Support Vector Machines

MFML Team

BITS Pilani
Pilani Campus

innovate achieve lead

BITS Pilani
Pilani Campus

Text Book(s)

T1	Christopher Bishop: Pattern Recognition and Machine Learning, Springer International Edition
T2	Tom M. Mitchell: Machine Learning, The McGraw-Hill Companies, Inc..

These slides are prepared by the instructor, with grateful acknowledgement of Prof. Tom Mitchell, Prof.. Burges, Prof. Andrew Moore and many others who made their course materials freely available online.

Topics to be covered

- Nonlinear SVM
- Kernel Trick
- SVM Kernels
- XOR problem using non linear SVM

Optimization Problem

Find $\mathbf{w}$ and b such that

$\Phi(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T\mathbf{w} = \frac{1}{2}\|\mathbf{w}\|^2$ is minimized;

and for all $\{(\mathbf{x_i}, y_i)\}$: $y_i(\mathbf{w}^T\mathbf{x_i} + b) \geq 1$

$L(w, b, \alpha_i) = \frac{1}{2}\|\mathbf{w}\|^2 - \Sigma\, \alpha_i [y_i(\mathbf{w}^T\mathbf{x_i} + b) - 1]$

Find $\alpha_1 \ldots \alpha_N$ such that

$\mathbf{Q}(\boldsymbol{\alpha}) = \Sigma\alpha_i - \frac{1}{2}(\sum_i \sum_j \alpha_i \alpha_j y_i y_j \mathbf{x_i} \cdot \mathbf{x_j})$ is maximized and

(1) $\Sigma\alpha_i y_i = 0$

(2) $\alpha_i \geq 0$ for all α_i

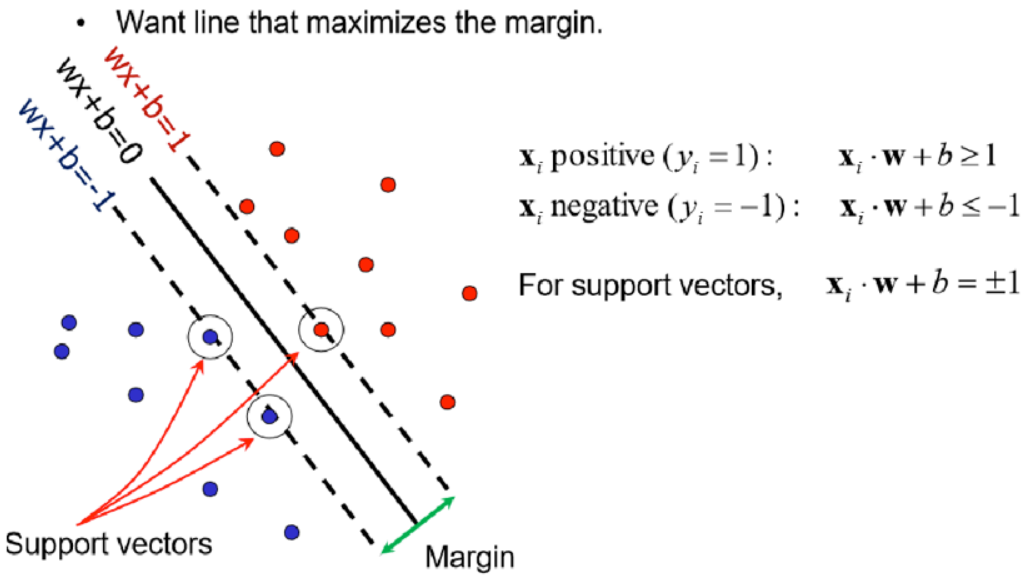
Support Vectors

Using KKT conditions :
$\alpha_i [y_i (\mathbf{w}^T\mathbf{x_i} + b) - 1] = 0$

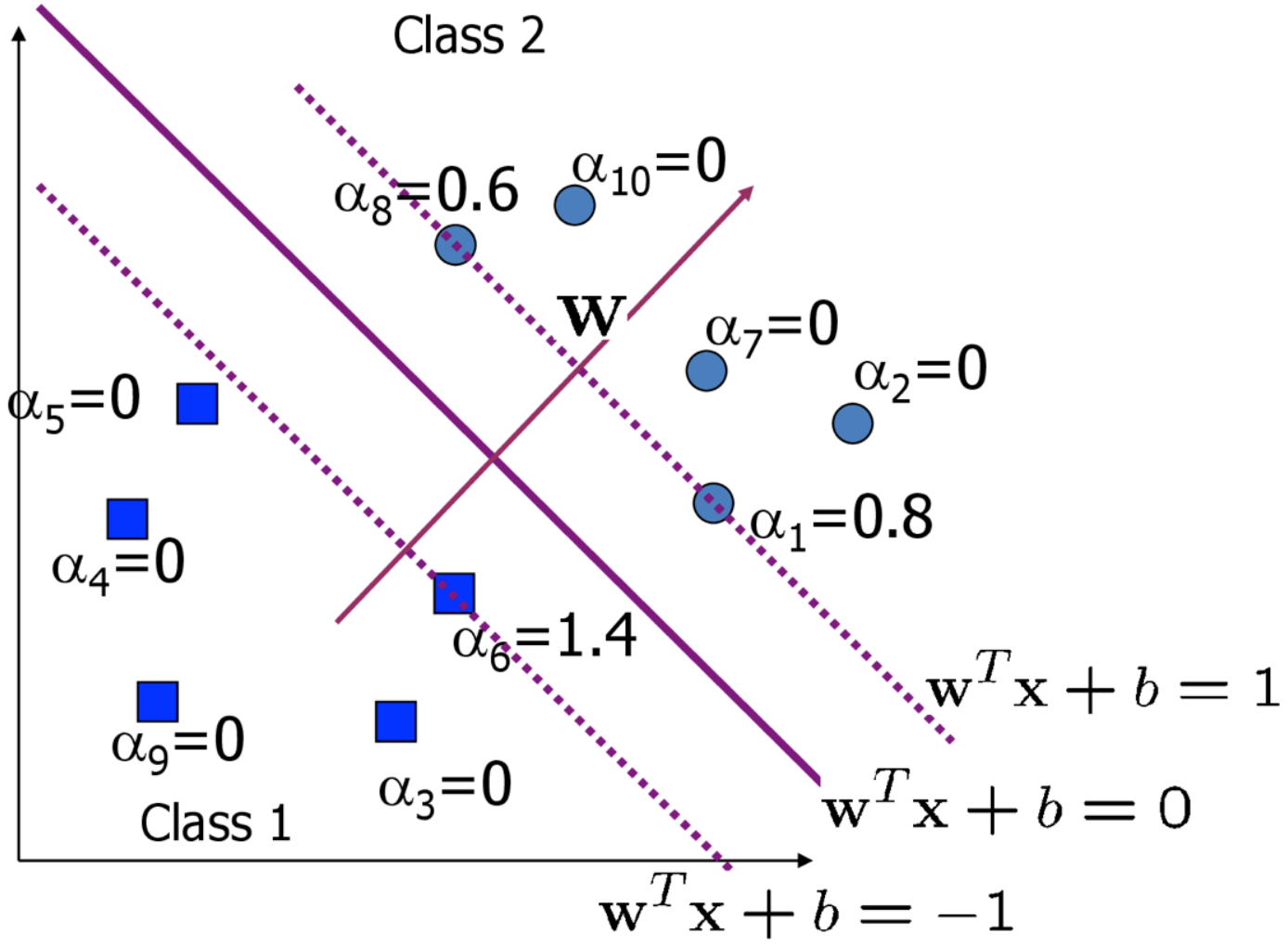
For this condition to be satisfied either $\alpha_i = 0$ and $y_i (\mathbf{w}^T\mathbf{x_i} + b) - 1 > 0$
OR
$y_i (\mathbf{w}^T\mathbf{x_i} + b) - 1 = 0$ and $\alpha_i > 0$

For support vectors:
$y_i (\mathbf{w}^T\mathbf{x_i} + b) - 1 = 0$

For all points other than support vectors:
$\alpha_i = 0$

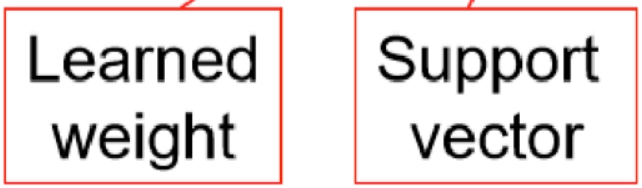


A Geometrical Interpretation



Solving the Optimization Problem

- Solution: $\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$



Solving the Optimization Problem

- Solution: $\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$

  $b = y_i - \mathbf{w} \cdot \mathbf{x}_i$ (for any support vector)

- Classification function:

$$f(x) = \text{sign}\,(\mathbf{w} \cdot \mathbf{x} + \text{b})$$

$$= \text{sign}\left(\sum_{\text{i}} \alpha_{\text{i}} y_i \mathbf{x}_i \cdot \mathbf{x} + b\right)$$

If f(x) < 0, classify as negative, otherwise classify as positive.

- Notice that it relies on an *inner product* between the test point ***x*** and the support vectors $\boldsymbol{x}_i$
- (Solving the optimization problem also involves computing the inner products $\boldsymbol{x}_i \cdot \boldsymbol{x}_j$ between all pairs of training points)

Linear SVMs: Overview

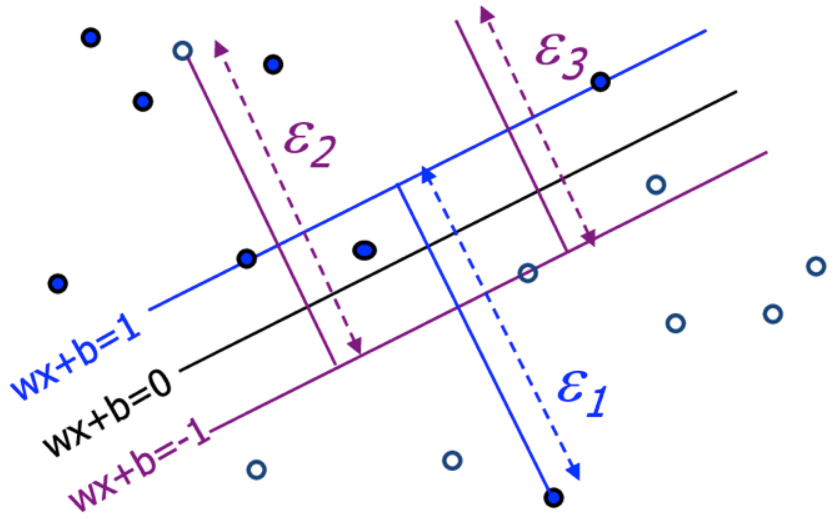
- The classifier is a *separating hyperplane.*
- Most "important" training points are support vectors; they define the hyperplane.
- Quadratic optimization algorithms can identify which training points x_i are support vectors with non-zero Lagrangian multipliers α_i.
- Both in the dual formulation of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

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

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

Soft Margin Classification

Slack variables ξ_i **can be added to allow misclassification of difficult or noisy examples.**

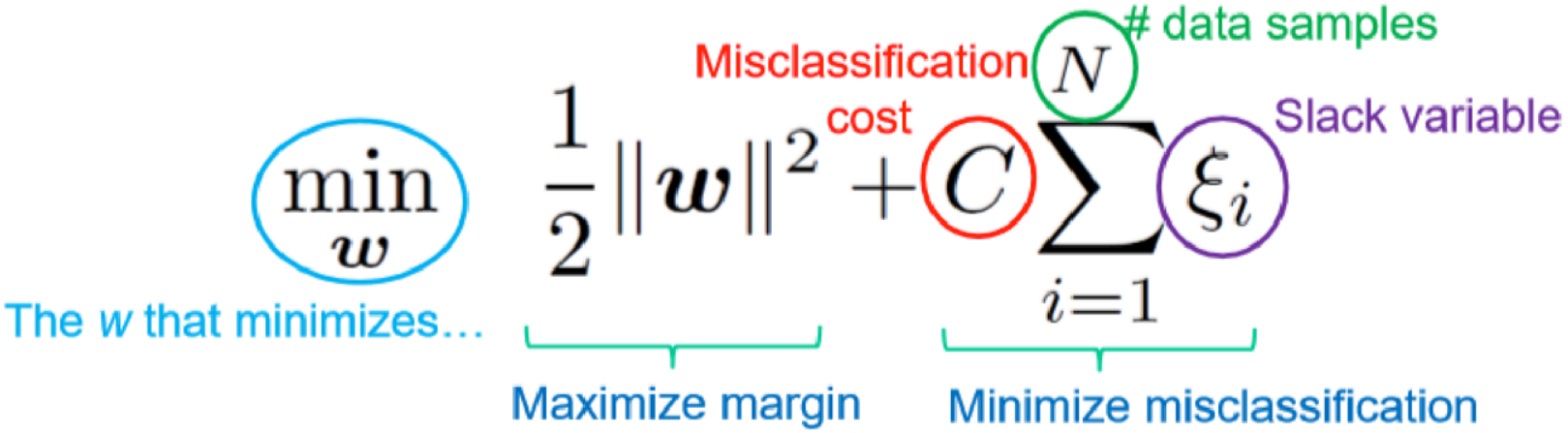


What should our quadratic optimization criterion be?

Minimize

$$\frac{1}{2}\mathbf{w}.\mathbf{w} + C\sum_{k=1}^{R}\varepsilon_k$$

Soft Margin



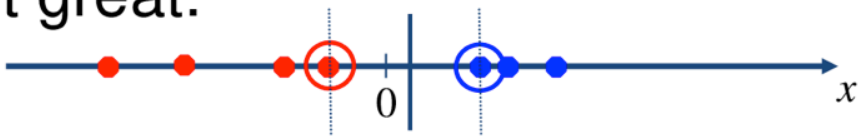
$$\min_{\boldsymbol{w}} \quad \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{N}\xi_i$$

$$\text{subject to} \quad y_i \boldsymbol{w}^T \boldsymbol{x}_i \geq 1 - \xi_i,$$

$$\xi_i \geq 0, \quad \forall i = 1, \ldots, N$$

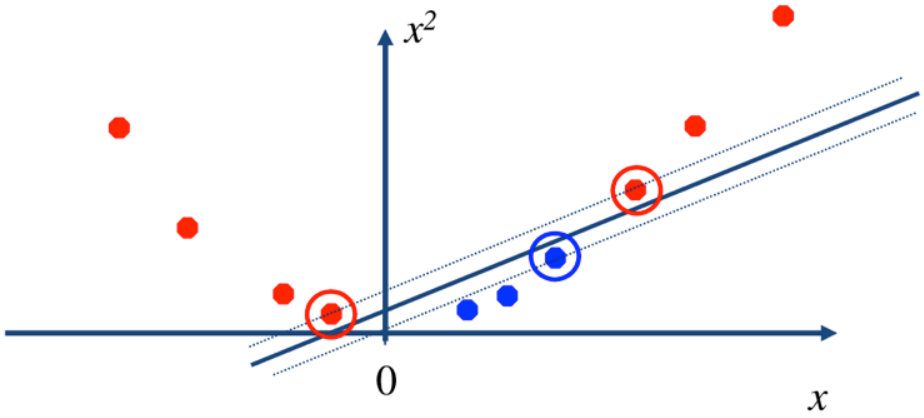
Non-linear SVMs

- Datasets that are linearly separable with some noise soft margin 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:



The "Kernel Trick"

- The linear classifier relies on dot product between vectors
  - $x_i^T \cdot x_j$
- If every data point is mapped into high-dimensional space via some transformation Φ: $x \rightarrow \varphi(x)$, the dot product becomes:

$$K(x_i, x_j) = \varphi(x_i)^T \varphi(x_j)$$

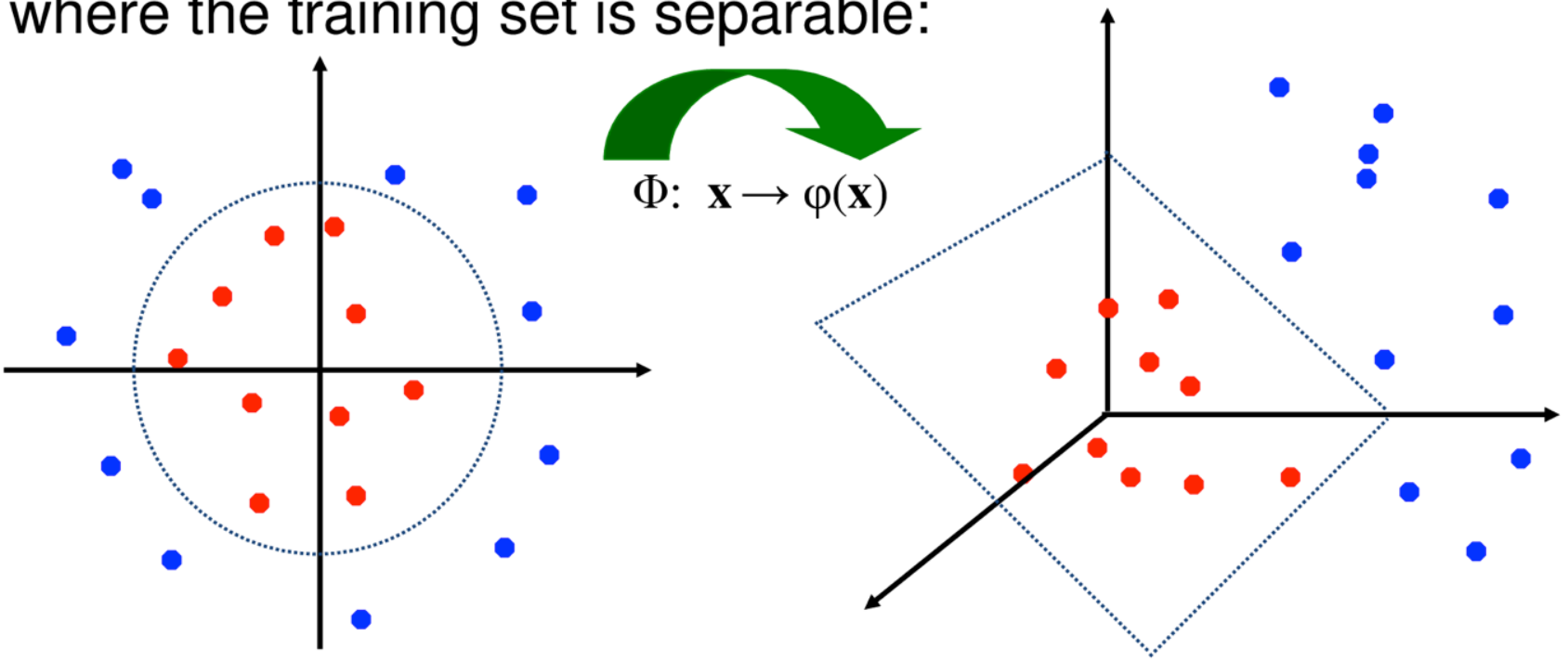
- A *kernel function* is some function that corresponds to an inner product in some expanded feature space.

SVM Kernels

- SVM algorithms use a set of mathematical functions that are defined as the kernel.
- Function of kernel is to take data as input and transform it into the required form.
- Different SVM algorithms use different types of kernel functions. Example *linear, nonlinear, polynomial, and sigmoid etc.*

Non-linear SVMs: Feature spaces

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



SVM – Overlapping Class Scenario

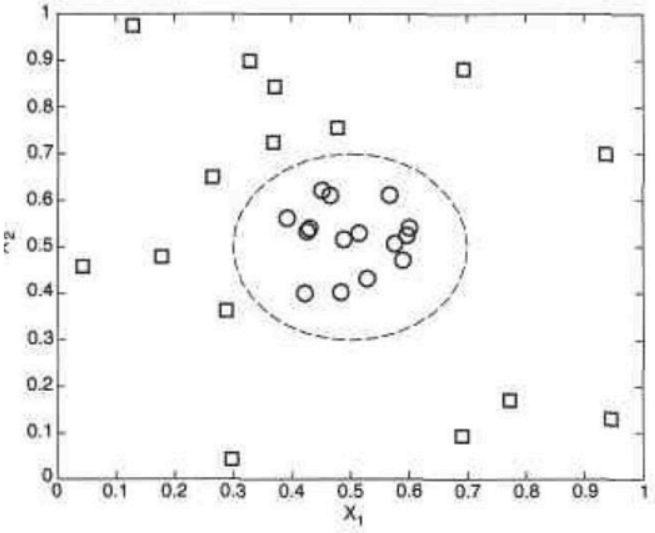
- Data is not separable linearly
- Margin will become inefficient
- Data needs to be transformed from original coordinate space **x** to a new space **Φ(x)**, so that linear decision boundary can be applied
- A non-linear transformation function is needed, like, ex:

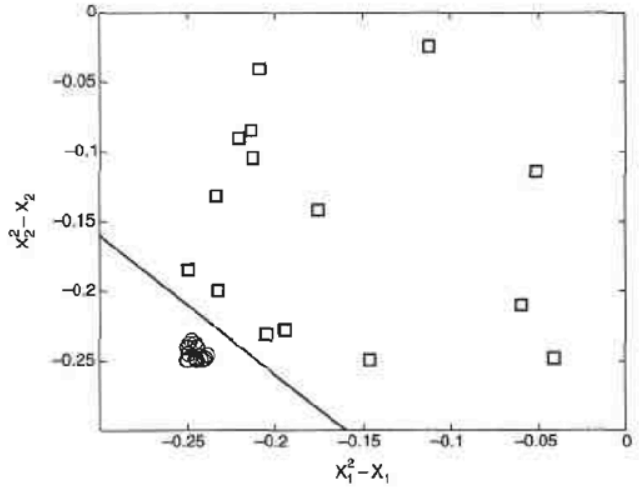
$$\Phi : (x_1, x_2) \longrightarrow (x_1^2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2, 1)$$

- In the transformed space we can choose w = (w_0, w_1, … , w_4) such that

$$w_4 x_1^2 + w_3 x_2^2 + w_2 \sqrt{2} x_1 + w_1 \sqrt{2} x_2 + w_0 = 0.$$

- The linear decision boundary in the transformed space has the following form: w. **Φ**(x) + b = 0





Non-linear SVMs Mathematically

- 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!

Non-linear SVM using kernel

1. Select a kernel function.
2. Compute pairwise kernel values between labeled examples.
3. Use this “kernel matrix” to solve for SVM support vectors & alpha weights.
4. To classify a new example: compute kernel values between new input and support vectors, apply alpha weights, check sign of output.

Nonlinear SVM - Overview

- SVM locates a separating hyperplane in the feature space and classify points in that space
- It does not need to represent the space explicitly, simply by defining a kernel function
- The kernel function plays the role of the dot product in the feature space.

XOR problem: Not linearly separable

For the XOR problem,

x_1	$(-1, -1)$	$\longrightarrow$	-1 y_1
x_2	$(-1, +1)$	$\longrightarrow$	$+1$ y_2
x_3	$(+1, -1)$	$\longrightarrow$	$+1$ y_3
x_4	$(+1, +1)$	$\longrightarrow$	-1 y_4

XOR problem: Choosing Kernel

Let us consider the XOR problem using SVMs (Cherkassy, 1998)

Start with a kernel

$$K(\underline{x}, \underline{x}_i) = (1 + \underline{x}^T \underline{x}_i)^2$$

$$\underline{x} = [x_1 \; x_2]^T \qquad \underline{x}_i = [x_{i1} \; x_{i2}]^T$$

$$K(\underline{x}, \underline{x}_i) = \left(1 + (x_1\ x_2) \cdot \begin{bmatrix} x_{i1} \\ x_{i2} \end{bmatrix}\right)^2$$

$$= \left(1 + x_1 x_{i1} + x_2 x_{i2}\right)^2$$

$$= 1 \cdot 1 + x_1^2 x_{i1}^2 + 2 x_1 x_2 x_{i1} x_{i2} + 2 x_1 x_{i1} + 2 x_2 x_{i2} + x_2^2 x_{i2}^2 + 2 x_1 x_{i1} + 2 x_2 x_{i2}$$

Let us express $K(\cdot,\cdot) = \langle \phi(\underline{x}), \phi(\underline{x}_i) \rangle$

$$\phi(x) = \left[1 \quad x_1^2 \quad \sqrt{2}\, x_1 x_2 \quad x_2^2 \quad \sqrt{2}\, x_1 \quad \sqrt{2}\, x_2\right]^T$$

$$\phi(\underline{x}_i) = \left[1 \quad x_{i1}^2 \quad \sqrt{2}\, x_{i1} x_{i2} \quad x_{i2}^2 \quad \sqrt{2}\, x_{i1} \quad \sqrt{2}\, x_{i2}\right]^T$$

348 / 348

For the XOR problem,

$(-1, -1) \longrightarrow -1$

$(-1, +1) \longrightarrow +1$

$(+1, -1) \longrightarrow +1$

$(+1, +1) \longrightarrow -1$

$K := \left[K(\underline{x}_i, \underline{x}_j) \right]$

$K(\underline{x}_i, \underline{x}_j) = \phi^T(\underline{x}_i)\,\phi(\underline{x}_j)$

Each $\underline{x}_i, \underline{x}_j$

$$K = \begin{bmatrix} 9 & 1 & 1 & 1 \\ 1 & 9 & 1 & 1 \\ 1 & 1 & 9 & 1 \\ 1 & 1 & 1 & 9 \end{bmatrix}$$

349 / 349

Dual of SVM

From the dual problem, the objective $Q(\underline{\alpha})$

$$Q(\underline{\alpha}) = \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 - \frac{1}{2}\Big(9\alpha_1^2 - 2\alpha_1\alpha_2 - 2\alpha_1\alpha_3 + 2\alpha_1\alpha_4 + 9\alpha_2^2 + 2\alpha_2\alpha_3 - 2\alpha_2\alpha_4 + 9\alpha_3^2 - 2\alpha_3\alpha_4 + 9\alpha_4^2 \Big)$$

$$\Big) \sum_{i=1}^{4}\sum_{j=1}^{4} \alpha_i \alpha_j y_i y_j K(\underline{x}_i, \underline{x}_j)$$

Solving for α

$$\frac{\partial Q(\underline{\alpha})}{\partial \alpha_i} = 0 \qquad i = 1, \cdots, 4$$

Doing this yields

$$\begin{cases} 9\alpha_1 - \alpha_2 - \alpha_3 + \alpha_4 = 1 \\ -\alpha_1 + 9\alpha_2 + \alpha_3 - \alpha_4 = 1 \\ -\alpha_1 + \alpha_2 + 9\alpha_3 - \alpha_4 = 1 \\ \alpha_1 - \alpha_2 - \alpha_3 + 9\alpha_4 = 1 \end{cases}$$

Solving this set of eqns

$$\alpha_{opt,i} = \frac{1}{8} \qquad i = 1, \cdots, 4$$

Now, all 4 inputs $\{\underline{x}_i\}_{i=1}^{4}$ are support vectors

$$Q(\alpha) = \frac{1}{4}$$

$$\frac{1}{2}\|\underline{w}_0\|^2 = \frac{1}{4} \Rightarrow \|\underline{w}_0\| = \frac{1}{\sqrt{2}}$$

$$\underline{w}_0 = \sum_{i=1}^{4} \alpha_{0,i}\, y_i\, \phi(\underline{x}_i)$$

$$= \frac{1}{8}\left[-\phi(\underline{x}_1) + \phi(\underline{x}_2) + \phi(\underline{x}_3) - \phi(\underline{x}_4)\right]$$

$$\underline{w}_0 = \left[0 \;\; 0 \;\; -\frac{1}{\sqrt{2}} \;\; 0 \;\; 0 \;\; 0\right]^T$$

For +ve support vectors we have

$$\sum_{i=1}^{4} \alpha_i y_i K(x_i, x_j) + b = +1 \quad (x_j \text{ is +ve support vector})$$

Put $x_j = x_2$ or x_4

$\alpha_i = \frac{1}{8} \quad \forall i = 1 \text{ to } 4$

$$K = \begin{bmatrix} 9 & 1 & 1 & 1 \\ 1 & 9 & 1 & 1 \\ 1 & 1 & 9 & 1 \\ 1 & 1 & 1 & 9 \end{bmatrix}$$

$$-\frac{K(x_1, x_2)}{8} + \frac{K(x_2, x_2)}{8} + \frac{K(x_3, x_2)}{8} - \frac{K(x_4, x_2)}{8} + b = 1$$

$$-\frac{1}{8} + \frac{9}{8} + \frac{1}{8} - \frac{1}{8} + b = 1$$

$$\boxed{b = 0}$$

Decision boundary of non linear SVM

$$w^T \phi(x) + b = 0$$

The opt. hyperplane is given by

$$\underline{w}_o^T \phi(\underline{x}) = 0$$

$$\begin{bmatrix} 0 & 0 & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ x_1^2 \\ \sqrt{2} x_1 x_2 \\ x_2^2 \\ \sqrt{2} x_1 \\ \sqrt{2} x_2 \end{bmatrix} = 0$$

$\Rightarrow$ $x_1 x_2 = 0$ is the decision boundary

Non Linear Decision boundary

$$y = -x_1 x_2$$

x		y
−1	−1	−1
−1	+1	+1
+1	−1	+1
+1	+1	−1