Soft Margin SVM

Functional margin $1 - \xi_i \leq 1$:

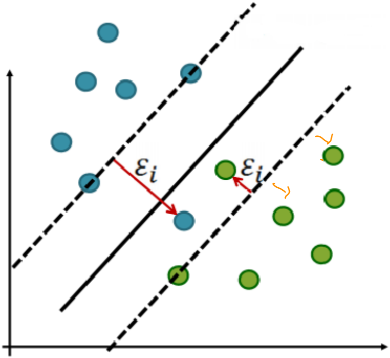
$$\min_{w,b,\xi} \frac{1}{2}||w||^2 + C\sum_{i=1}^{m}\xi_i$$

slack variable.

$$s.t.\ y^{(i)}(w^T x^{(i)} + b) \geq 1 - \xi_i$$

$$\xi_i \geq 0, i = 1, \ldots, m$$

- C: relative weight on the regularizer
- L_1 regularization let most $\xi_i = 0$, such that their functional margins $1 - \xi_i = 1$



Soft Margin SVM

The generalized Lagrangian function:

$$L(w, b, \xi, \alpha, r) = \frac{1}{2}||w||^2 + C\sum_{i=1}^{m}\xi_i - \sum_{i}^{m}\alpha_i\left[y^{(i)}(w^T x^{(i)} + b) - 1 + \xi_i\right] - \sum_{i=1}^{m} r_i\xi_i$$

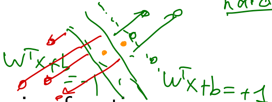
Soft Margin SVM

The generalized Lagrangian function:

$$L(w, b, \xi, \alpha, r) = \frac{1}{2}||w||^2 + C\sum_{i=1}^{m}\xi_i - \sum_{i}^{m}\alpha_i\left[y^{(i)}(w^T x^{(i)} + b) - 1 + \xi_i\right] - \sum_{i=1}^{m} r_i\xi_i$$

Dual problem:

Soft Margin SVM



hard margin: seperable

$y^i(w^Tx^i+b) \geq 1-\xi_i$

soft margin: allows unseperable samples.

The generalized Lagrangian function:

$$L(w,b,\xi,\alpha,r) = \frac{1}{2}||w||^2 + C\sum_{i=1}^{m}\xi_i - \sum_{i}^{m}\alpha_i\left[y^{(i)}(w^Tx^{(i)}+b) - 1 + \xi_i\right] - \sum_{i=1}^{m} r_i\xi_i$$

Dual problem:

$$\max_{\alpha} \; W(\alpha) = \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{m} y^{(i)}y^{(j)}\alpha_i\alpha_j\langle x^{(i)}, x^{(j)}\rangle$$

$$s.t. \; 0 \leq \alpha_i \leq C, i = 1, \ldots, m$$

$$\sum_{i=1}^{m}\alpha_i y^{(i)} = 0$$

w^* is the same as the non-regularizing case, but b^* has changed.

Soft Margin SVM

Dual problem:

$$\max_{\alpha} W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y^{(i)} y^{(j)} \alpha_i \alpha_j \langle x^{(i)}, x^{(j)} \rangle$$

$$s.t. \; 0 \leq \alpha_i \leq C, i = 1, \ldots, m$$

$$\sum_{i=1}^{m} \alpha_i y^{(i)} = 0$$

By the KKT dual-complentary conditions, for all i, $\alpha_i^* g_i(w^*) = 0$

$$\begin{aligned} \alpha_i = 0 \quad &\Longleftrightarrow \\ \alpha_i = C \quad &\Longleftrightarrow \\ 0 < \alpha_i < C \quad &\Longleftrightarrow \end{aligned}$$

Soft Margin SVM

Dual problem:

$$\max_{\alpha} W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y^{(i)} y^{(j)} \alpha_i \alpha_j \langle x^{(i)}, x^{(j)} \rangle$$

$$s.t.\ 0 \leq \alpha_i \leq C, i = 1, \ldots, m$$

$$\sum_{i=1}^{m} \alpha_i y^{(i)} = 0$$

By the KKT dual-complentary conditions, for all i, $\alpha_i^* g_i(w^*) = 0$

support vector

$\alpha_i \neq 0$

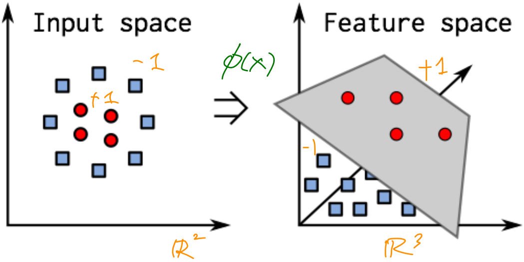
$\alpha_i = 0$	$\Longleftrightarrow$	$y^{(i)}(w^T x^{(i)} + b) \geq 1$	correct side of margin
$\alpha_i = C$	$\Longleftrightarrow$	$y^{(i)}(w^T x^{(i)} + b) \leq 1$	wrong side of margin
$0 < \alpha_i < C$	$\Longleftrightarrow$	$y^{(i)}(w^T x^{(i)} + b) = 1$	at margin

w^*

Kernel SVM

Non-linear SVM

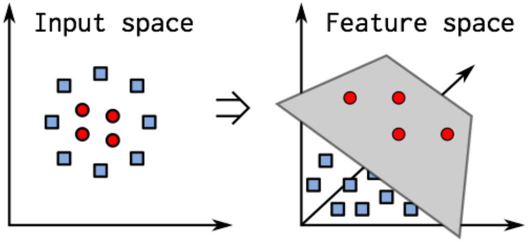
For non-separable data, we can use the **kernel trick**: Map input values $x \in \mathbb{R}^d$ to a higher dimension $\phi(x) \in \mathbb{R}^D$, such that the data becomes separable.



$\phi : \mathbb{R}^2 \to \mathbb{R}^3$.

Non-linear SVM

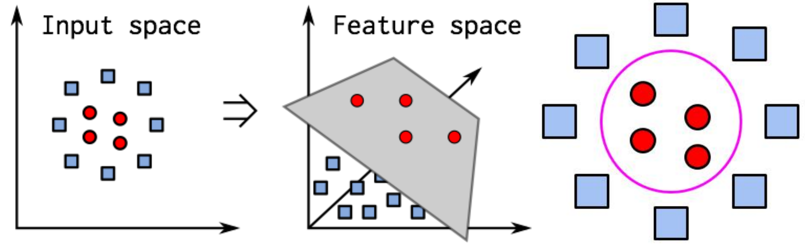
For non-separable data, we can use the **kernel trick**: Map input values $x \in \mathbb{R}^d$ to a higher dimension $\phi(x) \in \mathbb{R}^D$, such that the data becomes separable.



- ϕ is called a **feature mapping**.

Non-linear SVM

For non-separable data, we can use the **kernel trick**: Map input values $x \in \mathbb{R}^d$ to a higher dimension $\phi(x) \in \mathbb{R}^D$, such that the data becomes separable.



- ϕ is called a **feature mapping**.
- The classification function $w^T x + b$ becomes nonlinear: $w^T \phi(x) + b$

Kernel Function

Given a feature mapping ϕ, we define the **kernel function** to be

$$K(x, z) = \phi(x)^T \phi(z)$$

Kernel Function

Given a feature mapping ϕ, we define the **kernel function** to be

$$K(x,z) = \phi(x)^T\phi(z)$$

Some kernel functions are easier to compute than $\phi(x)$, e.g.

$K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$

$$K(x,z) = (x^Tz)^2 = \left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}\right)^2 .$$

$x, z \in \mathbb{R}^2$

$$= (x_1z_1 + x_2z_2)(x_1z_1 + x_2z_2)$$

$$= (x_1z_1)^2 + (x_1z_1)(x_1z_2) + (x_2z_2)(x_1z_1) + (x_2z_2)^2$$

$$= (x_1^2z_1^2) + (x_1x_2)(z_1z_2) + (x_2x_1)(z_2z_1) + (x_2^2z_2^2)$$

$$= \left\langle \begin{bmatrix} x_1^2 \\ x_1x_2 \\ x_2x_1 \\ x_2^2 \end{bmatrix}, \begin{bmatrix} z_1^2 \\ z_1z_2 \\ z_2z_1 \\ z_2^2 \end{bmatrix} \right\rangle$$

$$\phi\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = \begin{bmatrix} x_1^2 \\ x_1x_2 \\ x_2x_1 \\ x_2^2 \end{bmatrix}$$

$$= \langle \phi(x), \phi(z) \rangle .$$

Kernel Function

Given a feature mapping ϕ, we define the **kernel function** to be

$$K(x, z) = \phi(x)^T \phi(z)$$

Some kernel functions are easier to compute than $\phi(x)$, e.g.

$x, z \in \mathbb{R}^n.$

$$K(x, z) = (x^T z)^2 = \left(\sum_{i=1}^{n} x_i z_i\right)\left(\sum_{j=1}^{n} x_j z_j\right) = \sum_{i=1}^{n}\sum_{j=1}^{n} x_i x_j z_i z_j$$
$$= \phi(x)^T \phi(z)$$

Kernel Function

Given a feature mapping ϕ, we define the **kernel function** to be

$$K(x, z) = \phi(x)^T \phi(z)$$

Some kernel functions are easier to compute than $\phi(x)$, e.g.

A kernel example:

$$K(x, z) = (x^T z)^2 = \left(\sum_{i=1}^{n} x_i z_i\right)\left(\sum_{j=1}^{n} x_j z_j\right) = \sum_{i=1}^{n}\sum_{j=1}^{n} x_i x_j z_i z_j$$
$$= \phi(x)^T \phi(z)$$

where $\phi(x) = \begin{bmatrix} x_1 x_1 \\ x_1 x_2 \\ x_1 x_n \\ \vdots \\ x_2 x_1 \\ x_n x_{n-1} \\ x_n x_n \end{bmatrix}$ takes $O(n^2)$ operations to compute, while

$(x^T z)^2$ only takes $O(n)$

Kernel SVM

In the dual problem, replace $\langle x_i, y_j \rangle$ with $\langle \phi(x_i), \phi(y_i) \rangle = K(x_i, x_j)$

$<x_i, x_j> \Rightarrow <\phi(x_i), \phi(x_j)>$

$$\max_{\alpha} W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y^{(i)} y^{(j)} \alpha_i \alpha_j K(x_i, x_j)$$

$$s.t.\ 0 \leq \alpha_i \leq C, i = 1, \ldots, m$$

$$\sum_{i=1}^{m} \alpha_i y^{(i)} = 0$$

Kernel SVM

In the dual problem, replace $\langle x_i, y_j \rangle$ with $\langle \phi(x_i), \phi(y_i) \rangle = K(x_i, x_j)$

$$\max_{\alpha} W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y^{(i)} y^{(j)} \alpha_i \alpha_j K(x_i, x_j)$$

$$s.t.\ 0 \leq \alpha_i \leq C, i = 1, \ldots, m$$

$$\sum_{i=1}^{m} \alpha_i y^{(i)} = 0$$

$x^{(i)}$

No need to compute $w^* = \sum_{i=1}^{m} \alpha_i^* y^{(i)} \phi(x^{(i)})$ explicitly since

Given new sample x.

$$f(x) = w^T \phi(x) + b = \left(\sum_{i=1}^{m} \alpha_i y^{(i)} \phi(x^{(i)}) \right)^T \cdot \phi(x) + b$$

$$= \sum_{i=1}^{m} \alpha_i y^{(i)} \langle \phi(x^{(i)}), \phi(x) \rangle + b$$

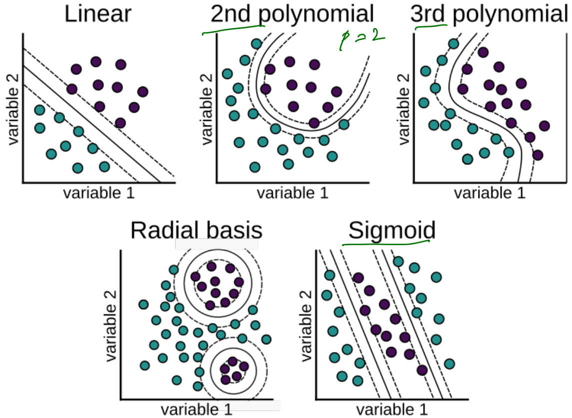
$$= \sum_{i=1}^{m} \alpha_i y^{(i)} K(x^{(i)}, x) + b$$

$\sum_{i=1..m}$
$\alpha_i \neq 0$

Kernel Matrix

kernel functions measure the similarity between samples x, z, e.g.

- Linear kernel: $K(x,z) = (x^T z)$
- Polynomial kernel: $K(x,z) = (x^T z + 1)^p$
- Gaussian / radial basis function (RBF) kernel:
  $K(x,z) = \exp\left(-\frac{||x-z||^2}{2\sigma^2}\right)$



$p=2$

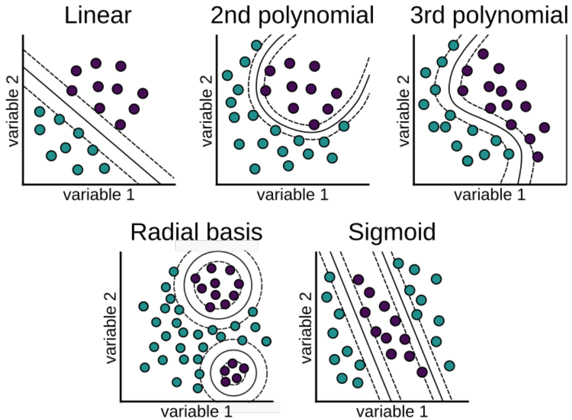
$p=3$

Kernel Matrix

$K(x_1, x_2)$ $\forall\ x, z \in \mathbb{R}^n.$

kernel functions measure the similarity between samples x, z, e.g.

- Linear kernel: $K(x, z) = (x^T z)$
- Polynomial kernel: $K(x, z) = (x^T z + 1)^p$
- Gaussian / radial basis function (RBF) kernel:
  $K(x, z) = \exp\left(-\frac{||x-z||^2}{2\sigma^2}\right)$

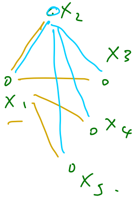


Can any function $K(x, y)$ be a kernel function?

$\parallel$

$\langle \phi(x), \phi(y) \rangle$ for some ϕ?

Kernel Matrix



Given m training samples $(x_1, y_1), \ldots, (x_m, y_m)$

Represent kernel function as a matrix $K \in \mathbb{R}^{m\times m}$ where $K_{i,j} = K(x_i, x_j) = \phi(x_i)^T\phi(x_j)$.

$K(x_1, x_m)$

$$K = \begin{array}{c} \\ 1 \\ \\ \\ m \end{array}\begin{array}{c} \begin{array}{cccc} 1 & 2 & & \end{array} \\ \begin{bmatrix} \phi(x_1)^T\phi(x_1) & \phi(x_1)^T\phi(x_2) & \cdots & \phi(x_1)^T\phi(x_m) \\ \phi(x_2)^T\phi(x_1) & & & \\ \vdots & & \ddots & \\ \phi(x_m)^T\phi(x_1) & \cdots & & \phi(x_m)^T\phi(x_m) \end{bmatrix} \end{array}$$

$$x_1 \longrightarrow \begin{bmatrix} \phi(x_1)^T\phi(x_1) \\ \phi(x_1)^T\phi(x_2) \\ \vdots \\ \phi(x_1)^T\phi(x_m) \end{bmatrix} = \begin{bmatrix} K(x_1, x_1) \\ K(x_1, x_2) \\ \vdots \\ K(x_1, x_m) \end{bmatrix}$$

Kernel Matrix

How to prove function K is a valid kernel?

e.g. $K(x,z) = (x^T z)^2$. ① $\to \langle \phi(x), \phi(z) \rangle$

② $\to$ applying mercer theorem

Represent kernel function as a matrix $K \in \mathbb{R}^{m \times m}$ where $K_{i,j} = K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$.

$(x_i^T x_j + c)^p$

Let $a \in \mathbb{R}^m$

Show $a^T K a \geq 0$.

$\Rightarrow$ K is a valid kernel

Theorem (Mercer)

Let $K : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ Then K is a valid (Mercer) kernel if and only if for any finite training set $\{x^{(1)}, \ldots, x^{(m)}\}$, K is symmetric positive semi-definite.

i.e. $K_{i,j} = K_{j,i}$ and $x^T K x \geq 0$ for all $x \in \mathbb{R}^m$

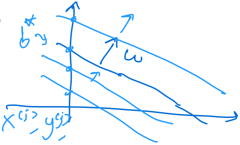
$K = K^T$

Kernel SVM Summary

- Input: m training samples $(x^{(i)}, y^{(i)}), y^i \in \{-1, 1\}$, kernel function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, constant $C > 0$
- Output: non-linear decision function $f(x)$
- Step 1: solve the dual optimization problem for α^*

$$\max_{\alpha} W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y^{(i)} y^{(j)} \alpha_i \alpha_j K(x^{(i)}, x^{(j)})$$

$$s.t.\ 0 \le \alpha_i \le C, \sum_{i=1}^{m} \alpha_i y^{(i)} = 0, i = 1, \ldots, m$$



- Step 2: compute the optimal decision function

for some $x^{(j)}, y^{(j)}$

$$b^* = y^{(j)} - \sum_{i=1}^{m} \alpha_i^* y^{(i)} K(x^{(i)}, x^{(j)}) \text{ for some } 0 < \alpha_j < C$$

$$f(x) = \sum_{i=1}^{m} \alpha_i y^{(i)} K(x^{(i)}, x) + b^*$$

In practice, it's more efficient to compute kernel matrix K in advance. $(m \times m)$

SVM in Practice

(SMO) $\rightarrow$, coordinate ascent α_i, α_j

Sequential Minimal Optimization: a fast algorithm for training soft margin kernel SVM

- Break a large SVM problem into smaller chunks, update two α_i's at a time
- Implemented by most SVM libraries.

SVM in Practice

Sequential Minimal Optimization: a fast algorithm for training soft margin kernel SVM

- Break a large SVM problem into smaller chunks, update two α_i's at a time
- Implemented by most SVM libraries.

Other related algorithms

- least square SVM

- Support Vector Regression (SVR)
- Multi-class SVM (Koby Crammer and Yoram Singer. 2002. *On the algorithmic implementation of multiclass kernel-based vector machines*. J. Mach. Learn. Res. 2 (March 2002), 265-292.)

SVM in Practice

Sequential Minimal Optimization: a fast algorithm for training soft margin kernel SVM

- Break a large SVM problem into smaller chunks, update two α_i's at a time
- Implemented by most SVM libraries.

Other related algorithms

- Support Vector Regression (SVR)
- Multi-class SVM (Koby Crammer and Yoram Singer. 2002. *On the algorithmic implementation of multiclass kernel-based vector machines*. J. Mach. Learn. Res. 2 (March 2002), 265-292.)