Support Vector Machines

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

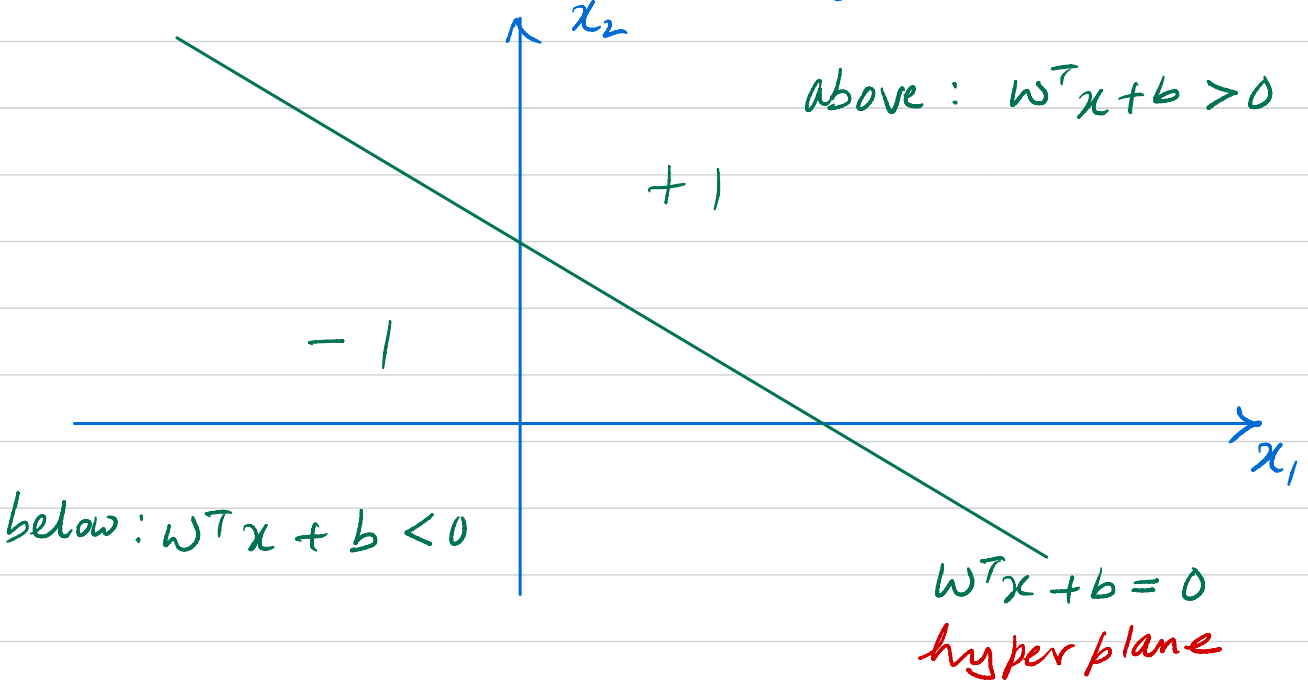
Given labelled training data $\{x_i, y_i\}_{i=1}^{N}$
build a "classifier" to predict label y for new input x.

Special Case: Binary Linear classifier

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

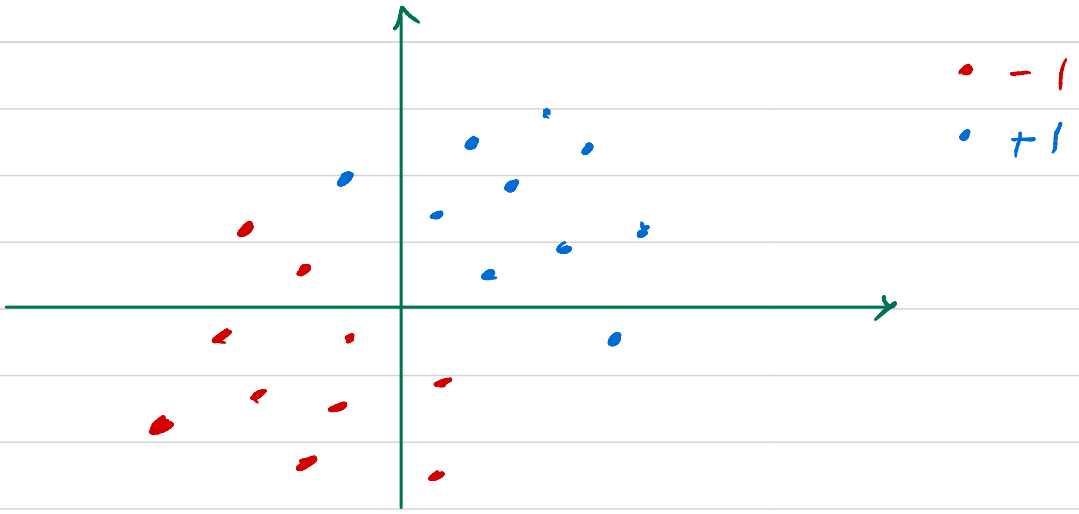
E.g., x is vector representing image of cat (+1) or dog (-1).

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

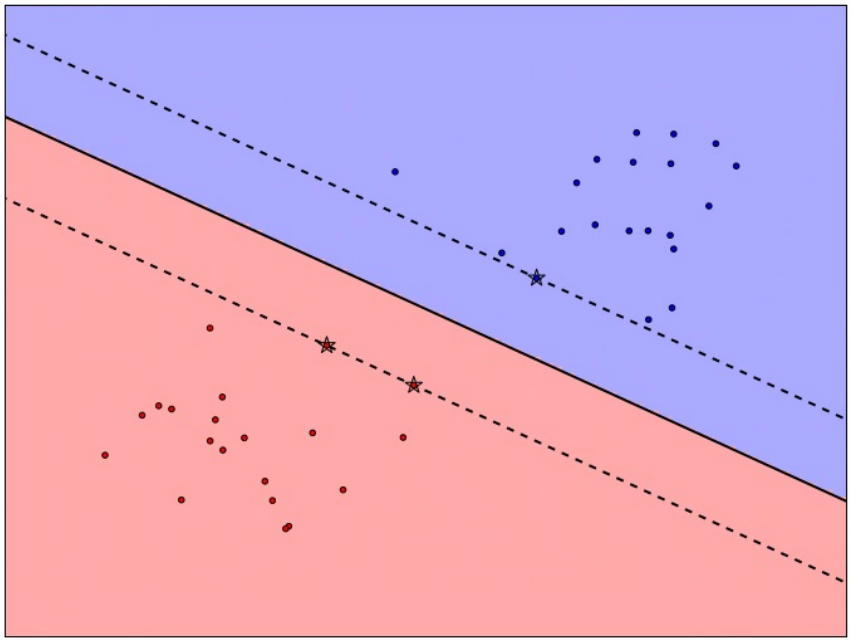


How to construct linear boundary from training data?

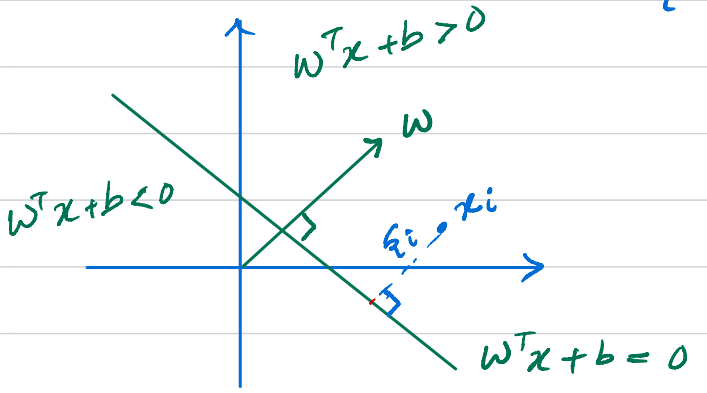
Linearly Separable Case



Infinitely 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

$$\Rightarrow w^T \left(x_i - \xi_i \frac{w}{\|w\|} \right) + b = 0$$

$$\Rightarrow w^T x_i - \xi_i \|w\| + b = 0$$

$$\Rightarrow \xi_i = \frac{|w^T x_i + b|}{\|w\|}$$

Similarly if $w^T x_i + b < 0$

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

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

$$L(z, \lambda) = \|z - x_i\|^2 + \lambda (w^T z + b)$$

$$\nabla_z L(z, \lambda) = 2(z^* - x_i) + \lambda w = 0$$

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

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

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

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

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

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

Our goal is to make ξ_i large for all i, i.e. to $\underset{w,b}{\text{maximize}} \; \min_i \xi_i$.

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

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

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

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

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

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

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

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

no β here

$\Longleftrightarrow$ (drop "$\sim$")

$$\underset{w,b}{\text{minimize}} \; \|w\|$$

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

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

SVM optimization:

$$\min_{w,b} \ \frac{1}{2}\|w\|^2$$

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

(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.

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

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

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

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

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

Dual problem is: $\max D(\mu)$ ← quadratic program

$$\text{s.t. } \mu \geq 0; \ \textstyle\sum \mu_i y_i = 0$$

If μ^* is solution to dual:

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

How about b^*?

Complementary slackness $\Rightarrow$ $\mu_i^* \left(1 - y_i (w^{*T} x_i + b^*)\right) = 0$

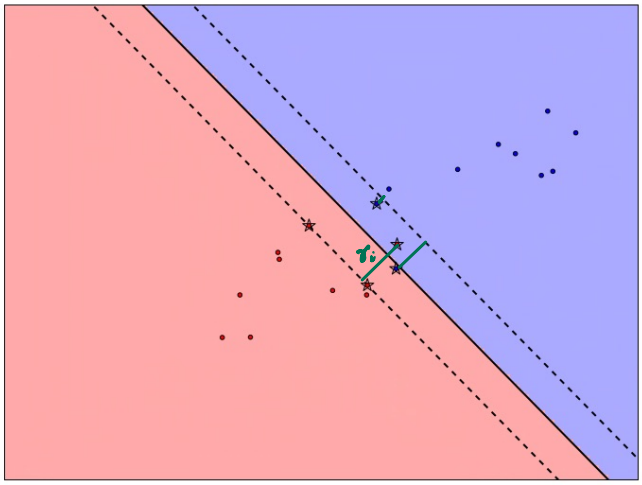
So for any i s.t. $\mu_i^* > 0$, we have

$$1 - y_i (w^{*T} x_i + b^*) = 0$$

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

x_i is a support vector

What if data are not linearly separable?



Allow for violation of margin: γ_i = margin violation

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

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

C controls margin violation

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

$$\gamma_i \geq 0 \quad \forall i$$

Dual: maximize $D(\mu)$ ← same as before

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

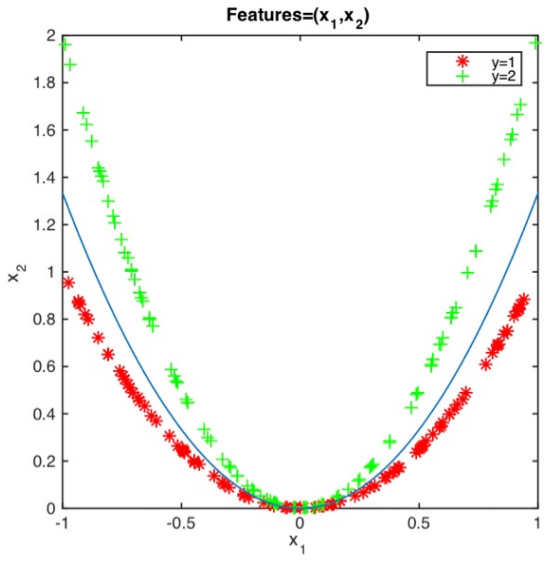
$$0 \leq \mu_i \leq C, \quad \forall i$$

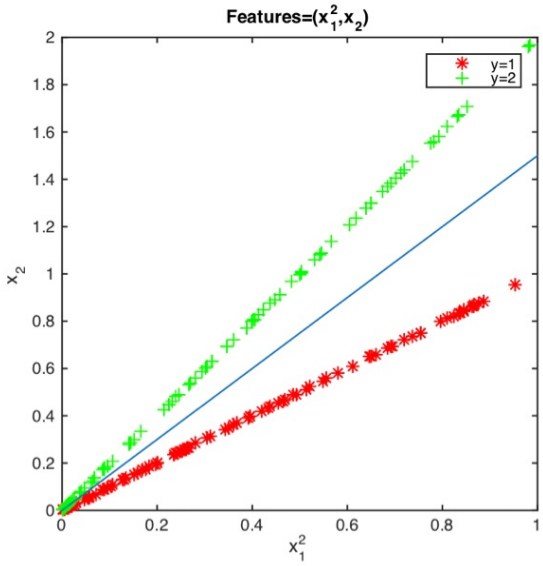
Kernel Trick and SVM

Map feature vector to new space

x (n-dim) $\xrightarrow{\phi(\cdot)\ \text{non-linear}}$ $\phi(x)$ (typically higher dimension than x)

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





Dot Products and Kernels

$x^T r$

In new space $\phi(x)^T \phi(r) \triangleq \kappa(x, r)$ ← 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: $w^{*T} x + b^* \gtrless 0$

$w^{*T} x$: dot product, replace with $\kappa(w^*, x)$

SVM design also only involves dot products of training inputs $\{x_i\}$ if we use dual!

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

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): $k(x, r) = h(\|x - r\|)$ (h monotonic)

e.g. Gaussian RBF:

$$k(x, r) = e^{-\beta \|x - r\|^2}, \quad \beta > 0$$

3. Polynomial: $k(x, r) = (1 + x^T r)^l, \quad l = 1, 2, \ldots$