

Support Vector Machines

Classifier Feature vector / input: $x \in \mathbb{R}^n$
 label / output: $y \in \{a_1, a_2, \dots, a_m\}$

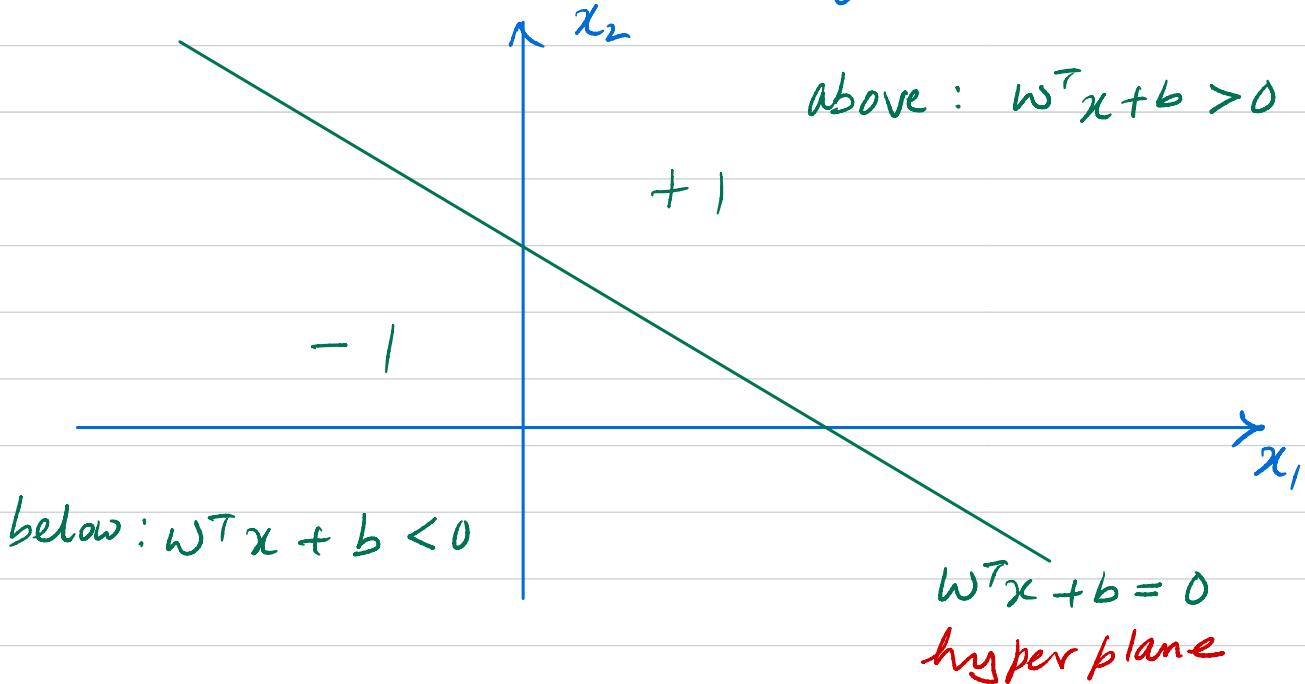
Given labelled training data $\{x_i, y_i\}_{i=1}^N$,
 build a "classifier" to predict label y
 for new input x .

Special Case : Binary Linear classifier

$$y \in \{+1, -1\}.$$

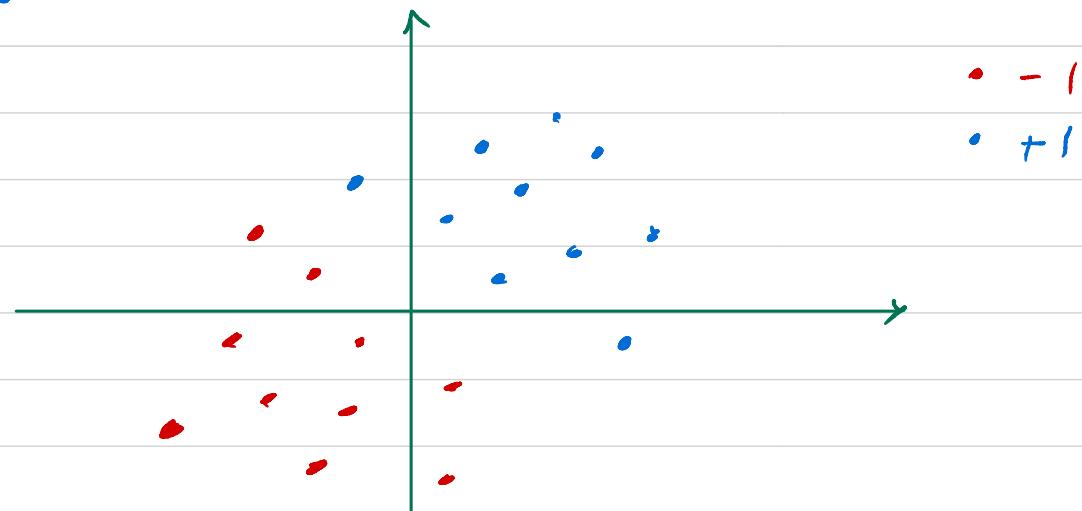
E.g., x is vector representing image of cat (+1)
 or dog (-1).

A linear classifier constructs a linear boundary
 in feature space to classify

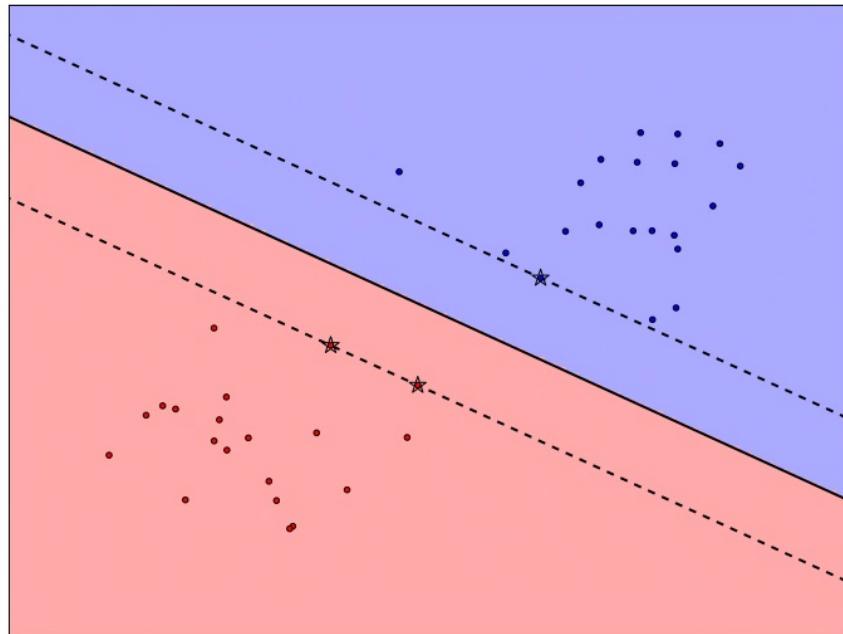


How to construct linear boundary from training data?

Linearly Separable Case

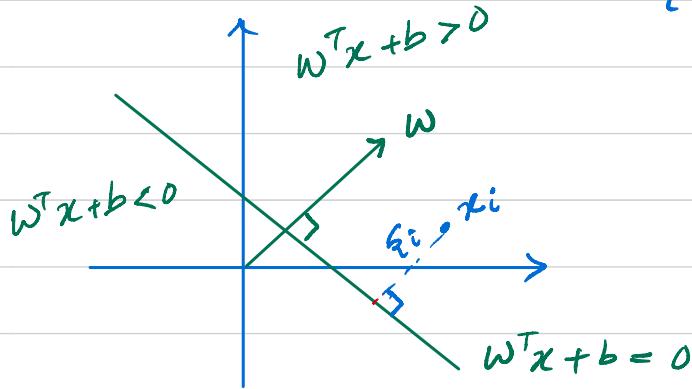


Ininitely many hyperplanes separating classes



SVM chooses hyperplane to maximize the margin
between training data from two classes.

ξ_i = distance of x_i to hyperplane



w is \perp to hyperplane. Why?

Any vector in the direction of hyperplane can be written as $x_1 - x_2$, where x_1 and x_2 satisfy $w^T x_1 + b = 0 = w^T x_2 + b$
i.e. $w^T (x_1 - x_2) = 0$

Thus $\frac{w}{\|w\|}$ is a unit vector \perp to hyperplane

$w^T x_i + b > 0 \Rightarrow x_i - \xi_i \frac{w}{\|w\|}$ is a point on hyperplane

$$\begin{aligned} \Rightarrow w^T \left(x_i - \xi_i \frac{w}{\|w\|} \right) + b &= 0 && \text{Similarly if } \\ \Rightarrow w^T x_i - \xi_i \|w\| + b &= 0 && w^T x_i + b < 0 \\ \Rightarrow \xi_i &= \frac{|w^T x_i + b|}{\|w\|} && \xi_i = -\frac{(w^T x_i + b)}{\|w\|} \end{aligned}$$

Alternatively, $\xi_i^2 = \min_{\bar{x}} \|x_i - \bar{x}\|^2$) convex opt. problem
s.t. $w^T \bar{x} + b = 0$

$$L(\bar{x}, \lambda) = \|\bar{x} - x_i\|^2 + \lambda (w^T \bar{x} + b)$$

$$\nabla_{\bar{x}} L(\bar{x}, \lambda) = 2(\bar{x} - x_i) + \lambda w = 0$$

$$\Rightarrow \bar{x}^* = -\frac{\lambda w}{2} + x_i \quad (\text{and } w^T \bar{x}^* = -b)$$

$$\text{Solve to get } \lambda = 2 \left(\frac{w^T x_i + b}{\|w\|^2} \right)$$

$$\xi_i = \|\bar{x}^* - x_i\| = \frac{|\lambda \|w\||}{2} = \frac{|w^T x_i + b|}{\|w\|}$$

$$y_i = +1 \equiv w^T x_i + b > 0$$

$$y_i = -1 \equiv w^T x_i + b < 0$$

Thus, $\xi_i = \frac{|w^T x_i + b|}{\|w\|} = \frac{y_i (w^T x_i + b)}{\|w\|}$

Our goal is to make ξ_i large for all i , i.e.

to maximize $\min_{w,b} \xi_i$.

Rewrite this as: $\underset{w,b,\Sigma}{\text{maximize}} \quad \xi$

$$\text{s.t. } \frac{y_i (w^T x_i + b)}{\|w\|} \geq \xi \quad \forall i$$

Define $\beta = \xi \|w\|$. Then opt. becomes

$$\underset{w,b,\beta}{\text{maximize}} \quad \frac{\beta}{\|w\|}$$

$$\text{s.t. } y_i (w^T x_i + b) \geq \beta \quad \forall i$$

Define $\tilde{w} = \frac{w}{\beta}$ and $\tilde{b} = \frac{b}{\beta}$. Then opt. becomes

$$\underset{\tilde{w}, b, \beta}{\text{maximize}} \quad \frac{1}{\|\tilde{w}\|}$$

$$\text{s.t. } y_i (\tilde{w}^T x_i + \tilde{b}) \geq 1 \quad \forall i \quad) \begin{matrix} \text{no} \\ \beta \end{matrix} \text{ here}$$

$$\begin{matrix} \Leftrightarrow \\ \downarrow \\ \text{drop "}" \end{matrix} \quad \begin{matrix} \text{minimize} \\ w, b \end{matrix} \quad \begin{matrix} \|w\| \\ \text{s.t. } y_i (w^T x_i + b) \geq 1 \quad \forall i \end{matrix}$$

Replace $\|w\|$ by $\frac{1}{2} \|w\|^2$ to get quadratic program

$$\text{SVM optimization: } \min_{w, b} \frac{1}{2} \|w\|^2$$

$$\text{s.t. } y_i (w^T x_i + b) \geq 1 \quad \forall i$$

(x_i, y_i) where constraint is active are called
Support vectors

Dual of SVM Optimization Problem

Can easily check that Slater's condition for strong duality holds.

$$L(w, b, \mu) = \frac{1}{2} \|w\|^2 + \sum_{i=1}^N \mu_i (1 - y_i (w^T x_i + b))$$

$$\nabla_w L(w, b, \mu) = 0 \Rightarrow w = \sum_{i=1}^N \mu_i y_i x_i$$

$$\nabla_b L(w, b, \mu) = 0 \Rightarrow \sum_{i=1}^N \mu_i y_i = 0$$

$$\begin{aligned} \Rightarrow D(\mu) &= \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \mu_i \mu_j y_i y_j (x_i^T x_j) + \sum_{i=1}^N \mu_i \\ &\quad - \sum_{i=1}^N \sum_{j=1}^N \mu_i y_i \mu_j y_j x_j^T x_i \\ &= \sum_{i=1}^N \mu_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \mu_i \mu_j y_i y_j (x_i^T x_j) \end{aligned}$$

Dual problem is : $\max D(\mu) \leftarrow \text{quadratic}$
 s.t. $\mu \geq 0, \sum \mu_i y_i = 0 \quad \text{program}$

If μ^* is solution to dual :

$$w^* = \sum_{i=1}^N \mu_i^* y_i x_i \leftarrow \text{Note: Typically very few non-zero } \mu_i^* \text{'s}$$

How about b^* ?

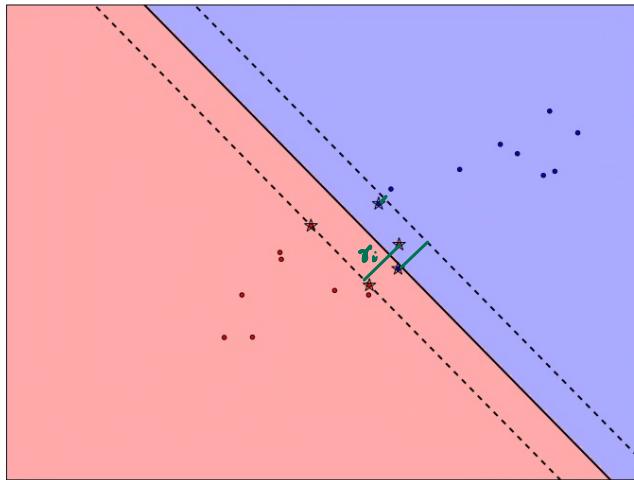
Complementary slackness $\Rightarrow \mu_i^* (1 - y_i (\omega^T x_i + b^*)) = 0$

So for any i s.t. $\mu_i^* > 0$, we have

$$1 - y_i (\omega^T x_i + b^*) = 0 \quad | \quad x_i \text{ is a support vector}$$

$$\Rightarrow b^* = \frac{1}{y_i} - \omega^T x_i$$

What if data are not linearly separable?



Allow for violation of margin: $\tau_i = \text{margin violation}$

$\tau_i \geq 0$. Points on "correct" side of margin have $\tau_i = 0$.

$$\underset{\omega, b, \tau_i}{\text{minimize}} \quad \frac{1}{2} \|\omega\|^2 + C \sum_{i=1}^N \tau_i$$

controls margin violation

$$y_i (\omega^T x_i + b) \geq 1 - \tau_i, \quad \forall i$$

$$\tau_i \geq 0 \quad \forall i$$

Dual: $\underset{\mu}{\text{maximize}} \quad D(\mu) \leftarrow \text{same as before}$

$$\text{s.t. } \sum \mu_i y_i = 0$$

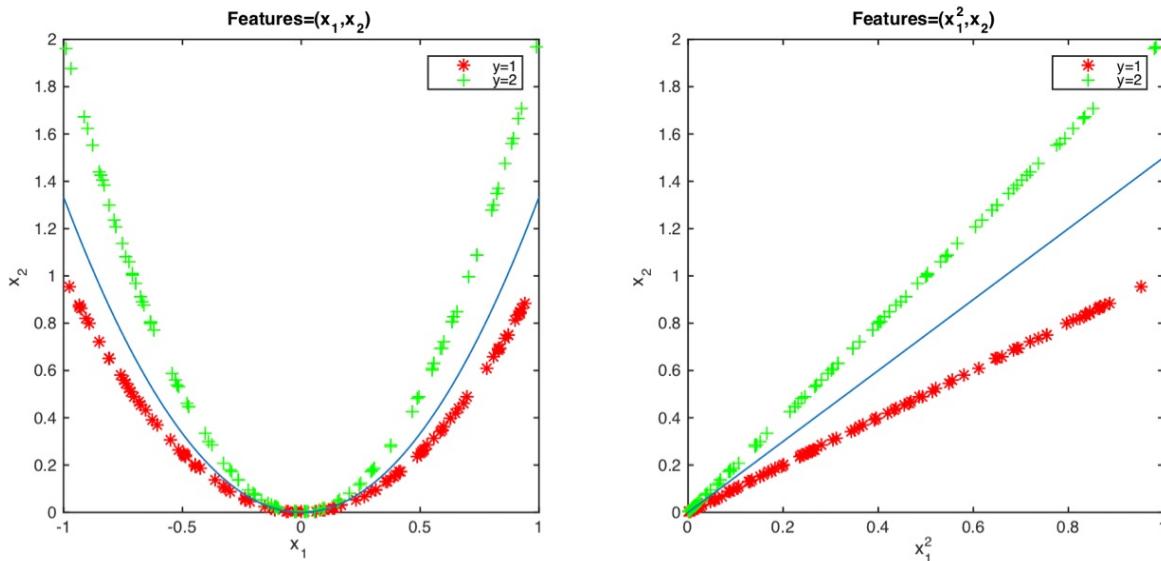
$$0 \leq \mu_i \leq C, \quad \forall i$$

Kernel Trick and SVM

Map feature vector to new space



Why? Easier to separate training data using linear boundaries in new space



Dot Products and Kernels

In new space $\phi(x)^T \phi(r) \stackrel{\Delta}{=} k(x, r)$
↖ kernel

So if classifier design and implementation only require dot products, then we can use kernel to compute dot product without explicitly computing ϕ .

SVM Implementation : $\underbrace{w^T x}_\text{dot product} + b \geq 0$

SVM design also only involves dot products if we use dual!

$$D(\mu) = \sum_{i=1}^N \mu_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \mu_i \mu_j y_i y_j (x_i^T x_j)$$

$\underbrace{\phantom{\sum_{i=1}^N \sum_{j=1}^N \mu_i \mu_j y_i y_j (x_i^T x_j)}}$
replace with
 $K(x_i, x_j)$

Mercer Kernel

- Properties :
- 1) $K(\cdot, \cdot)$ is continuous
 - 2) $K(x, r) = K(r, x)$
 - 3) $K(\cdot, \cdot)$ is positive semi-definite

For any Mercer Kernel, we can show that $\exists \phi$
s.t. $K(x, r) = \phi(x)^T \phi(r)$.

- Examples
1. Linear: $K(x, r) = x^T r$ (no kernel)
 2. Radial basis function (RBF): $K(x, r) = h(\|x-r\|)$
e.g. Gaussian RBF:

$$K(x, r) = e^{-\beta \|x-r\|^2}, \beta > 0$$
 monotonic ↑
 3. Polynomial: $K(x, r) = (1 + x^T r)^l, l=1, 2, \dots$

