{x}_0)$$

Fixing $y_2, y_3, \ldots, y_m$ and letting $y_1 \to \infty$, we have $g_1(\mathbf{x}_0) \leqslant 0$; similarly $g_i(\mathbf{x}_0) \leqslant 0$, $i = 2, \ldots, m.$, so $\mathbf{x}_0$ belongs to the feasible set $\mathbf{X}$. Now set $\mathbf{y} = 0$ in the inequality $F(\mathbf{x}_0, \mathbf{y}_0) \geqslant F(\mathbf{x}_0, \mathbf{y})$, we have

$$0 \leqslant y_1^0 g_1(\mathbf{x}_0) + y_2^0 g_2(\mathbf{x}_0) + \cdots + y_m^0 g_m(\mathbf{x}_0) \leqslant 0,$$

whence $y_1^0 g_1(\mathbf{x}_0) + y_2^0 g_2(\mathbf{x}_0) + \cdots + y_m^0 g_m(\mathbf{x}_0) = 0, \quad F(\mathbf{x}_0, \mathbf{y}_0) = f(\mathbf{x}_0)$. Since $F(\mathbf{x}_0, \mathbf{y}_0) \leqslant F(\mathbf{x}, \mathbf{y}_0)\, \forall \mathbf{x} \geqslant 0$, we deduce that $\forall \mathbf{x} \in \mathbf{X}$,

$$f(\mathbf{x}_0) \leqslant f(\mathbf{x}) + y_1^0 g_1(\mathbf{x}) + y_2^0 g_2(\mathbf{x}) + \cdots + y_m^0 g_m(\mathbf{x}) \leqslant f(\mathbf{x})$$

which shows that $\mathbf{x}_0$ is a solution of the convex programming problem. $\square$

Karush-Kuhn-Tucker (KKT) Condition

Suppose that the convex functions $f, g_1, g_2, \ldots, g_m : \mathbb{R}^p \to \mathbb{R}$ are differentiable. Then $(\mathbf{x}_0, \mathbf{y}_0)$ is a saddle point of the Lagrangian F iff

$$
\begin{aligned}
&\mathbf{x}_0 \geqslant 0 \\
&\frac{\partial F}{\partial x_j}(\mathbf{x}_0, \mathbf{y}_0) = \frac{\partial f}{\partial x_j}(\mathbf{x}_0) + \sum_{i=1}^{m} y_i^0 \frac{\partial g_i}{\partial x_j}(\mathbf{x}_0) \geqslant 0 \\
&\frac{\partial F}{\partial x_j}(\mathbf{x}_0, \mathbf{y}_0) = 0 \quad \text{whenever } x_j^0 > 0 \\
&\mathbf{y}_0 \geqslant 0 \\
&\frac{\partial F}{\partial y_j}(\mathbf{x}_0, \mathbf{y}_0) = g_j(\mathbf{x}_0) \leqslant 0 \\
&\frac{\partial F}{\partial y_j}(\mathbf{x}_0, \mathbf{y}_0) = 0 \quad \text{whenever } y_j^0 > 0
\end{aligned}
$$

Determine the Hyperplane with Maximum Margin

$$\text{maximize } \frac{1}{\|\mathbf{w}\|} \Longleftrightarrow \text{minimize } \|\mathbf{w}\|^2$$
$$\text{subject to} \quad y_i (\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \geqslant 0 \quad \forall i = 1, 2, \ldots, n.$$

The above constrained optimization problem can be solved via the Lagrangian approach: set

$$\mathcal{L} = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i \{y_i (\mathbf{w} \cdot \mathbf{x}_i + b) - 1\} \tag{1}$$

with n Lagrange multipliers $\alpha_i \geqslant 0$, the problem becomes to minimize $\mathcal{L}$...

KKT Condition

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = 0 \Longrightarrow \mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i \tag{2}$$

$$\frac{\partial \mathcal{L}}{\partial b} = 0 \Longrightarrow \sum_{i=1}^{n} \alpha_i y_i = 0 \tag{3}$$

$$y_i (\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \geqslant 0 \quad i = 1, 2, \ldots, n \tag{4}$$

$$\alpha_i \geqslant 0 \quad i = 1, 2, \ldots, n \tag{5}$$

$$\alpha_i \{y_i (\mathbf{w} \cdot \mathbf{x}_i + b) - 1\} = 0 \quad i = 1, 2, \ldots, n \tag{6}$$

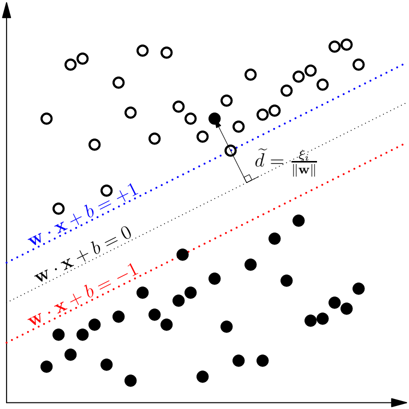
Substitute the KKT condition (2) into $\mathcal{L}$ (1), we have the dual Lagrangian $\mathcal{L}_D$ (Wolfe dual):

$$\mathcal{L}_D = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j$$

Now the optimization problem becomes maximizing $\mathcal{L}_D$ subject to

$$\sum_{i=1}^{n} \alpha_i y_i = 0.$$

Linearly Non-Separable Case



We introduce *positive slack variables* $\{\xi_i\}_{i=1}^{n}$ into the constraints

$$\begin{gathered}
\mathbf{w} \cdot \mathbf{x}_i + b \geqslant +1 - \xi_i \quad \text{for } y_i = +1 \\
\mathbf{w} \cdot \mathbf{x}_i + b \leqslant -1 + \xi_i \quad \text{for } y_i = -1 \\
\xi_i \geqslant 0 \quad i = 1, 2, \ldots, n.
\end{gathered}$$

These constraints can be combined as

$$y_i \left(\mathbf{w} \cdot \mathbf{x}_i + b \right) - 1 + \xi_i \geqslant 0 \quad i = 1, 2, \ldots n.$$

If error occurs, $\xi_i > 1$. The objective function is changed to

$$\text{minimize} \quad \frac{1}{2} \left\| \mathbf{w} \right\|^2 + C \sum_{i=1}^{n} \xi_i$$

where C (capacity) controls the tolerance to errors on the training set.

The above constrained optimization problem can be solved via the Lagrangian approach: set

$$\mathcal{L} = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n} \alpha_i \left\{y_i \left(\mathbf{w} \cdot \mathbf{x}_i + b\right) - 1 + \xi_i\right\} - \sum_{i=1}^{n} \xi_i \mu_i \tag{7}$$

with $2n$ Lagrange multipliers $\alpha_i, \mu_i \geqslant 0$.

KKT Condition

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = 0 \Longrightarrow \mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i \tag{8}$$

$$\frac{\partial \mathcal{L}}{\partial b} = 0 \Longrightarrow \sum_{i=1}^{n} \alpha_i y_i = 0 \tag{9}$$

$$\frac{\partial \mathcal{L}}{\partial \xi_i} = 0 \Longrightarrow C - \alpha_i - \xi_i = 0 \quad i = 1, 2, \ldots, n \tag{10}$$

$$y_i \left(\mathbf{w} \cdot \mathbf{x}_i + b\right) - 1 + \xi_i \geqslant 0 \quad i = 1, 2, \ldots, n \tag{11}$$

$$\alpha_i \geqslant 0 \quad i = 1, 2, \ldots, n \tag{12}$$

$$\mu_i \geqslant 0 \quad i = 1, 2, \ldots, n \tag{13}$$

$$\xi_i \geqslant 0 \quad i = 1, 2, \ldots, n \tag{14}$$

$$\alpha_i \left\{y_i \left(\mathbf{w} \cdot \mathbf{x}_i + b\right) - 1 + \xi_i\right\} = 0 \quad i = 1, 2, \ldots, n \tag{15}$$

Substitute the KKT condition (8) into $\mathcal{L}$ (7), we have the dual Lagrangian $\mathcal{L}_D$ (Wolfe dual):

$$\mathcal{L}_D = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j$$

Now the optimization problem becomes maximizing $\mathcal{L}_D$ subject to

$$0 \leqslant \alpha_i \leqslant C \quad \text{and} \quad \sum_{i=1}^{n} \alpha_i y_i = 0.$$

Nonlinear SVM: Kernel Trick

How can the above methods be generalized to the case where the seperating boundary is not linear? Map the data into another space $\mathcal{H}$ and perform classification there. Say the mapping function be $\Psi : \mathbb{R}^d \to \mathcal{H}$. The training algorithm now depends on $\Psi(\mathbf{x}_i) \cdot \Psi(\mathbf{x}_j)$. If there were a "kernel function" K such that

$$K(\mathbf{x}_i, \mathbf{x}_j) = \Psi(\mathbf{x}_i) \cdot \Psi(\mathbf{x}_j),$$

we don't need to know the exact form of Ψ.

Mercer's Condition. Examples of Kernel

Mercer's Condition

$$K(\mathbf{x}_i, \mathbf{x}_j) = \Psi(\mathbf{x}_i) \cdot \Psi(\mathbf{x}_j)$$

iff

$$\int K(\mathbf{x}, \mathbf{y}) g(\mathbf{x}) g(\mathbf{y}) \, \mathrm{d}\mathbf{x} \, \mathrm{d}\mathbf{y} \geqslant 0 \quad \text{for square integrable functions } g.$$

Examples of Kernel:

$$K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\frac{1}{2}(\mathbf{x}_i - \mathbf{x}_j)^\top \Sigma^{-1} (\mathbf{x}_i - \mathbf{x}_j)}$$
$$K(\mathbf{x}_i, \mathbf{x}_j) = \left(\mathbf{x}_i^\top \mathbf{x}_j + 1\right)^p$$
$$K(\mathbf{x}_i, \mathbf{x}_j) = \tanh\left(k\,\mathbf{x}_i^\top \mathbf{x}_j + \delta\right)$$