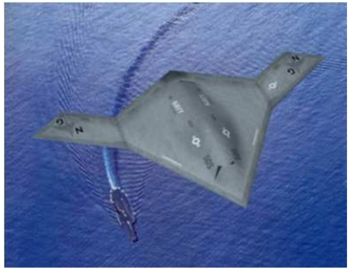
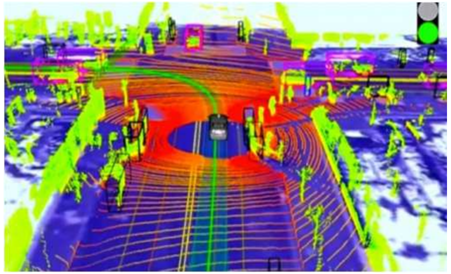
Intelligent Image and Graphics Processing

智能图像图形处理

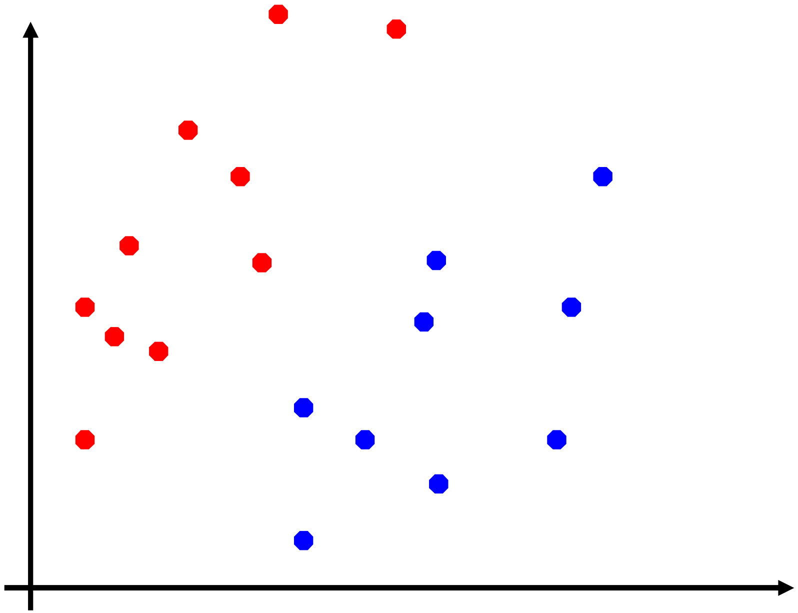
布树辉 bushuhui@nwpu.edu.cn
http://www.adv-ci.com





Support Vector Machine

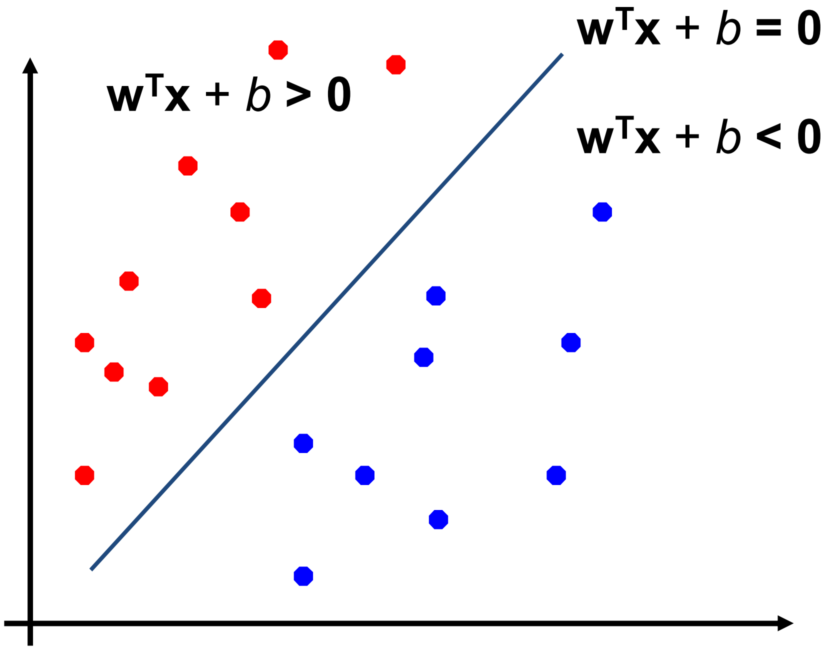
Problem



How to find a model to separate two types data?

Perceptron Revisited: Linear Separators

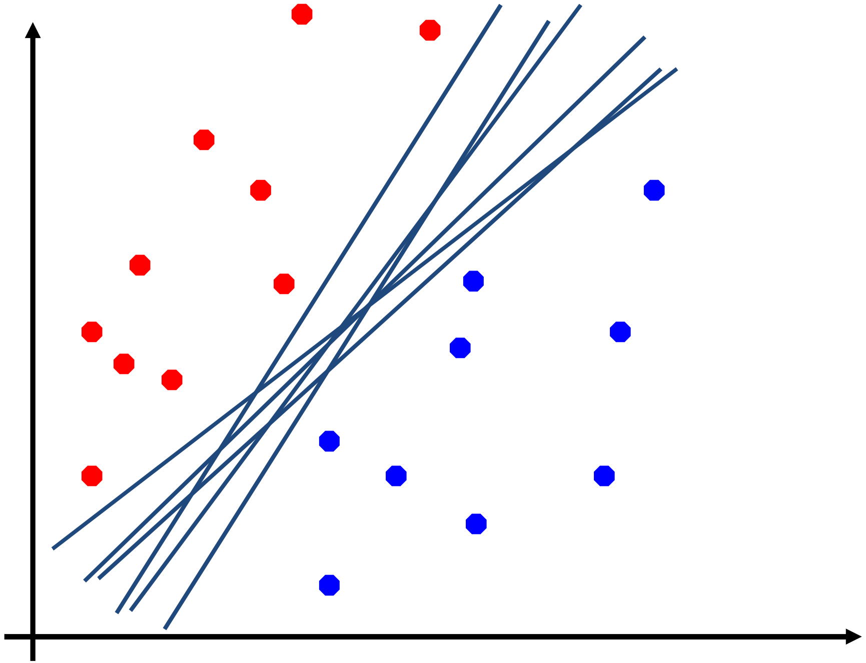
- Binary classification can be viewed as the task of separating classes in feature space:



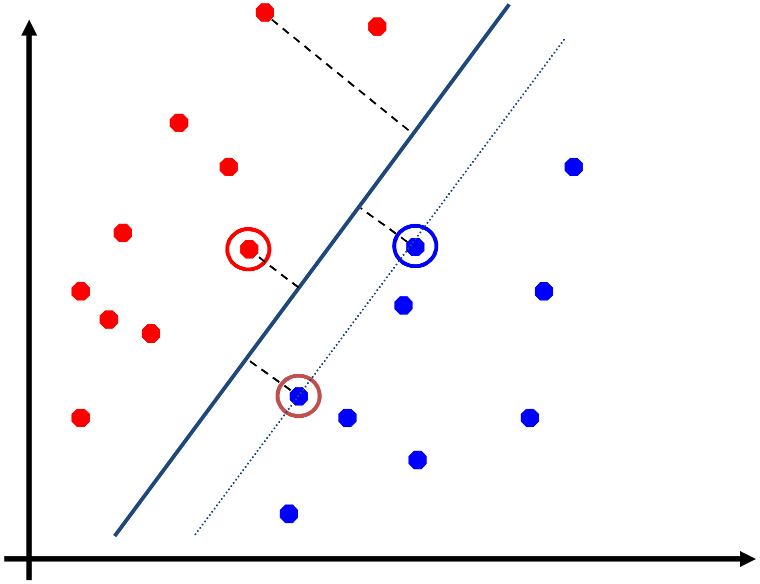
$$f(\mathbf{x}) = \text{sign}(\mathbf{w}^T\mathbf{x} + b)$$

Linear Separators

- Which of the linear separators is optimal?



Functional Margin

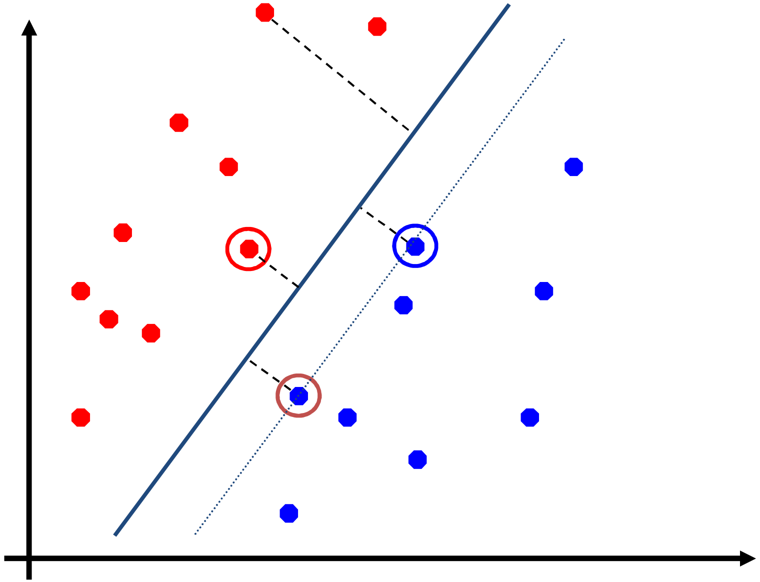


Functional margin: $\hat{r}_i = y_i(wx_i + b)$

The margin for all training data:

$$\hat{r} = \min \hat{r}_i \quad \text{(i=1,…N)}$$

Geometric Margin



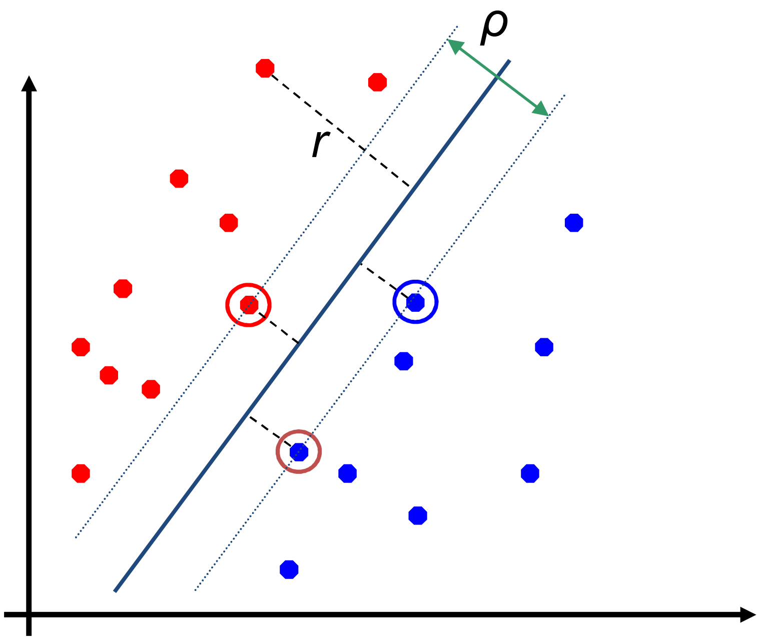
$$let\ \|w\| = 1$$

Then the functional margin become geometric margin:

$$r_i = y_i \left(\frac{wx_i + b}{\|w\|} \right)$$

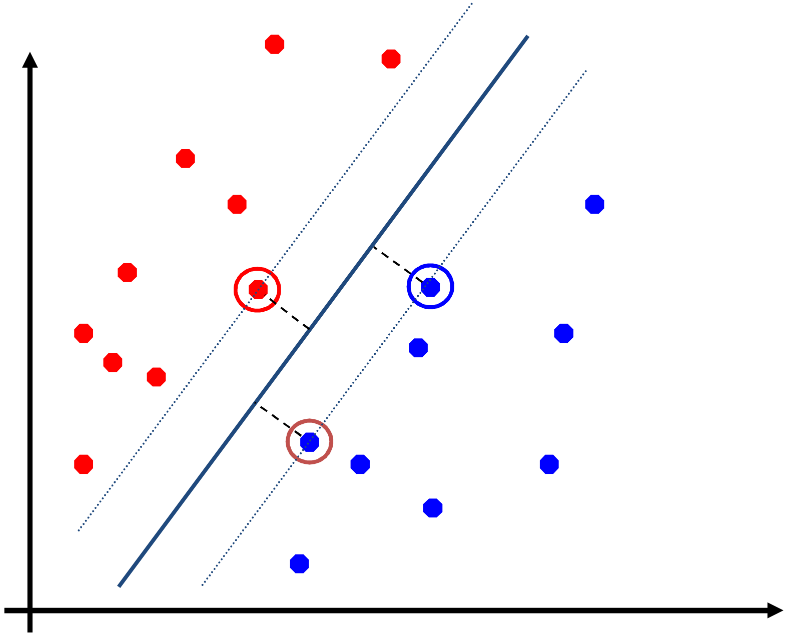
Classification Margin

- Distance from example $\mathbf{x}_i$ to the separator is $r = \frac{\mathbf{w}^T\mathbf{x}_i + b}{\|\mathbf{w}\|}$
- Examples closest to the hyperplane are ***support vectors***.
- ***Margin*** ρ of the separator is the distance between support vectors.



Maximum Margin Classification

- Maximizing the margin is good according to intuition and PAC theory.
- Implies that only support vectors matter; other training examples are ignorable.



Linear SVM

- Let training set $\{(\mathbf{x}_i, y_i)\}_{i=1..n}$, $\mathbf{x}_i \in \mathbf{R}^d$, $y_i \in \{-1, 1\}$ be separated by a hyperplane with margin ρ. Then for each training example $(\mathbf{x}_i, y_i)$:

$$\begin{aligned} \mathbf{w}^\mathrm{T}\mathbf{x}_i + b &\leq -\rho/2 \quad \text{if } y_i = -1 \\ \mathbf{w}^\mathrm{T}\mathbf{x}_i + b &\geq \rho/2 \quad \text{if } y_i = 1 \end{aligned} \quad \Leftrightarrow \quad y_i(\mathbf{w}^\mathrm{T}\mathbf{x}_i + b) \geq \rho/2$$

- For every support vector $\mathbf{x}_s$ the above inequality is an equality. After rescaling **w** and b by $\rho/2$ in the equality, we obtain that distance between each $\mathbf{x}_s$ and the hyperplane is

$$r = \frac{y_s(\mathbf{w}^T\mathbf{x}_s + b)}{\|\mathbf{w}\|} = \frac{1}{\|\mathbf{w}\|}$$

- Then the margin can be expressed through (rescaled) **w** and b as: $\rho = 2r = \dfrac{2}{\|\mathbf{w}\|}$

Linear SVM

- Then we can formulate the *quadratic optimization problem:*

$$\max_{w,b} \quad \gamma$$

$$\text{s.t.} \quad y_i\left(\frac{w}{\|w\|}\cdot x_i + \frac{b}{\|w\|}\right) \geqslant \gamma, \quad i=1,2,\cdots,N$$

- Considering the relation between functional margin and geometric margin:

$$\max_{w,b} \quad \frac{\hat{\gamma}}{\|w\|}$$

$$\text{s.t.} \quad y_i(w\cdot x_i + b) \geqslant \hat{\gamma}, \quad i=1,2,\cdots,N$$

Linear SVM

$$\max_{w,b} \quad \frac{\hat{\gamma}}{\| w \|}$$

$$\text{s.t.} \quad y_i(w \cdot x_i + b) \geqslant \hat{\gamma}, \quad i = 1, 2, \cdots, N$$

$$\hat{\gamma} = 1$$

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

$$\text{s.t.} \quad y_i(w \cdot x_i + b) - 1 \geqslant 0, \quad i = 1, 2, \cdots, N$$

The Optimization Problem Solution

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

$$\text{s.t.} \quad y_i(w \cdot x_i + b) - 1 \geqslant 0, \quad i = 1, 2, \cdots, N$$

Introducing Lagrange multiplier

$$L(w,b,\alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{N} \alpha_i y_i (w \cdot x_i + b) + \sum_{i=1}^{N} \alpha_i$$

The Optimization Problem Solution

$$L(w,b,\alpha)=\frac{1}{2}\|w\|^2-\sum_{i=1}^{N}\alpha_i y_i(w\cdot x_i+b)+\sum_{i=1}^{N}\alpha_i$$

$$\max_{\alpha}\min_{w,b} L(w,b,\alpha)$$

$$\nabla_w L(w,b,\alpha)=w-\sum_{i=1}^{N}\alpha_i y_i x_i=0$$

$$\nabla_b L(w,b,\alpha)=\sum_{i=1}^{N}\alpha_i y_i=0$$

$$w=\sum_{i=1}^{N}\alpha_i y_i x_i$$

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

The Optimization Problem Solution

$$L(w,b,\alpha)=\frac{1}{2}\|w\|^2-\sum_{i=1}^{N}\alpha_i y_i(w\cdot x_i+b)+\sum_{i=1}^{N}\alpha_i \qquad w=\sum_{i=1}^{N}\alpha_i y_i x_i$$

$$L(w,b,\alpha)=\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j(x_i\cdot x_j)-\sum_{i=1}^{N}\alpha_i y_i\left(\left(\sum_{j=1}^{N}\alpha_j y_j x_j\right)\cdot x_i+b\right)+\sum_{i=1}^{N}\alpha_i$$

$$=-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j(x_i\cdot x_j)+\sum_{i=1}^{N}\alpha_i$$

Therefore: $\min\limits_{w,b} L(w,b,\alpha)=-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j(x_i\cdot x_j)+\sum_{i=1}^{N}\alpha_i$

The Optimization Problem Solution

$$\max_{\alpha} \; -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j (x_i \cdot x_j) + \sum_{i=1}^{N}\alpha_i$$

$$\text{s.t.} \quad \sum_{i=1}^{N}\alpha_i y_i = 0 \qquad \alpha_i \geqslant 0, \quad i=1,2,\cdots,N$$

$$\Downarrow$$

$$\min_{\alpha} \quad \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j (x_i \cdot x_j) - \sum_{i=1}^{N}\alpha_i$$

$$\text{s.t.} \quad \sum_{i=1}^{N}\alpha_i y_i = 0$$

$$\alpha_i \geqslant 0, \quad i=1,2,\cdots,N$$

How to find *alpha* ?

$$\min_{\alpha} \quad \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j (x_i \cdot x_j) - \sum_{i=1}^{N}\alpha_i$$

$$\text{s.t.} \quad \sum_{i=1}^{N}\alpha_i y_i = 0$$

$$\alpha_i \geqslant 0, \quad i = 1,2,\cdots,N$$

K = X'*X ;

```
alpha = quadprog(Y*K*Y, - ones(n,1), ...
    [], [], ...
    y, 0, ...
    zeros(n,1), C * ones(n,1), ...
    [], optimset('display','off','largescale', 'off', 'algorithm', 'active-set')) ;
```

Determining the model parameter

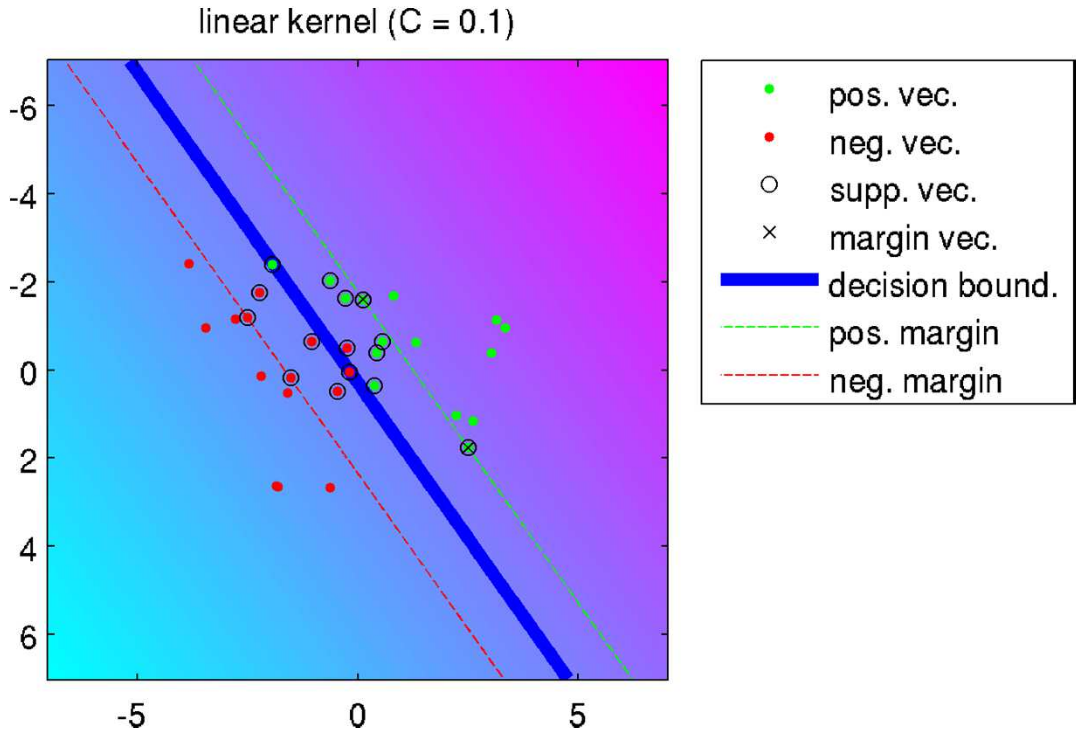
$$\alpha^* = (\alpha_1^*, \alpha_2^*, \cdots, \alpha_l^*)^{\mathrm{T}}$$

$$w^* = \sum_{i=1}^{N} \alpha_i^* y_i x_i$$

$$b^* = y_j - \sum_{i=1}^{N} \alpha_i^* y_i (x_i \cdot x_j)$$

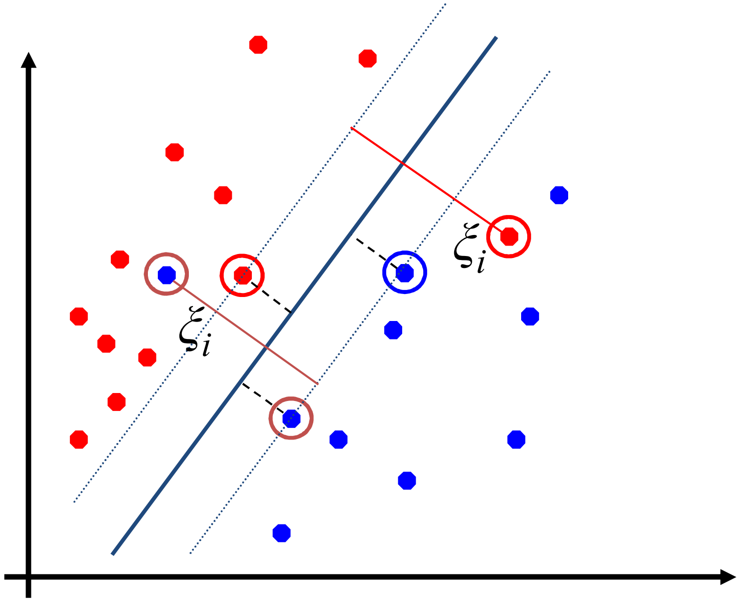
$$f(x) = \operatorname{sign}\left(\sum_{i=1}^{N} \alpha_i^* y_i (x \cdot x_i) + b^* \right)$$

Example


linear kernel (C = 0.1)
-6
-4
-2
0
2
4
6
-5
0
5
pos. vec.
neg. vec.
supp. vec.
margin vec.
decision bound.
pos. margin
neg. margin

Soft Margin Classification

- What if the training set is not linearly separable?
- *Slack variables* ξ_i can be added to allow misclassification of difficult or noisy examples, resulting margin called *soft*.



Soft Margin Classification

- The old formulation:

Find **w** and b such that

$\mathbf{\Phi}(\mathbf{w}) = \mathbf{w}^T\mathbf{w}$ is minimized

and for all $(\mathbf{x}_i, y_i)$, $i=1..n$: $y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1$

- Modified formulation incorporates slack variables:

Find **w** and b such that

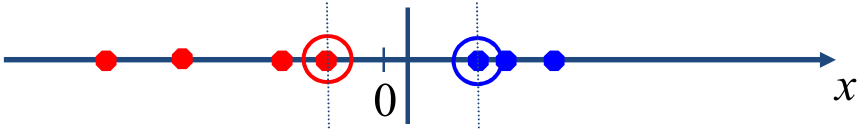
$\mathbf{\Phi}(\mathbf{w}) = \mathbf{w}^T\mathbf{w} + C\Sigma\xi_i$ is minimized

and for all $(\mathbf{x}_i, y_i)$, $i=1..n$: $y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i$, $\xi_i \geq 0$

- Parameter *C* can be viewed as a way to control overfitting: it “trades off” the relative importance of maximizing the margin and fitting the training data.

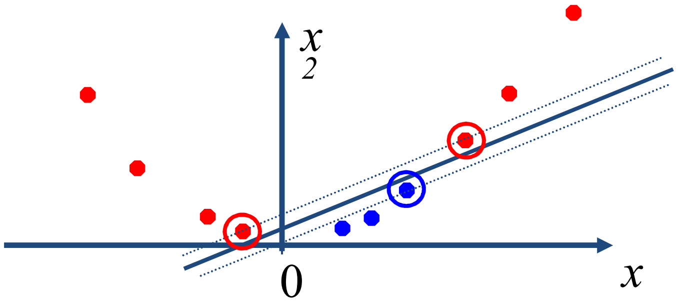
Non-linear SVMs

- Datasets that are linearly separable with some noise work out great:



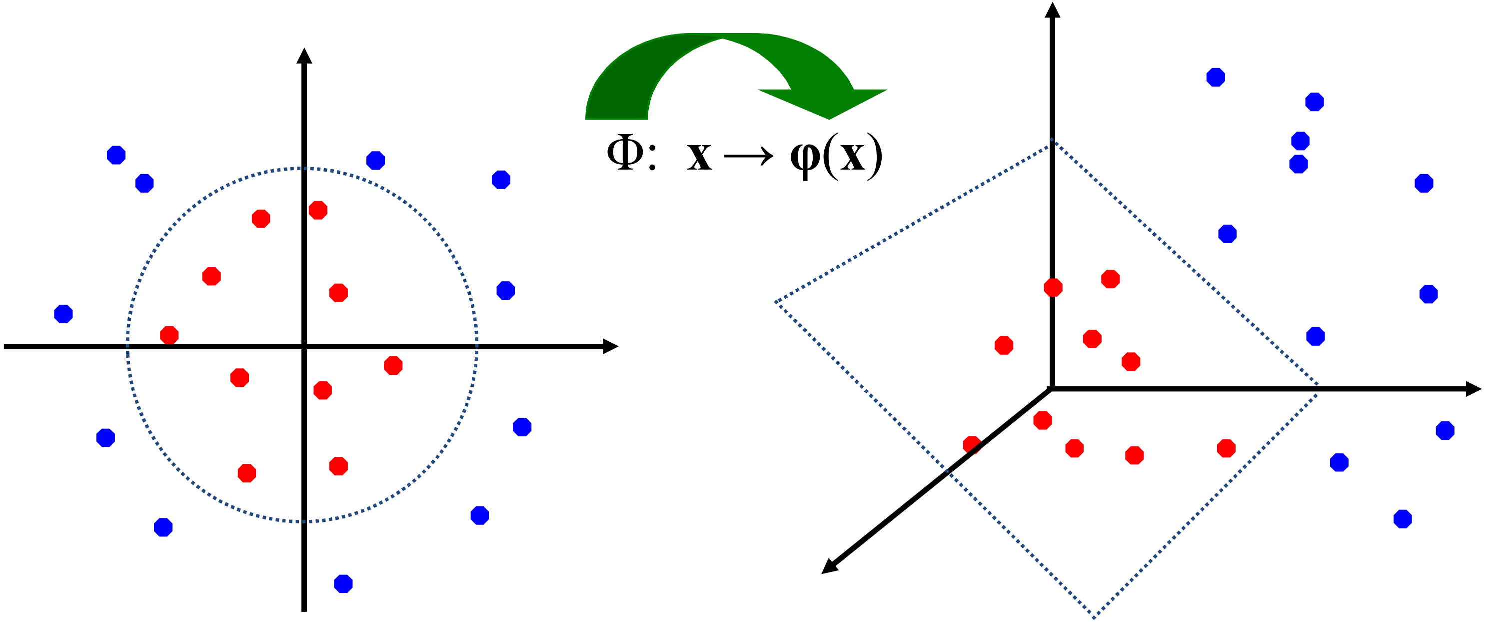
- But what are we going to do if the dataset is just too hard?

- How about… mapping data to a higher-dimensional space:



Non-linear SVMs: Feature spaces

- General idea: the original feature space can always be mapped to some higher-dimensional feature space where the training set is separable:



Examples of Kernel Functions

- Linear: $K(\mathbf{x}_i,\mathbf{x}_j)= \mathbf{x}_i^T\mathbf{x}_j$
  - Mapping Φ: $\mathbf{x} \rightarrow \boldsymbol{\phi}(\mathbf{x})$, where $\boldsymbol{\phi}(\mathbf{x})$ is **x** itself
- Polynomial of power *p*: $K(\mathbf{x}_i,\mathbf{x}_j)= (1+ \mathbf{x}_i^T\mathbf{x}_j)^p$
  - Mapping Φ: $\mathbf{x} \rightarrow \boldsymbol{\phi}(\mathbf{x})$, where $\boldsymbol{\phi}(\mathbf{x})$ has $\binom{d+p}{p}$ dimensions
- Gaussian (radial-basis function): $K(\mathbf{x}_i,\mathbf{x}_j) = e^{-\frac{\|\mathbf{x}_i-\mathbf{x}_j\|^2}{2\sigma^2}}$
  - Mapping Φ: $\mathbf{x} \rightarrow \boldsymbol{\phi}(\mathbf{x})$, where $\boldsymbol{\phi}(\mathbf{x})$ is *infinite-dimensional*: every point is mapped to *a function* (a Gaussian); combination of functions for support vectors is the separator.
- Higher-dimensional space still has *intrinsic* dimensionality *d* (the mapping is not *onto*), but linear separators in it correspond to *non-linear* separators in original space.

Non-linear SVMs Mathematically

- Dual problem formulation:

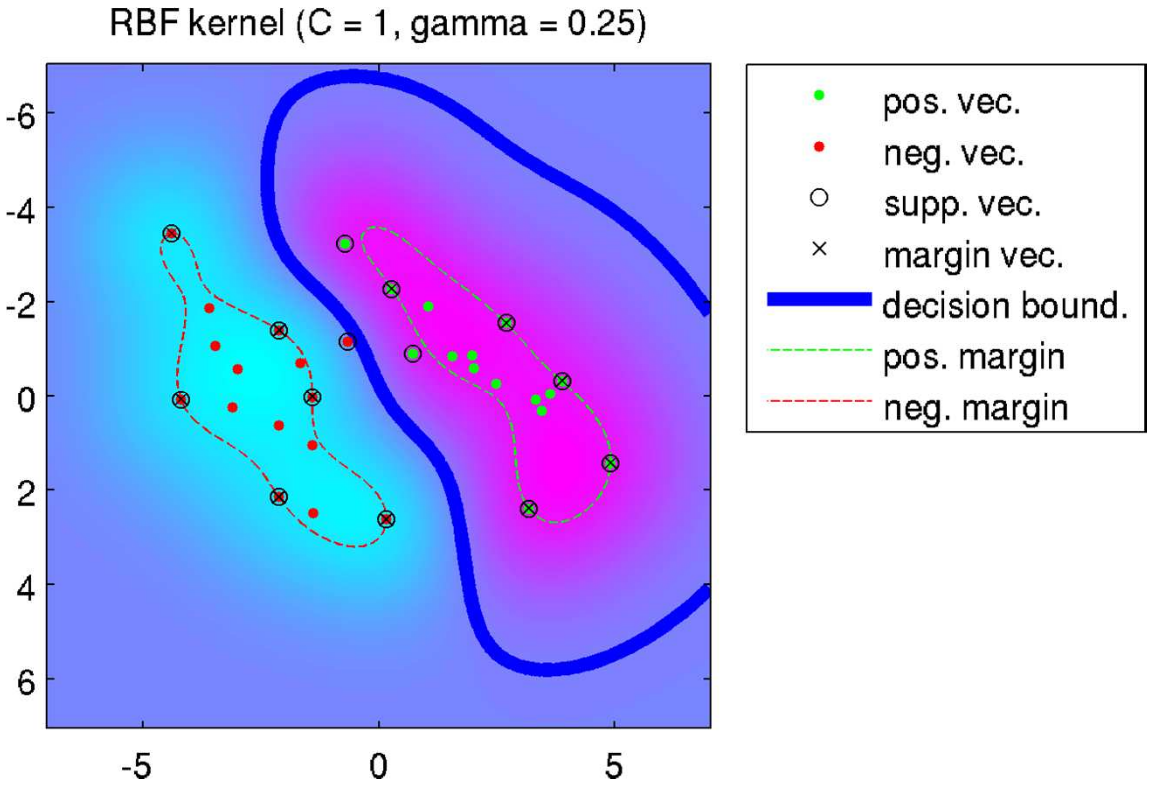
Find $\alpha_1 \dots \alpha_n$ such that
$\mathbf{Q}(\boldsymbol{\alpha}) = \Sigma\alpha_i - \frac{1}{2}\Sigma\Sigma\alpha_i\alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$ is maximized and
(1) $\Sigma\alpha_i y_i = 0$
(2) $\alpha_i \geq 0$ for all α_i

- The solution is:

$$f(\mathbf{x}) = \Sigma\alpha_i y_i K(\mathbf{x}_i, \mathbf{x}_j) + b$$

- Optimization techniques for finding α_i's remain the same!

Example


RBF kernel (C = 1, gamma = 0.25)
-6
-4
-2
0
2
4
6
-5
0
5
pos. vec.
neg. vec.
supp. vec.
margin vec.
decision bound.
pos. margin
neg. margin