Artificial Intelligence & Machine Learning and Pattern Recognition ——Support Vector Machine (Opt.)

Yanghui Rao
Assistant Prof., Ph.D
School of Mobile Information Engineering,
Sun Yat-sen University
raoyangh@mail.sysu.edu.cn

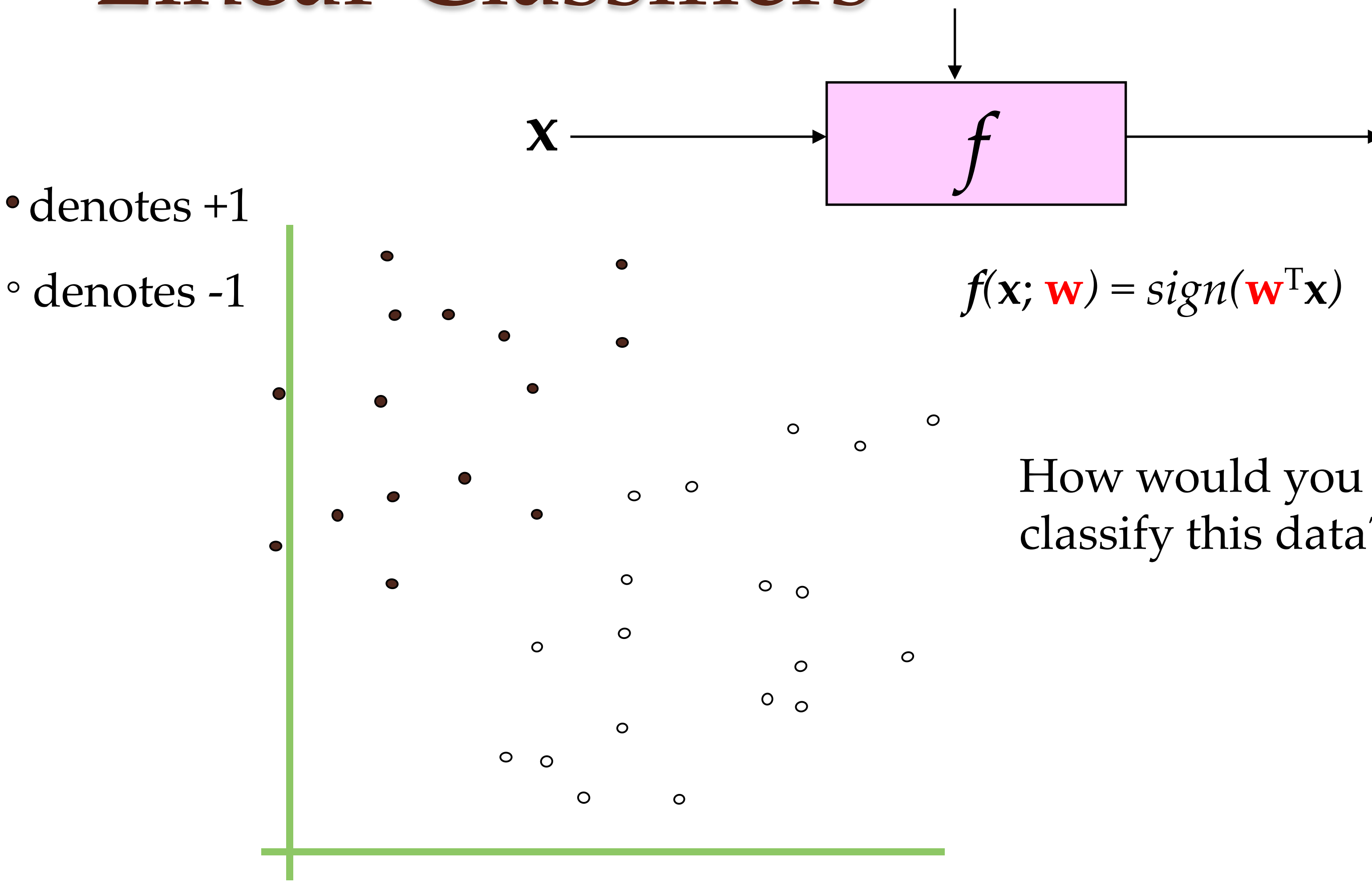
Support Vector Machine

- SVM (支持向量机) is a classifier derived from statistical learning theory by Vapnik and Chervonenkis
- Initially popularized in the Neural Information Processing Systems (NIPS) community, now an important and active field of all Machine Learning research.
- Vapnik Chervonenkis theory

Support Vector Machine

- SVMs are learning systems that
  - use a hypothesis space of *linear functions*
  - in a high dimensional feature space — *Kernel function*
  - trained with a learning algorithm from optimization theory — *Lagrange*
  - Implements a learning bias derived from statistical learning theory — *Generalisation* SVM is a classifier derived from statistical learning theory by Vapnik and Chervonenkis

Linear Classifiers

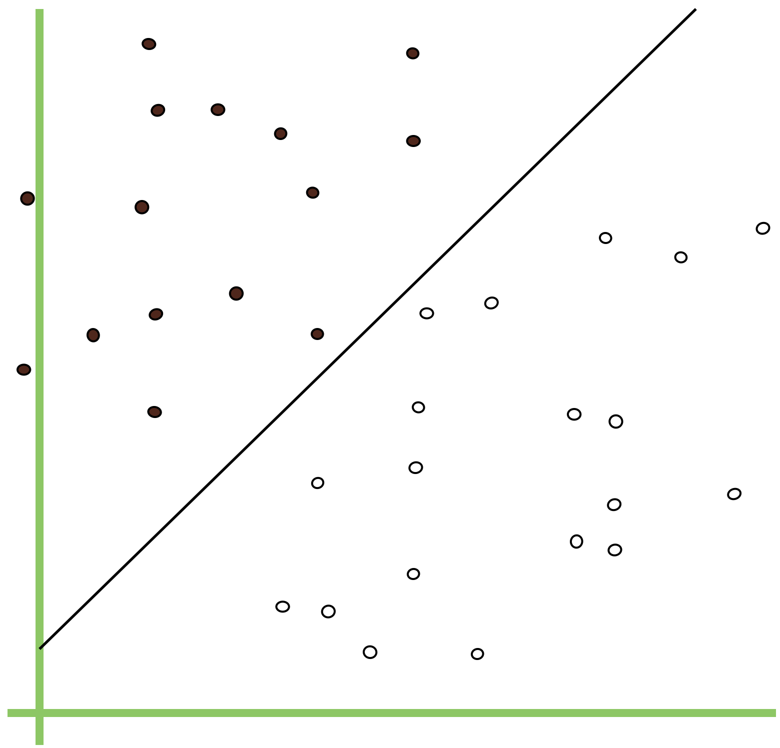


$$f(\mathbf{x}; \mathbf{w}) = sign(\mathbf{w}^T \mathbf{x})$$

How would you classify this data?

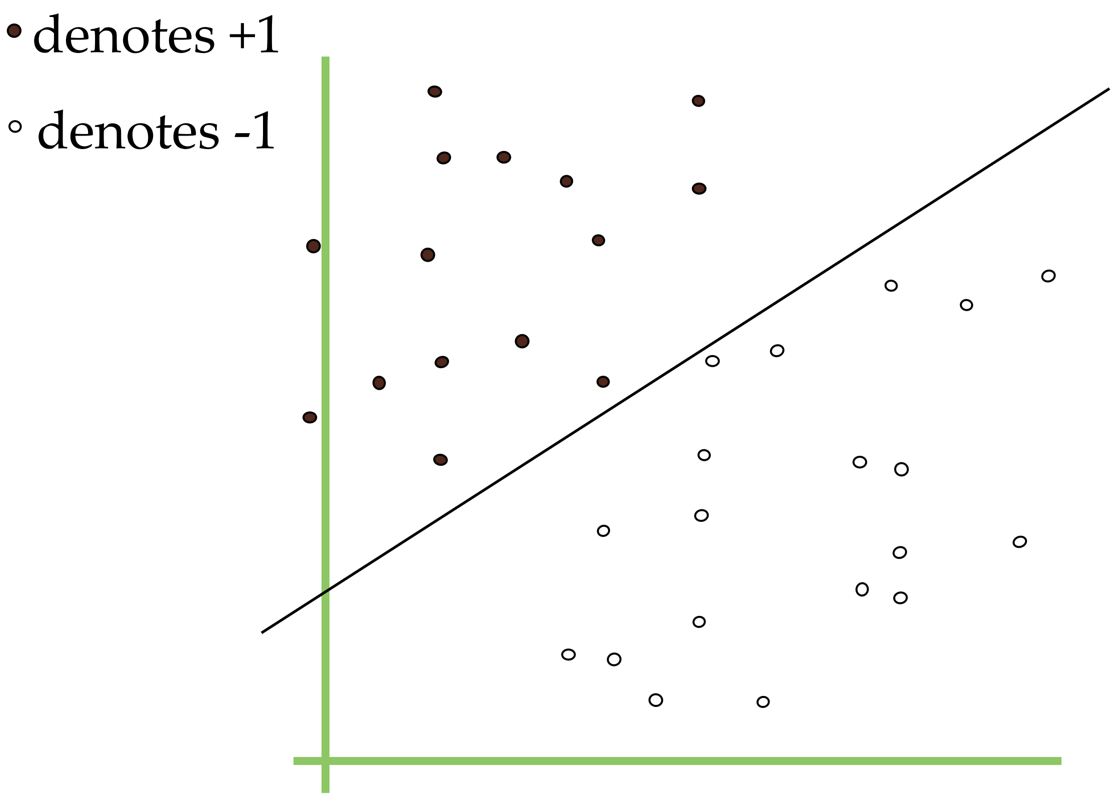
Linear Classifiers

- denotes +1
- denotes -1



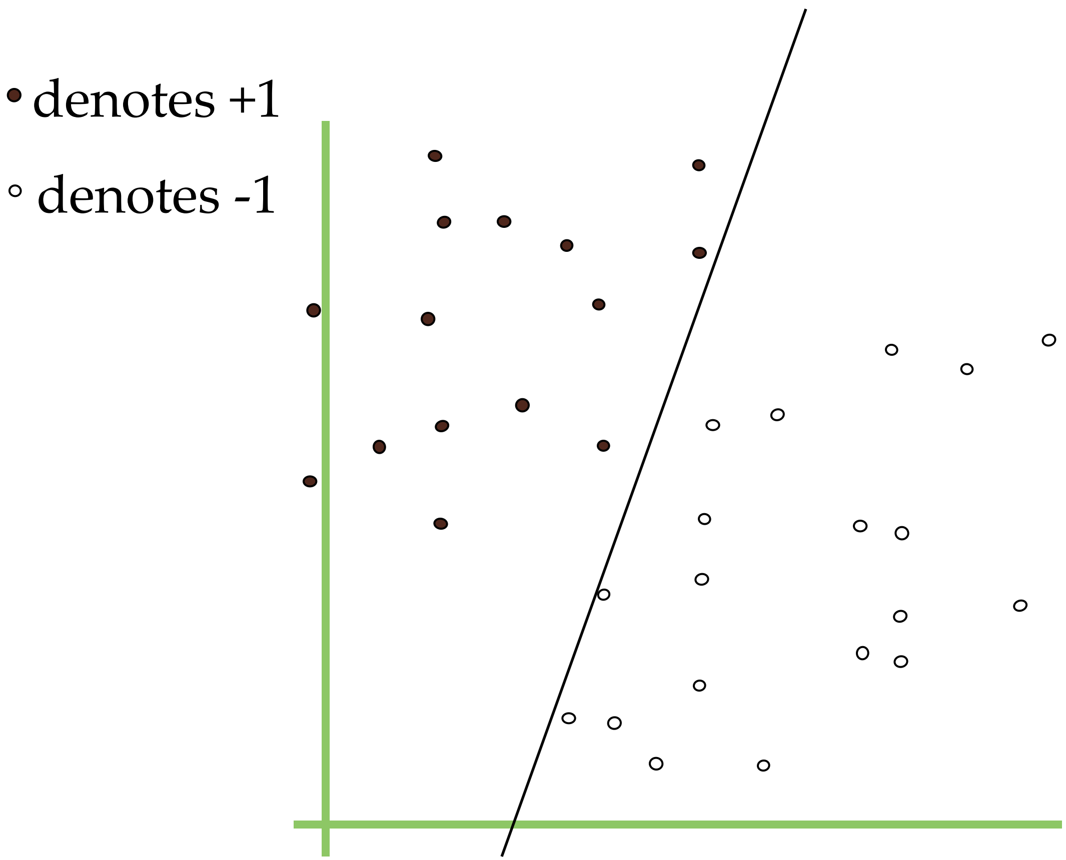
How would you classify this data?

Linear Classifiers



How would you classify this data?

Linear Classifiers

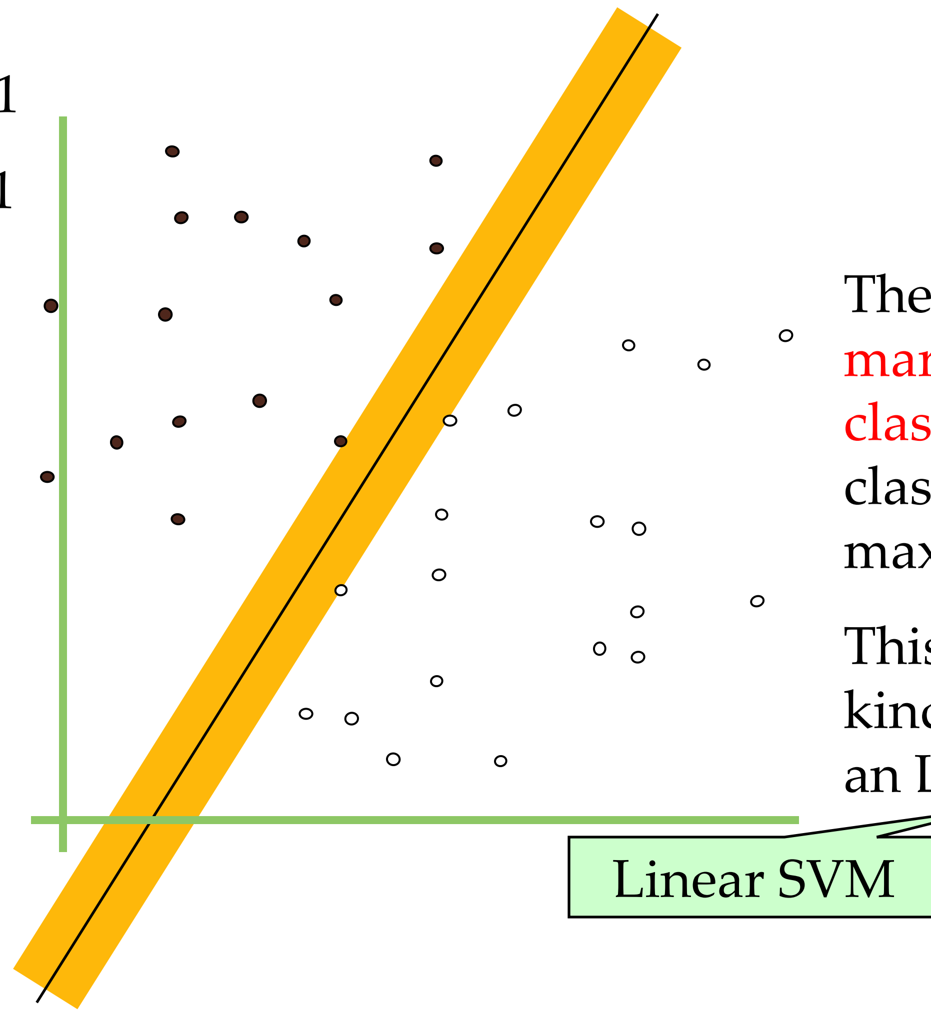


How would you classify this data?

Maximum Margin

• denotes +1

◦ denotes -1



The maximum margin linear classifier is the linear classifier with the maximum margin.

This is the simplest kind of SVM (Called an LSVM)

Maximum Margin

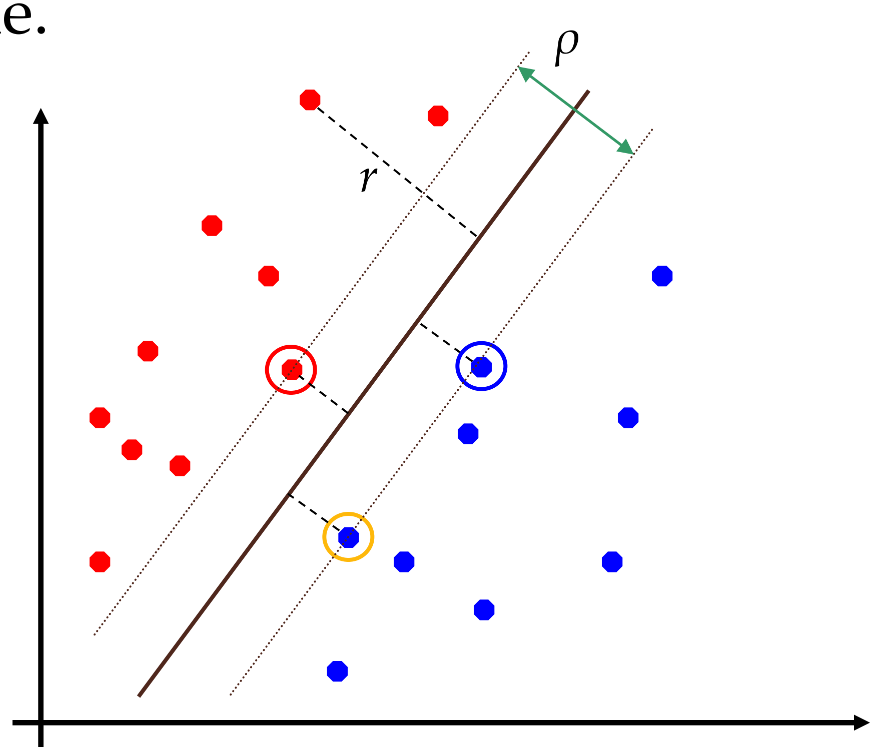
- The geometric margin of the separator

$$\rho = \frac{1}{2}\left(\frac{\tilde{\mathbf{w}}^{\mathrm{T}}\tilde{\mathbf{x}}^{+}}{\|\mathbf{w}\|} - \frac{\tilde{\mathbf{w}}^{\mathrm{T}}\tilde{\mathbf{x}}^{-}}{\|\mathbf{w}\|}\right)$$

- In order to find the maximum ρ , we must find the minimum $\|\mathbf{w}\|$
  - subject to (s.t.) $y_i(\tilde{\mathbf{w}}^{\mathrm{T}}\tilde{\mathbf{x}}_i) - 1 \geq 0, \quad i = 1, 2, \ldots, n$
  - Examples closest to the hyperplane are *support vectors*.

Maximum Margin

- Margin ρ of the separator is the distance between support vectors
- Maximizing the margin implies that only support vectors matter; other training examples are ignorable.



Maximum Margin

- Need to optimize a quadratic function subject to linear constraints.
- Quadratic optimization problems are a well-known class of mathematical programming problems for which several (non-trivial) algorithms exist.
- The solution involves constructing a dual problem where a Lagrange multiplier α_i is associated with every inequality constraint in the primal (original) problem:

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

Maximum Margin

- Given a solution $\alpha_1 \ldots \alpha_n$ to the dual problem, solution to the primal is:

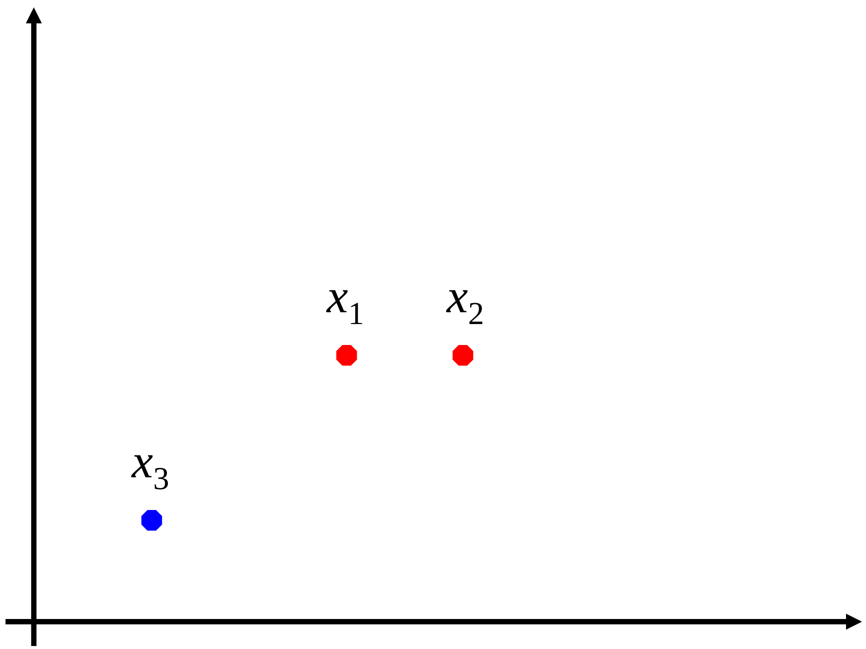
$$\boxed{\mathbf{w} = \Sigma\alpha_i y_i \mathbf{x}_i \qquad w_0 = y_k - \Sigma\alpha_i y_i \mathbf{x}_i^{\mathbf{T}} \mathbf{x}_k \quad \text{for any } \alpha_k > 0}$$

- Each non-zero α_i indicates that corresponding x_i is a support vector.
- Then the classifying function is (note that we don't need **w** explicitly):

$$\boxed{f(\mathbf{x}) = \Sigma\alpha_i y_i \mathbf{x}_i^{\mathbf{T}} \mathbf{x} + w_0}$$

Example

- Training data with "+"
  - $x_1 = (3,3)$, $x_2 = (4,3)$
- Training data with "-"
  - $x_3 = (1,1)$



Example

- Minimize the following objective function:

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

$$= \frac{1}{2}(18\alpha_1^2 + 25\alpha_2^2 + 2\alpha_3^2 + 42\alpha_1\alpha_2 - 12\alpha_1\alpha_3 - 14\alpha_2\alpha_3) - \alpha_1 - \alpha_2 - \alpha_3$$

$$s.t. \qquad \alpha_1 + \alpha_2 - \alpha_3 = 0$$

$$\alpha_i \geq 0, \qquad i = 1, 2, 3$$

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

Example

- Minimize the following objective function:

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

$$= \frac{1}{2}(18\alpha_1^2 + 25\alpha_2^2 + 2\alpha_3^2 + 42\alpha_1\alpha_2 - 12\alpha_1\alpha_3 - 14\alpha_2\alpha_3) - \alpha_1 - \alpha_2 - \alpha_3$$

$$s.t. \quad \boxed{\alpha_1 + \alpha_2 - \alpha_3 = 0}$$

$$\alpha_i \geq 0, \qquad i = 1,2,3$$

$$\min_{\alpha_1,\alpha_2}(4\alpha_1^2 + \frac{13}{2}\alpha_2^2 + 10\alpha_1\alpha_2 - 2\alpha_1 - 2\alpha_2)$$

$$s.t. \qquad \alpha_i \geq 0, \qquad i = 1,2$$

$$\frac{\partial O}{\partial \alpha_1} = 8\alpha_1 + 10\alpha_2 - 2 = 0 \qquad \alpha_1 = \frac{3}{2}$$

$$\frac{\partial O}{\partial \alpha_2} = 13\alpha_2 + 10\alpha_1 - 2 = 0 \qquad \alpha_2 = -1$$

Example

- Minimize the following objective function:

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

$$= \frac{1}{2}(18\alpha_1^2 + 25\alpha_2^2 + 2\alpha_3^2 + 42\alpha_1\alpha_2 - 12\alpha_1\alpha_3 - 14\alpha_2\alpha_3) - \alpha_1 - \alpha_2 - \alpha_3$$

$$s.t. \quad \boxed{\alpha_1 + \alpha_2 - \alpha_3 = 0}$$

$$\alpha_i \geq 0, \quad i = 1,2,3$$

$$\min_{\alpha_1,\alpha_2}(4\alpha_1^2 + \frac{13}{2}\alpha_2^2 + 10\alpha_1\alpha_2 - 2\alpha_1 - 2\alpha_2)$$

$$s.t. \quad \alpha_i \geq 0, \quad i = 1,2$$

$$\frac{\partial O}{\partial \alpha_1} = 8\alpha_1 + 10\alpha_2 - 2 = 0$$

$$\frac{\partial O}{\partial \alpha_2} = 13\alpha_2 + 10\alpha_1 - 2 = 0$$

$$\alpha_1 = \frac{3}{2}$$

$$\boxed{\alpha_2 = -1}$$

$$\alpha_1 = 0, \alpha_2 = \frac{2}{13}, O = -\frac{2}{13}$$

$$\boxed{\alpha_2 = 0, \alpha_1 = \frac{1}{4}, O = -\frac{1}{4}}$$

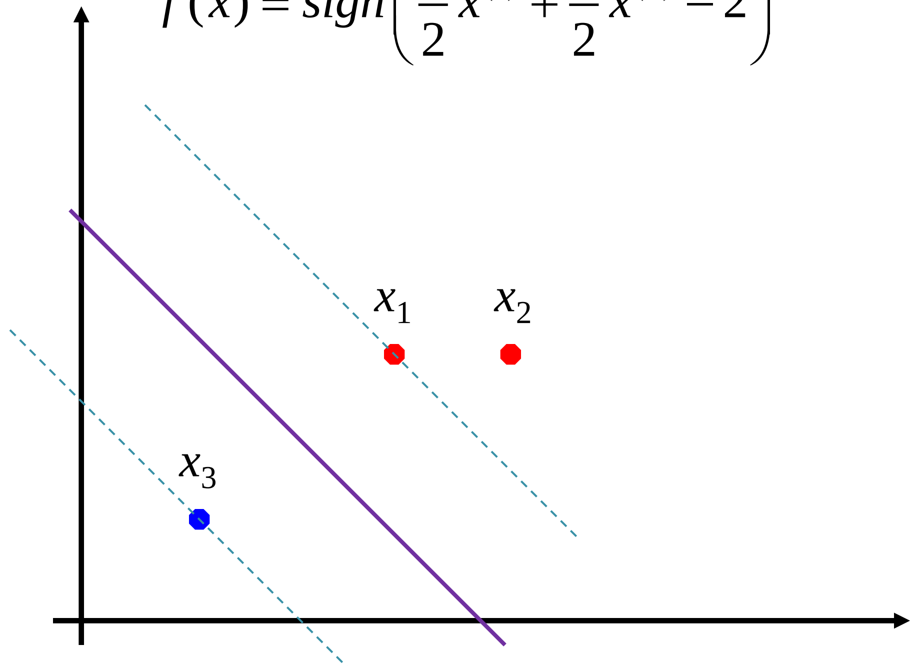
Example

- x_1 and x_3 are support vectors $\quad \alpha_1^* = \frac{1}{4}, \alpha_2^* = 0, \alpha_3^* = \frac{1}{4}$

$$\boxed{\mathbf{w} = \Sigma \alpha_i y_i \mathbf{x}_i \qquad w_0 = y_k - \Sigma \alpha_i y_i \mathbf{x}_i^{\mathbf{T}} \mathbf{x}_k \quad \text{for any } \alpha_k > 0}$$

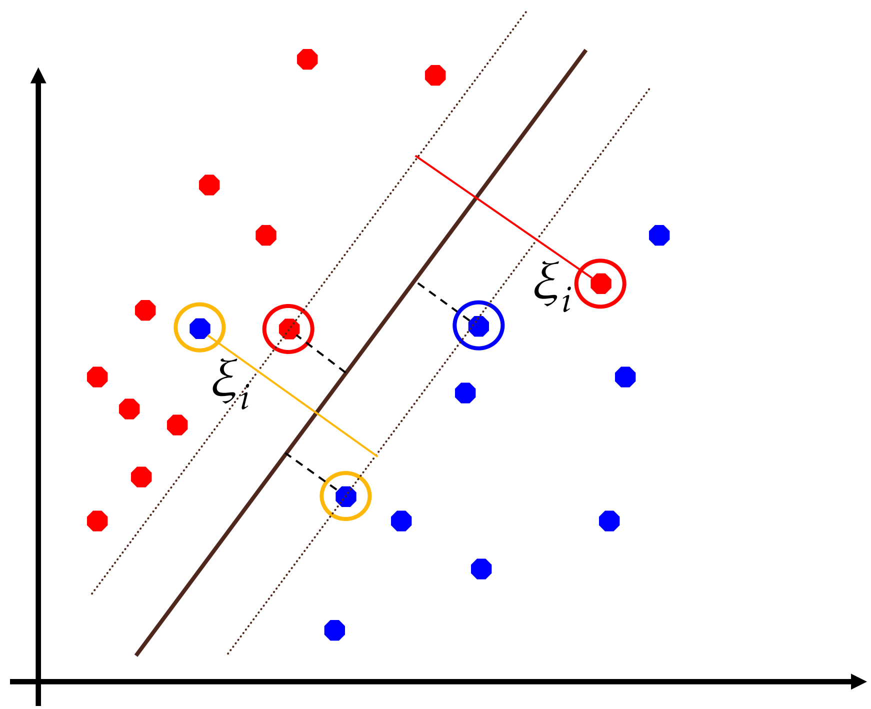
$$w_1^* = w_2^* = \frac{1}{2}, w_0^* = -2$$

$$f(x) = sign\left(\frac{1}{2}x^{(1)} + \frac{1}{2}x^{(2)} - 2\right)$$



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 $\mathbf{w}$ and b such that
$\mathbf{\Phi}(\mathbf{w}) = \mathbf{w}^{\mathrm{T}}\mathbf{w}$ is minimized
and for all $(\mathbf{x}_i, y_i)$, $i=1..n$: $y_i\,(\mathbf{w}^{\mathbf{T}}\mathbf{x}_i + w_0) \geq 1$

- Modified formulation incorporates slack variables:

Find $\mathbf{w}$ and b such that
$\mathbf{\Phi}(\mathbf{w}) = \mathbf{w}^{\mathrm{T}}\mathbf{w} + C\Sigma\xi_i$ is minimized
and for all $(\mathbf{x}_i, y_i)$, $i=1..n$: $y_i\,(\mathbf{w}^{\mathbf{T}}\mathbf{x}_i + w_0) \geq 1 - \xi_{i,}$, $\xi_i \geq 0$

- 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.

Soft Margin Classification

- Solution to the dual problem is:

$$\boxed{\begin{array}{l}\mathbf{w} \;= \Sigma \alpha_i y_i \mathbf{x}_i \\ w_0 = y_k(1 - \xi_k) - \Sigma \alpha_i y_i \mathbf{x}_i^{\mathbf{T}} \mathbf{x}_k \quad \text{for any } k \text{ s.t. } \alpha_k > 0\end{array}}$$

- Again, we don't need to compute **w** explicitly for classification:

$$\boxed{f(\mathbf{x}) = \Sigma \alpha_i y_i \mathbf{x}_i^{\mathbf{T}} \mathbf{x} + w_0}$$

Linear SVMs: Overview

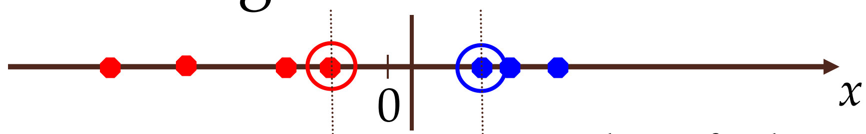
- The classifier is a *separating hyperplane.*
- Most "important" training points are support vectors; they define the hyperplane.
- Quadratic optimization algorithms can identify which training points $\mathbf{x}_i$ are support vectors with non-zero Lagrangian multipliers α_i.
- Both in the dual formulation of the problem and in the solution training points appear only inside inner products:

Find $\alpha_1 \ldots \alpha_N$ such that
$\mathbf{Q}(\boldsymbol{\alpha}) = \Sigma\alpha_i - \frac{1}{2}\Sigma\Sigma\alpha_i\alpha_j y_i y_j \boxed{\mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j}$ is maximized and
(1) $\Sigma\alpha_i y_i = 0$
(2) $0 \leq \alpha_i \leq C$ for all α_i

$$f(\mathbf{x}) = \Sigma\alpha_i y_i \boxed{\mathbf{x}_i^{\mathrm{T}}\mathbf{x}} + w_0$$

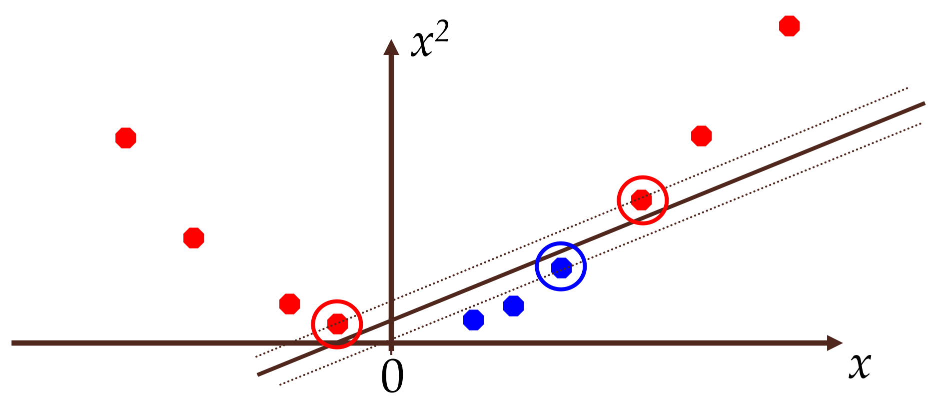
Non-linear SVMs

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



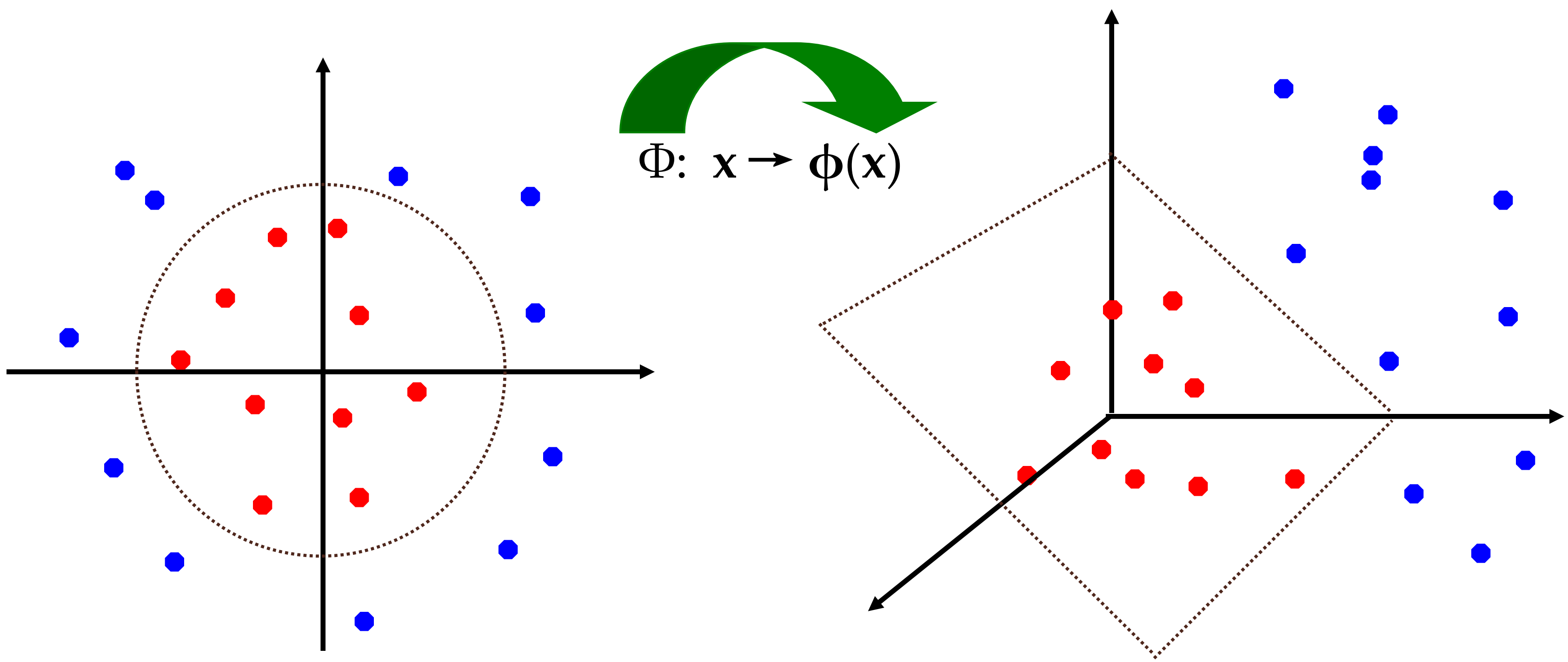
- But what are we going to do if the dataset is just too hard?

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



Non-linear SVMs

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



Non-linear SVMs: Kernel

- The linear classifier relies on inner product between vectors $K(\mathbf{x}_i,\mathbf{x}_j)=\mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j$
- If every data point is mapped into high-dimensional space via some transformation Φ: $\mathbf{x} \rightarrow \phi(\mathbf{x})$, the inner product becomes:

  $K(\mathbf{x}_i,\mathbf{x}_j)= \boldsymbol{\phi}(\mathbf{x}_i)^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_j)$
- A *kernel function* is a function that is equivalent to an inner product in some feature space. Thus, a kernel function implicitly maps data to a high-dimensional space (without the need to compute each $\phi(x)$ explicitly).

Example

$$\phi \;:\; \mathbf{x} = (x_1, x_2) \longmapsto \phi(\mathbf{x}) = (x_1^2, x_2^2, \sqrt{2}x_1x_2) \in F = \mathbb{R}^3.$$

$$\begin{aligned} \langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle &= \left\langle (x_1^2, x_2^2, \sqrt{2}x_1x_2), (z_1^2, z_2^2, \sqrt{2}z_1z_2) \right\rangle \\ &= x_1^2z_1^2 + x_2^2z_2^2 + 2x_1x_2z_1z_2 \\ &= (x_1z_1 + x_2z_2)^2 = \langle \mathbf{x}, \mathbf{z} \rangle^2 . \end{aligned}$$

$$\kappa(\mathbf{x}, \mathbf{z}) = \langle \mathbf{x}, \mathbf{z} \rangle^2$$

The same kernel computes the inner product corresponding to the four-dimensional feature map

$$\phi \;:\; \mathbf{x} = (x_1, x_2) \longmapsto \phi(\mathbf{x}) = (x_1^2, x_2^2, x_1x_2, x_2x_1) \in F = \mathbb{R}^4.$$

a kernel $\kappa(x, z)$ satisfying $\kappa(\mathbf{x}, \mathbf{z}) = \langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle$

Kernel Functions

- Linear: $K(\mathbf{x}_i,\mathbf{x}_j)= \mathbf{x}_i^{\mathbf{T}}\mathbf{x}_j$
  - Mapping Φ: $\mathbf{x} \rightarrow \boldsymbol{\phi}(\mathbf{x})$, where $\boldsymbol{\phi}(\mathbf{x})$ is $\mathbf{x}$ itself
- Polynomial of power p: $K(\mathbf{x}_i,\mathbf{x}_j)= (1+ \mathbf{x}_i^{\mathbf{T}}\mathbf{x}_j)^p$
  - 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

- Dual problem formulation:

Find $\alpha_1 \ldots \alpha_n$ 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:

$f(\mathbf{x}) = \Sigma\alpha_i y_i K(\mathbf{x}_i, \mathbf{x}_j) + w_0$

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