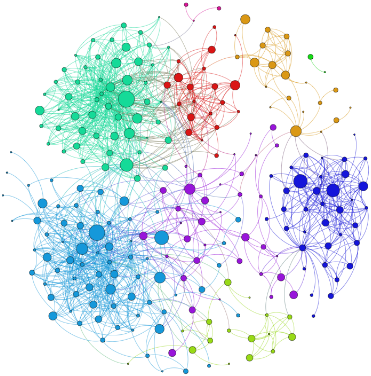
Classification

Jiaming Mao

Xiamen University



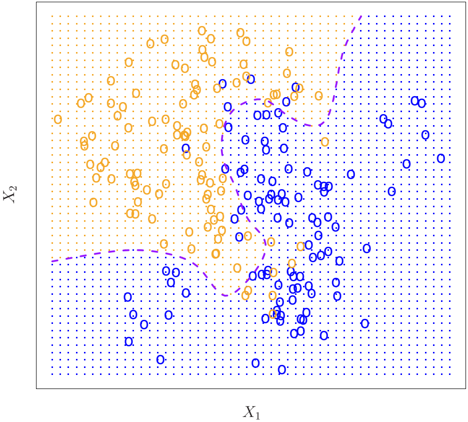
This version: Fall 2018

Contact: jmao@xmu.edu.cn

Course homepage: [under construction]

Separating Hyperplanes

In logistic regression, we first estimate $p(y|x)$ and then use $\widehat{p}(y|x)$ to derive the decision boundaries that classify y. An alternative approach is to estimate the decision boundaries directly.



Separating Hyperplanes

- A linear decision boundary can be expressed as:

$$b + w'x = 0 \tag{1}$$

, where $x = (x_1, \ldots, x_p)$[1].

- (1) is called a **hyperplane**. A hyperplane in p dimensions is a flat affine subspace of dimension $p - 1$.
  - In $p = 2$ dimensions, a hyperplane is a line. In $p = 3$ dimensions, a hyperplane is a plane.
- Given a hyperplane (1), the two sets of points $\{x : b + w'x > 0\}$ and $\{x : b + w'x < 0\}$ lie respectively on the two sides of the hyperplane. We can think of the hyperplane as dividing $p-$dimensional space into two halves.

[1] Equivalently, a hyperplane can be expressed as $\beta' x = 0$, where $\beta = (\beta_0, \beta_1, \ldots, \beta_p)$ and $x = (1, x_1, \ldots, x_p)$. Here we let $x = (x_1, \ldots, x_p)$, $b = \beta_0$, and $w = (\beta_1, \ldots, \beta_p)$.

Separating Hyperplanes

Now consider a binary classification problem where $y \in \{-1, 1\}$. y is said to be **linearly separable** in the feature space $\mathcal{X}$ if there exists a hyperplane $b + w'x = 0$ that can perfectly separates $y = 1$ from $y = -1$.

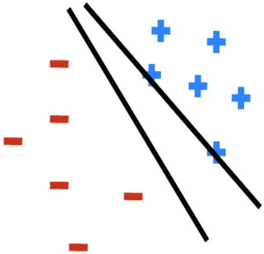
In this case, we can label $y = 1$ for $\{x : b + w'x > 0\}$ and $y = -1$ for $\{x : b + w'x < 0\}$. Thus, given a data set $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, a **separating hyperplane** has the property that

$$y_i \left(b + w'x_i\right) > 0 \quad \forall i$$

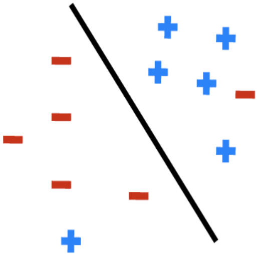
Given a separating hyperplane, we have the following classifier:

$$\widehat{y} = f(x) = \text{sign}\left(b + w'x\right)$$

Separating Hyperplanes



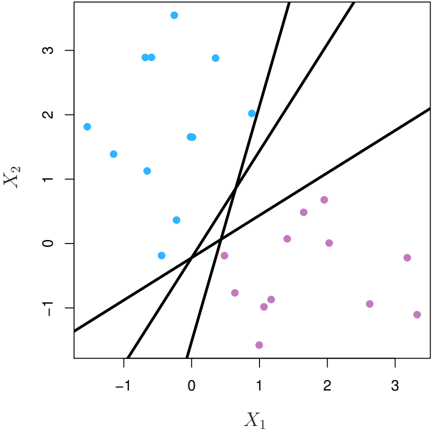
linearly separable



not linearly separable

Separating Hyperplanes

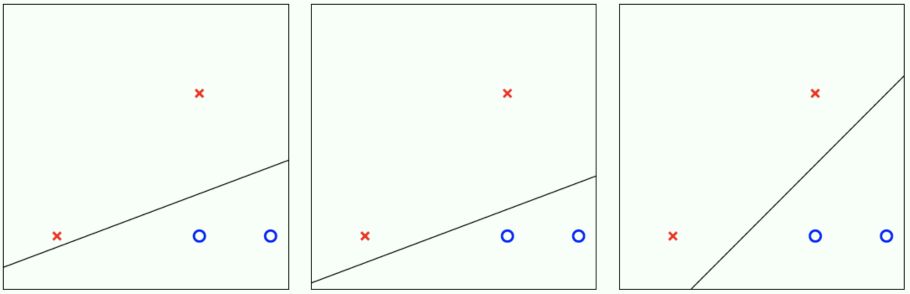
Q: if the data is separable, then there can be infinitely many separating hyperplanes. Which one should we pick?



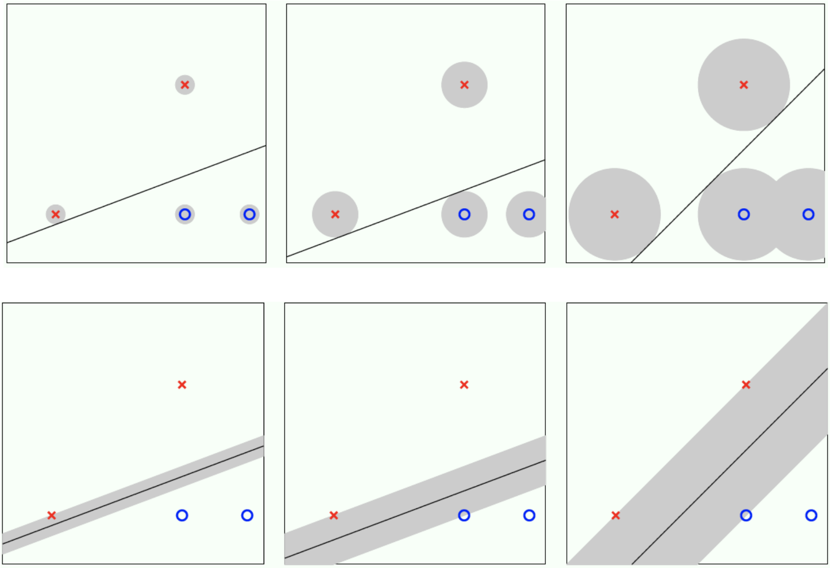
Optimal Separating Hyperplane

- Idea: find the separating hyperplane that is farthest away from any of the training data points.
- The **margin** is the distance from the hyperplane to the *nearest* data point.
- The optimal separating hyperplane is the one that maximizes the margin and is called the **maximal margin hyperplane**.
- Intuitively, large margin provides more protection against noise in the training data.
- Although each separating hyperplane perfectly separates the training data ($E_{in} = 0$), hyperplanes with larger margin have lower *variance* and hence generalize better out of sample (smaller E_{out}).

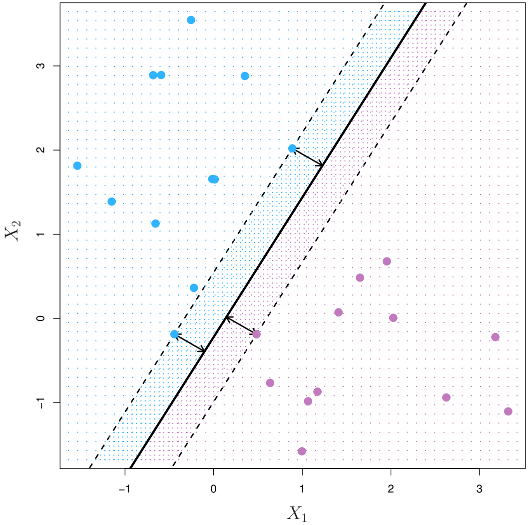
Optimal Separating Hyperplane



Optimal Separating Hyperplane



Optimal Separating Hyperplane



The Optimal Separating Hyperplane

Linear Algebra of a Hyperplane

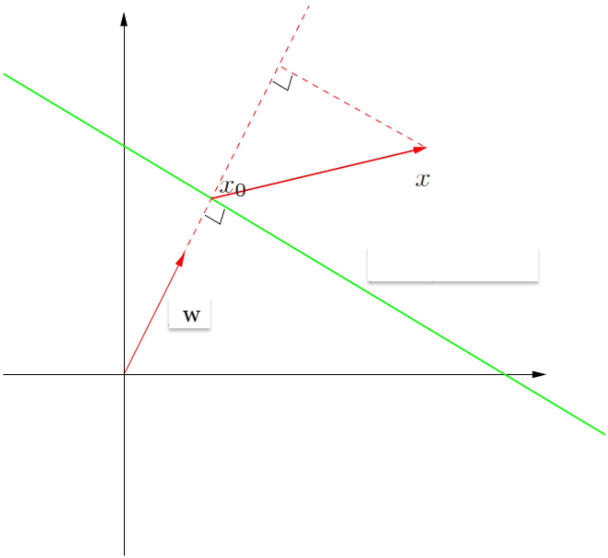
- Let $\mathbb{H}(b, w)$ be a hyperplane defined by $b + w'x = 0$. For any two points $x_1, x_2 \in \mathbb{H}$, we have $w'(x_1 - x_2) = 0$. Hence let $\widetilde{w} \equiv \frac{w}{\|w\|}$ is the unit vector *normal* to the hyperplane.

- For any point $x \notin \mathbb{H}$, the distance from x to $\mathbb{H}$ is

$$\left|\widetilde{w}'(x - x_0)\right| = \frac{1}{\|w\|}\left|b + w'x\right| \tag{2}$$

, where x_0 is any point $\in \mathbb{H}$ such that $w'x_0 = -b$.

Linear Algebra of a Hyperplane



Linear Algebra of a Hyperplane

(2) $\Rightarrow$ given a data set $\mathcal{D}$, the margin of a hyperplane $\mathbb{H}(b, w)$ is

$$\min_{i \in \{1,\ldots,N\}} \left\{ \frac{1}{\|w\|} \left| b + w' x_i \right| \right\}$$

Note that for any hyperplane $b + w'x = 0$, if we multiply (b, w) by a constant $\kappa > 0$, then the result will be the same hyperplane. Thus, we can always let $\kappa = \frac{1}{\min_i \{|b + w' x_i|\}}$, so that after rescaling, the hyperplane $\mathbb{H}(b, w)$ has the following properties:

$$\min_i \left\{ \left| b + w' x_i \right| \right\} = 1 \tag{3}$$

$$\text{margin} = \frac{1}{\|w\|} \tag{4}$$

, i.e. we can always normalize the coefficients of a hyperplane to make sure that (3) and (4) hold.

Linear Hard-Margin SVM

Therefore, given training data $\mathcal{D}$, if $\mathcal{D}$ is linearly separable, then we can find the optimal separating hyperplane by first scaling each candidate separating hyperplane so that $\min_i \{|b + w'x_i|\} = 1$ and then pick the one with the smallest $\|w\|$ (i.e. the largest margin). This is equivalent to solving the following problem:

$$\min_{b,w} \frac{1}{2} \|w\|^2 \tag{5}$$

s.t.[2]

$$y_i \left(b + w'x_i\right) \geq 1, \quad \forall i \in \{1, \ldots, N\}$$

(5) $\Rightarrow \left(\widehat{b}, \widehat{w}\right)$. The classifier $\widehat{y} = \text{sign}\left(\widehat{b} + \widehat{w}'x_i\right)$ is called the *linear* hard-margin **support vector machine** (**SVM**).

[2] If a hyperplane is separating, then $|b + w'x_i| = y_i \left(b + w'x_i\right)$.

Linear Hard-Margin SVM

(5) $\Rightarrow$ the Lagrange function is:

$$\mathbb{L} = \frac{1}{2}\left\|w\right\|^2 - \sum_{i=1}^{N} \alpha_i \left[y_i \left(b + w' x_i\right) - 1\right] \tag{6}$$

Minimizing (6) w.r.t. b and w $\Rightarrow$

$$w = \sum_i \alpha_i y_i x_i \tag{7}$$

$$0 = \sum_i \alpha_i y_i \tag{8}$$

Linear Hard-Margin SVM

Substituting (7) and (8) into (6) ⇒ the dual formulation of the SVM problem[3]:

$$\min_{\{\alpha_1,\ldots,\alpha_N\}} \left\{ \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j x_i' x_j - \sum_{i=1}^{N}\alpha_i \right\} \tag{9}$$

s.t.

$$\alpha_i \geq 0$$

$$\sum_i \alpha_i y_i = 0$$

[3](5) is called the primal formulation.

Linear Hard-Margin SVM

Solving (9) $\Rightarrow \{\widehat{\alpha}_i\}_{i=1}^N$. Then (7) $\Rightarrow$

$$\widehat{w} = \sum_i \widehat{\alpha}_i y_i x_i \tag{10}$$

In addition, according to the Karush-Kuhn-Tucker (KKT) conditions, the solution satisfies:

$$\widehat{\alpha}_i \left[y_i \left(\widehat{b} + \widehat{w}' x_i \right) - 1 \right] = 0 \tag{11}$$

, which helps us pin down $\widehat{b}$ once we have solved for $\{\widehat{\alpha}_i\}_{i=1}^N$ and $\widehat{w}$[4].

[4] For any $\widehat{\alpha}_i > 0$, (11) $\Rightarrow \widehat{b} = y_i - \widehat{w}' x_i$. Although we typically average over data points for which $\widehat{\alpha}_i > 0$ to obtain $\widehat{b} = \frac{1}{N_{\mathcal{S}}} \sum_{i \in \mathcal{S}} (y_i - \widehat{w}' x_i)$, where $\mathcal{S} = \{i : \widehat{\alpha}_i > 0\}$.

Support Vectors

In the solution to (9), only a small subset of $\widehat{\alpha}_i's$ will be nonzero. Hence $\left(\widehat{b}, \widehat{w}\right)$ only depend on a small subset of points for which $\widehat{\alpha}_i > 0$. These points are called **support vectors**.

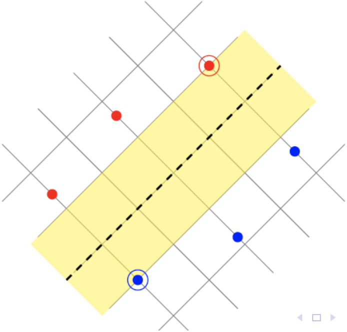
Let $\mathcal{S} = \{i : \widehat{\alpha}_i > 0\}$ denote the set of support vectors, then the SVM classifier can be expressed as

$$\begin{aligned} \widehat{y} &= \operatorname{sign}\left(\widehat{b} + \widehat{w}'x\right) \\ &= \operatorname{sign}\left(\widehat{b} + \sum_{i \in \mathcal{S}} \widehat{\alpha}_i y_i x_i' x\right) \end{aligned} \tag{12}$$

Thus, to compute the optimal hyperplane, only the support vectors are needed. This is a key property of the SVM: once the model is trained, a significant proportion of the data can be discarded and only the support vectors retained.

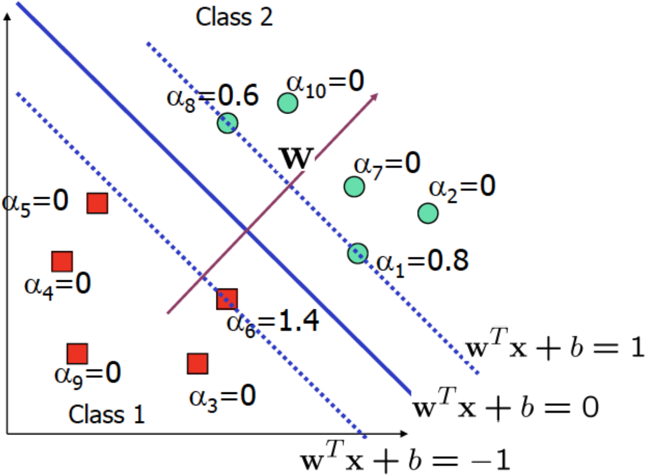
Support Vectors

Geometrically, support vectors lie on the boundary of the optimal hyperplane's margin. This can be seen from (11): for support vectors, $\widehat{\alpha}_i > 0 \Rightarrow y_i\left(\widehat{b} + \widehat{w}'x_i\right) = 1$[5]. In a sense, they "support" the margin of the optimal hyperplane and prevent it from expanding further.



[5] The reverse is not true: it is possible for points to be on the boundary but not support vectors.

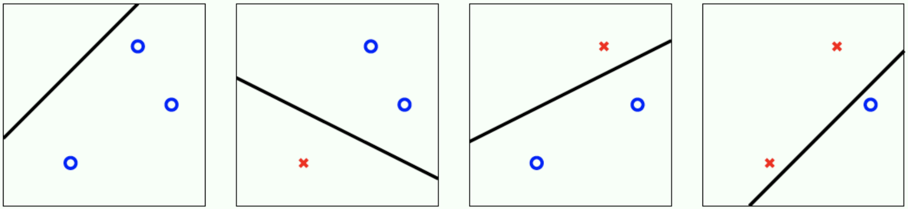
Support Vectors



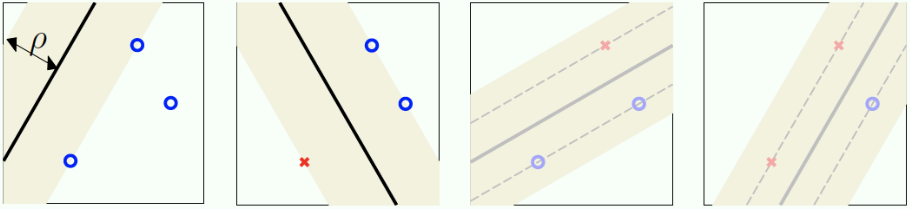
VC Analysis

- Recall that the VC dimension of a hypothesis set $\mathcal{H}$ is the size of the largest data set that $\mathcal{H}$ can shatter.
- Consider the hypothesis set $\mathcal{H}_\rho$ containing all hyperplanes of margin at least ρ.
- $\rho \uparrow \Rightarrow$ the number of points that $\mathcal{H}_\rho$ can shatter $\downarrow \Rightarrow d_{VC}(\mathcal{H}_\rho) \downarrow$

VC Analysis



Here $\mathcal{H}_0$ implements all dichotomies.



For this particular margin, $\mathcal{H}_\rho$ implements only 4 of the 8 dichotomies.

VC Analysis

VC Dimension of Separating Hyperplanes

Suppose the input space $\mathcal{X}$ is the ball of radius R in $\mathbb{R}^p$, so $\|x\| \leq R$. Then the VC dimension of a separating hyperplane in $\mathcal{X}$ with margin ρ is:

$$d_{VC}(\rho) \leq 1 + \left\lceil \frac{R^2}{\rho^2} \right\rceil \tag{13}$$

, where $\lceil x \rceil$ denotes the smallest integer greater than or equal to x.

VC Analysis

- This result establishes a crucial link between the margin and good generalization.
- The margin can be thought of as a control of *model complexity*.
- In particular, note that (13) does not explicitly depend on the dimension p of the input space[6]. Therefore, if we transform the data to a high dimensional space, as long as we are able to obtain separating hyperplanes with large enough margin, we obtain good generalization.

[6] Recall that the VC dimension of hyperplanes in $\mathbb{R}^p$ with $\rho = 0$ is at most $p + 1$. Therefore, we can use either (13) or $p + 1$, whichever is smaller, to bound the VC dimension of separating hyperplanes with margin ρ.

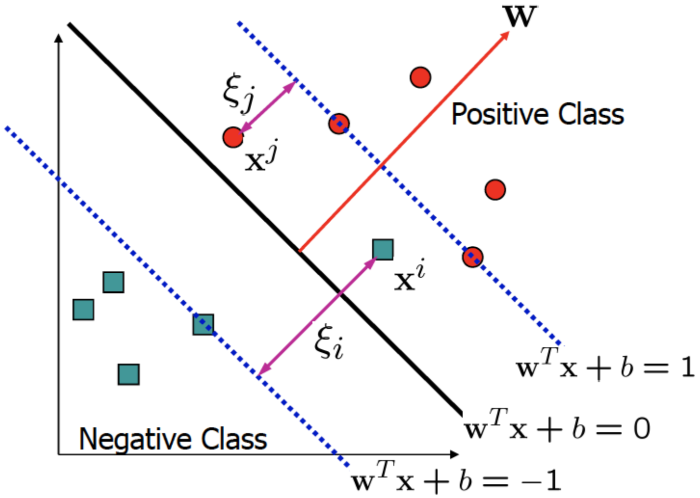
Linear Soft-Margin SVM

- What happens when the data are *not* linearly separable? We can still use a linear classifier, but we are going to have misclassifications.

- Introduce the **slack variables** $\xi_i \geq 0$: ξ_i measures the amount of **margin violations**.

  - $\xi_i = 0$: (x_i, y_i) is correctly classified and resides outside the margin.
  - $\xi_i \in (0, 1]$: (x_i, y_i) is correctly classified but resides *inside* the margin.
  - $\xi_i > 1$: (x_i, y_i) is misclassified.
  - $\sum_{i=1}^{N} \xi_i$ is an upperbound on the number of misclassified points.

Linear Soft-Margin SVM



Linear Soft-Margin SVM

To maximize the margin while controlling for classification error, we solve the following problem:

$$\min_{b,w,\xi}\left\{\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i\right\} \tag{14}$$

s.t.

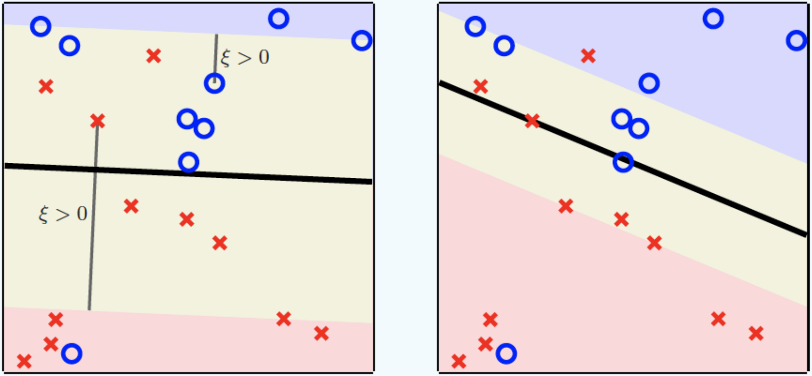
$$\xi_i \geq 0, \quad y_i\left(b + w'x_i\right) \geq 1 - \xi_i, \quad \forall i \in \{1, \ldots, N\}$$

(14) is called the *linear* soft-margin **support vector machine**.

Linear Soft-Margin SVM

- In (14), we have the dual objectives of maximizing the margin and minimizing the amount of margin violations. The parameter C controls the tradeoff between the two objectives:
  - $C \uparrow$: less tolerant of margin violations $\Rightarrow$ narrower margin
  - $C \downarrow$: more tolerant of margin violations $\Rightarrow$ wider margin
  - $C = 0$: ignores the data entirely.
  - $C \to \infty$: the data *have* to be separable (back to hard-margin SVM)
- Equivalently, C can be thought of as controlling the tradeoff between model complexity and in-sample fit, or, the bias-variance tradeoff:
  - $C \uparrow \Rightarrow$ bias $\downarrow$, variance $\uparrow$
  - $C \downarrow \Rightarrow$ bias $\uparrow$, variance $\downarrow$

Linear Soft-Margin SVM



(a) $C = 1$ (b) $C = 500$

Linear Soft-Margin SVM

(14) $\Rightarrow$ the Lagrange function is:

$$\mathbb{L} = \frac{1}{2}\left\|w\right\|^2 + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\alpha_i\left[y_i\left(b + w'x_i\right) - (1 - \xi_i)\right] - \sum_{i=1}^{N}\mu_i\xi_i \quad (15)$$

Minimizing (15) w.r.t. $b, w, \xi_i \Rightarrow$

$$w = \sum_i \alpha_i y_i x_i \quad (16)$$

$$0 = \sum_i \alpha_i y_i \quad (17)$$

$$\alpha_i = C - \mu_i,\ \forall i \quad (18)$$

Linear Soft-Margin SVM

Substituting (16)− (18) into (15) ⇒ the dual problem:

$$\min_{\{\alpha_1,\dots,\alpha_N\}} \left\{ \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j x_i' x_j - \sum_{i=1}^{N} \alpha_i \right\} \tag{19}$$

s.t.

$$0 \le \alpha_i \le C, \quad \sum_i \alpha_i y_i = 0$$

Linear Soft-Margin SVM

Solving (19) $\Rightarrow \{\widehat{\alpha}_i\}_{i=1}^N$. Then (16) $\Rightarrow \widehat{w} = \sum_i \widehat{\alpha}_i y_i x_i$. In addition, according to the KKT conditions:

$$\widehat{\mu}_i \widehat{\xi}_i = \left(C - \widehat{\alpha}_i\right) \widehat{\xi}_i = 0 \tag{20}$$

$$\widehat{\alpha}_i \left[y_i \left(\widehat{b} + \widehat{w}' x_i \right) - \left(1 - \widehat{\xi}_i \right) \right] = 0 \tag{21}$$

, which helps us pin down $\left\{\widehat{\xi}_i\right\}_{i=1}^N$ and $\widehat{b}$ once we have solved for $\{\widehat{\alpha}_i\}_{i=1}^N$ and $\widehat{w}$[7].

[7] For $\widehat{\alpha}_i \in [0, C)$, (20) $\Rightarrow \widehat{\xi}_i = 0$. Therefore, we can obtain $\widehat{b}$ from (21) for $\widehat{\alpha}_i \in (0, C)$. Once we have $\widehat{b}$, (21) allows us to calculate $\widehat{\xi}_i$ for any $\widehat{\alpha}_i = C$.

Linear Soft-Margin SVM

The SVM classifier is:

$$\widehat{y} = \text{sign}\left(\widehat{b} + \widehat{w}'x\right) = \text{sign}\left(\widehat{b} + \sum_{i \in \mathcal{S}} \widehat{\alpha}_i y_i x_i' x\right) \tag{22}$$

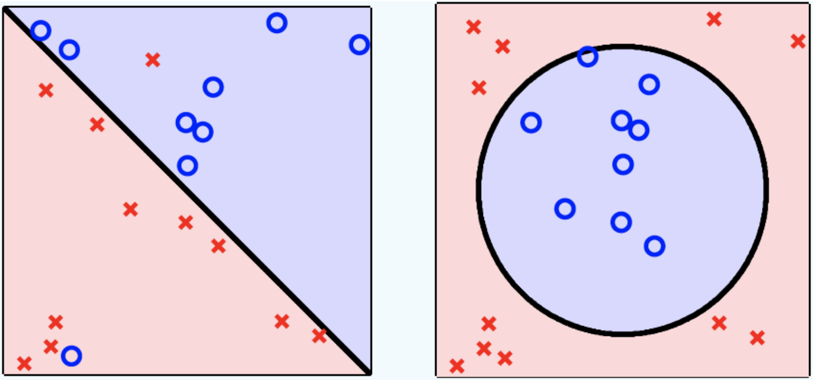
, where $\mathcal{S} = \{i : \widehat{\alpha}_i > 0\}$ is the set of support vectors.

For soft-margin SVM, there are two kinds of support vectors:

- **margin support vectors**: $\widehat{\alpha}_i \in (0, C)$. These points lie on the edge of the margin ($\widehat{\xi}_i = 0,\ y_i\left(\widehat{b} + \widehat{w}'x_i\right) = 1$).
- **non-margin support vectors**: $\widehat{\alpha}_i = C$. These are the points that violate the margin ($\widehat{\xi}_i > 0$)[8].

[8] Note that not all points that lie on the edge of the margin are necessarily margin support vectors. However, *all* points that violate the margin are non-margin support vectors.

Nonlinear SVM



(a) Few noisy data. (b) Nonlinearly separable.

Nonlinear SVM

To construct nonlinear decision boundaries, we apply a feature transform $\Phi : \mathcal{X} \to \mathcal{Z}$ and solve problem (14) with $z = \Phi(x)$ in place of x.

This gives us the following dual problem[9]:

$$\min_{\{\alpha_1,\ldots,\alpha_N\}} \left\{ \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j z_i' z_j - \sum_{i=1}^{N} \alpha_i \right\} \tag{23}$$

s.t.

$$0 \leq \alpha_i \leq C, \quad \sum_i \alpha_i y_i = 0$$

Solving the problem gives us the classifier:

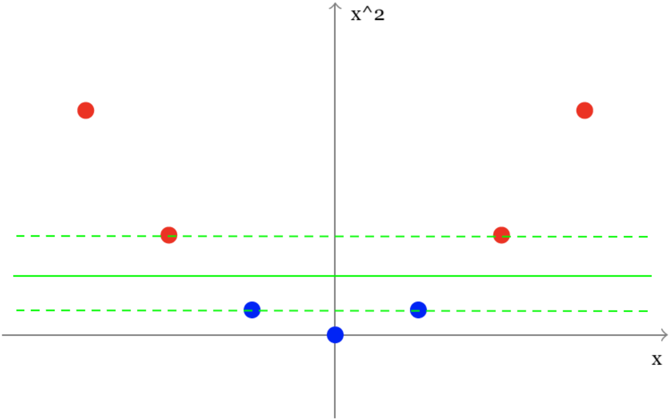
$$\widehat{y} = \text{sign}\left(\widehat{b} + \sum_{i \in \mathcal{S}} \widehat{\alpha}_i y_i z_i' z \right) \tag{24}$$

[9] This is a nonlinear soft-margin SVM.

Nonlinear SVM

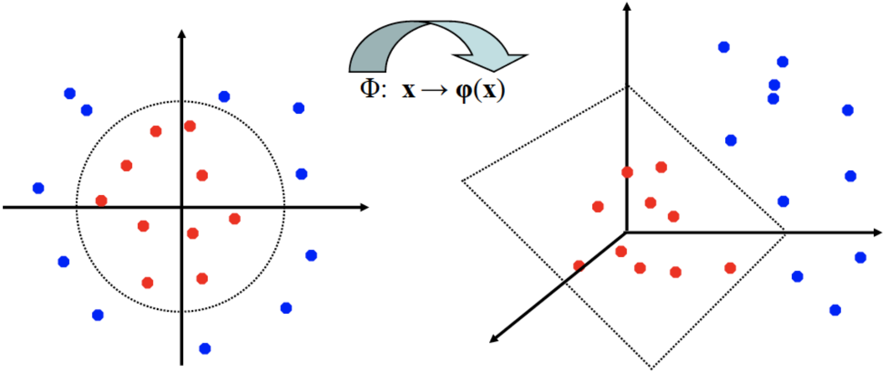
- The following sample in $\mathbb{R}$ is not linearly separable:

- But if we map $x \to (x, x^2)$, it becomes linearly separable:



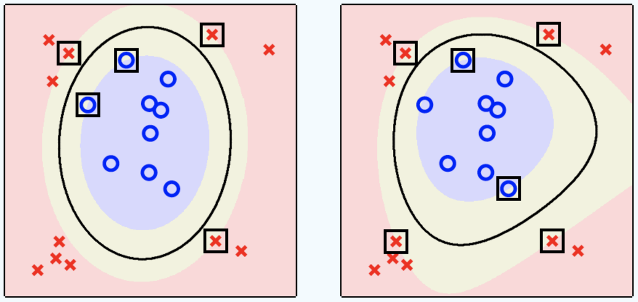
Nonlinear SVM

The original input space can be transformed to some higher-dimensional feature space such that the data is linearly separable:



Nonlinear SVM

The SVM has a potential robustness to overfitting even after transforming to a much higher dimension.



Left: 2nd order polynomial transform (Φ_2); Right: 3rd order polynomial transform (Φ_3); Note that the dimension of Φ_3 is nearly double that of Φ_2, yet SVM with Φ_3 is not severely overfitting[10].

[10]It can be proved that $\frac{1}{N}$(# support vectors) provides an upperbound for an unbiased estimate of E_{out} . Here, the number of support vectors (boxed) only increases from 5 to 6 when Φ_3 is used instead of Φ_2.

The Kernel Trick

- Notice that (23) and (24) depend on z only through inner products of the type $z_i' z_j$.

- We can replace $z_i' z_j$ with a **kernel function** $K(x_i, x_j)$ that effectively computes $z_i' z_j = \Phi'(x_i)\,\Phi(x_j)$ without the need to transform $\{x_i, x_j\}$ into $\{z_i, z_j\}$ first – this is called "**the kernel trick**."

- Using the kernel trick is more *computationally efficient*: instead of applying a feature transform $\Phi : \mathcal{X} \to \mathcal{Z}$ and calculating inner products in $\mathcal{Z}$ space, we choose an appropriate kernel that *implicitly* maps x to a higher dimensional space, while taking less time to compute.

The Kernel Trick

Using the kernel trick, the dual problem becomes:

$$\min_{\{\alpha_1,\ldots,\alpha_N\}} \left\{ \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j K\left(x_i, x_j\right) - \sum_{i=1}^{N} \alpha_i \right\} \tag{25}$$

s.t.

$$0 \leq \alpha_i \leq C, \quad \sum_i \alpha_i y_i = 0$$

Final hypothesis:

$$\widehat{y} = \text{sign}\left(\widehat{b} + \sum_{i \in \mathcal{S}} \widehat{\alpha}_i y_i K\left(x_i, x\right)\right) \tag{26}$$

Polynomial Kernel

$k-$**Degree Polynomial Kernel**:

$$K(u,v) = (1+u'v)^k$$

, where u, v are vectors in $p-$dimensional space.

Quadratic Kernel

When $p = 2$, $k = 2$,

$$\begin{aligned} K(u,v) &= (1+u'v)^2 \\ &= (1+u_1v_1+u_2v_2)^2 \\ &= 1+2u_1v_1+2u_2v_2+(u_1v_1)^2+(u_2v_2)^2+2u_1v_1u_2v_2 \\ &= \Phi(u)'\Phi(v) \end{aligned}$$

, where $\Phi(u) = \left(1, \sqrt{2}u_1, \sqrt{2}u_2, u_1^2, u_2^2, \sqrt{2}u_1u_2\right)$.

Polynomial Kernel

Quadratic Kernel (cont.)

- Thus, using the quadratic kernel is equivalent to applying the feature transform Φ and computing the inner product $\Phi(u)' \Phi(v)$.
- Computing time is $\mathcal{O}(p)$ for $k-$degree polynomial kernels in $p-$dimensional space, as opposed to $\mathcal{O}\left(p^k\right)$ for calculating inner products in the corresponding transformed feature spaces.

Gaussian Kernel[11]:

$$K(u, v) = \exp\left(-\gamma \|u - v\|^2\right)$$

- The corresponding feature transform is *infinite-dimensional*.

[11] Also called **radial basis function** (**RBF**) **kernel**.

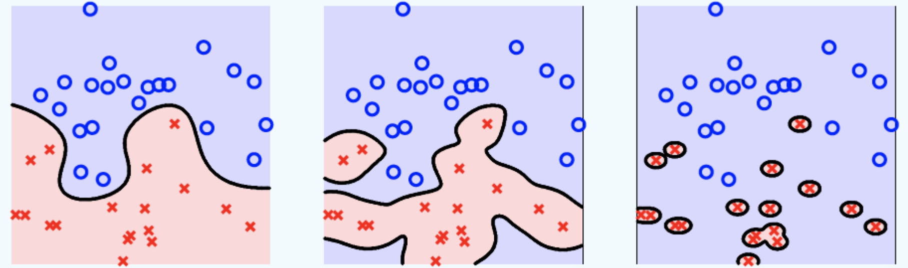
Gaussian Kernel

When $p = 1$, $\gamma = 1$,

$$\begin{aligned} K(u, v) &= e^{-(u-v)^2} \\ &= e^{-u^2} \left(\sum_{k=0}^{\infty} \frac{2^k u^k v^k}{k!} \right) e^{-v^2} \\ &= \Phi(u)' \Phi(v) \end{aligned}$$

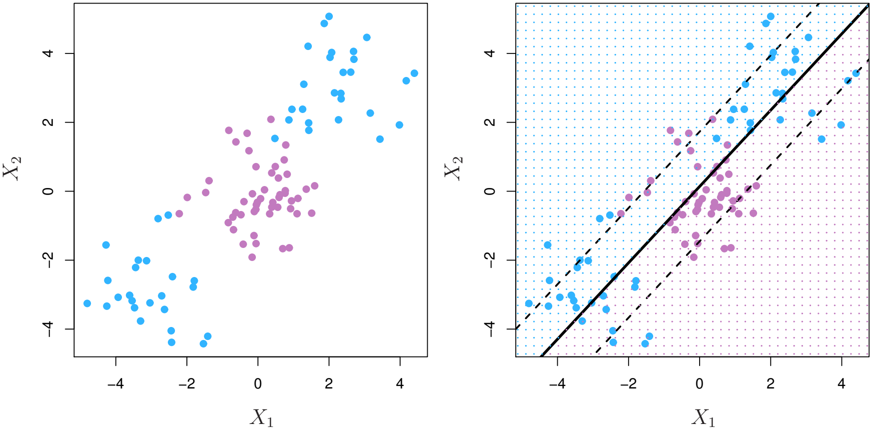
, where $\Phi(u) = e^{-u^2} \left(1, \sqrt{2}u, \frac{\sqrt{2^2}}{2!} u, \frac{\sqrt{2^3}}{3!} u^3, \ldots \right)$.

Gaussian Kernel

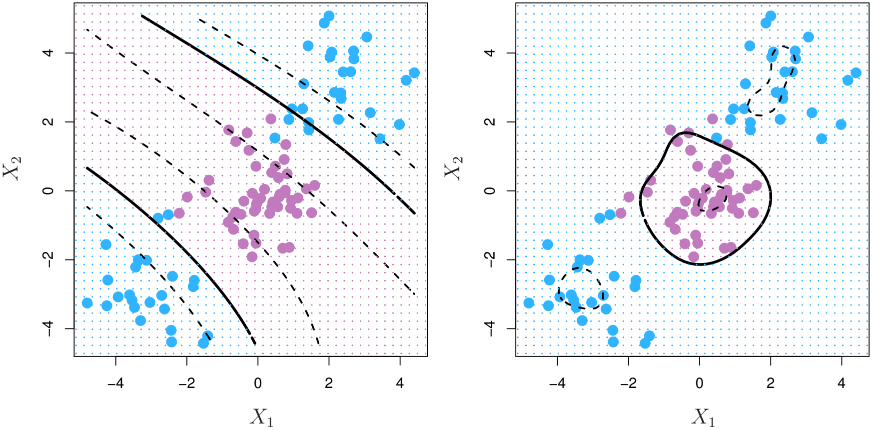


$\exp(-1\|\mathbf{x}-\mathbf{x}'\|^2)$ $\exp(-10\|\mathbf{x}-\mathbf{x}'\|^2)$ $\exp(-100\|\mathbf{x}-\mathbf{x}'\|^2)$

Linear SVM



Nonlinear SVM



Left: polynomial kernel ($k = 3$); Right: gaussian kernel ($\gamma = 1$)

Multiclass SVM

SVM can be extended to solving multiclass problems with J classes.

- *One-versus-one (all-pairs) classification*: construct $\binom{J}{2}$ pairwise SVMs, each comparing a class j vs. another class k. Assign an observation to the class that wins the most pairwise competitions.
- *One-versus-all classification*: fit J SVMs, each compares a class j (coded as $+1$) vs. the remaining $J-1$ classes (coded as -1). Let $\widehat{g}_j(x) = \widehat{b}^j + \sum_i \widehat{\alpha}_i^j y_i K(x_i, x)$[12]. Then given an observation x_0, let $\widehat{y}_0 = \arg\max_j \{\widehat{g}_j(x_0)\}$.

[12] For linear SVM, $K(u, v) = u'v$.