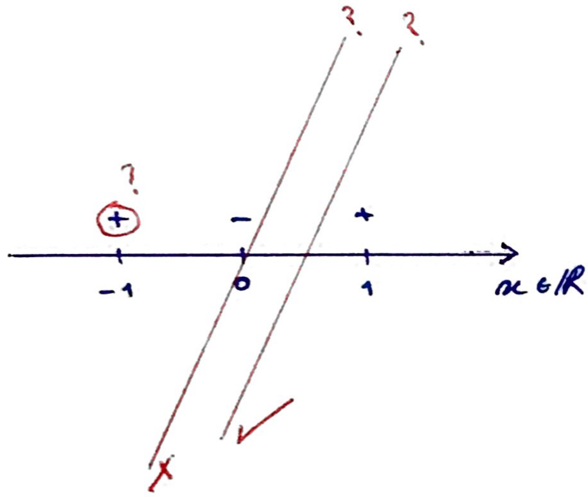
Unit 2 – Lecture 6: Nonlinear Classification

Objectives:

- Derive non-linear classifiers from feature maps
- Move from coordinate parameterization to weighting examples
- Compute Kernel functions induced from feature maps
- Use Kernel perceptron, Kernel linear regression
- Understand the properties of Kernel functions

Higher Order Feature vectors:



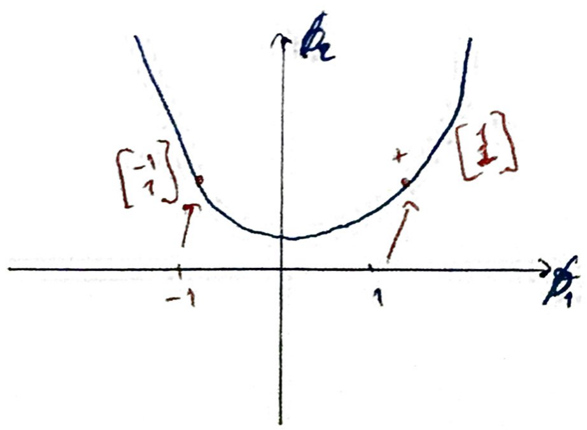
In this case we can't use linear classifiers to map the data.

We can remedy this situation by introducing a feature transformation feeding a different type of example to the linear classifier.

$$\underset{\in \mathbb{R}}{x} \longrightarrow \underset{\in \mathbb{R}^2}{\phi(x)} = \begin{bmatrix} x \\ x^2 \end{bmatrix} \leftarrow$$ we add additional feature like x^2

$$\theta \longrightarrow \theta = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}$$

$$h(x, \theta, \theta_0) = \text{sign}(\theta \cdot \phi(x) + \theta_0)$$
$$= \text{sign}(\theta_1 \cdot x + \theta_2 x^2 + \theta_0)$$



$$\phi(x) = \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix} = \begin{bmatrix} x \\ x^2 \end{bmatrix}$$ in this case

$$1 \longrightarrow \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$-1 \longrightarrow \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

A linear classifier in the new feature coordinate implies a non linear classifier in the x

Introduction to Non-linear Classification:

Non linear classification : $h(x; \theta, \theta_0) = \text{sign}(\theta \cdot \phi(x) + \theta_0)$

Non linear Regression : $f(x; \theta, \theta_0) = \theta \cdot \phi(x) + \theta_0$

By mapping input examples explicitly into feature vectors, and performing linear classification or regression on top of such feature vectors, we get a lot of expressive power.

But the downside is that these vectors can be quite high dimensional

Computational Efficiency:

Computing the inner product between two feature vectors can be cheap even if the vectors are very high dimensional.

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

Different examples x and x'

$$\phi(x') = [\underbrace{x_1', x_2'}, \underbrace{x_1'^2, \sqrt{2}\, x_1' x_2', x_2'^2}]^T$$

$$K(x, x') = \phi(x) \cdot \phi(x') = (x \cdot x') + (x \cdot x')^2$$

Kernel function

inner product or dot product

The Kernel can be evaluated very cheaply, even though we would have to explicitly construct very high dimensional feature vectors.

Task : linear methods $\longrightarrow$ methods that can operate in terms of kernels.

$$\text{Sign}(\theta \cdot \phi(x) + \theta_0) \longrightarrow K(x, x')$$

Kernel as Dot Products:

$$\phi(x) = [x_1, x_2, x_1^2, \sqrt{2} x_1 x_2, x_2^2] \qquad x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$\phi(x') = [x_1', x_2', x_1'^2, \sqrt{2} x_1' x_2', x_2'^2] \qquad x' = \begin{bmatrix} x_1' \\ x_2' \end{bmatrix}$$

$$\phi(x) \cdot \phi(x') = x_1 x_1' + x_2 x_2' + x_1^2 x_1'^2 + 2 x_1 x_2 x_1' x_2' + x_2^2 x_2'^2$$

$$= (x_1 x_1' + x_2 x_2') + (x_1 x_1' + x_2 x_2')^2$$

$$= x \cdot x' + (x \cdot x')^2$$

$$K(x, x') = \phi(x) \cdot \phi(x') = x_1 x_1' + x_2 x_2' + x_3 x_3' + x_2 x_3' + x_3 x_2' \leftarrow \phi(x) = [x_1, x_2 + x_3]$$

How the Kernel Perceptron Algorithm Works?

Perceptron:

$(\{(x^{(i)}, y^{(i)}), i = 1, \dots m\}, T)$:

initialize $\theta = 0$ (vector)

for $t = 1, \dots, T$

for $i = 1, \dots, m$

if $y^{(i)}(\theta \cdot x^{(i)}) \leq 0$

then update $\theta = \theta + y^{(i)} x^{(i)}$

Radial Basis Kernel:

$$K(x, x') = \exp\left(-\frac{1}{2} \|x - x'\|^2\right)$$

We introduced that we can always express θ as: $\theta = \sum_{j=1}^{m} \alpha_j y^{(j)} \phi(x^{(j)})$

Kernel Perceptron:

$(\{(x^{(i)}, y^{(i)}), i = 1, \dots, m\}, T\}$

initialize $\alpha_1, \alpha_2, \dots, \alpha_m$ to some values $\rightarrow \theta = 0 \Leftrightarrow$ setting $\alpha_j = 0$ for all j

for $t = 1, \dots T$

for $i = 1, \dots m$

if (Mistake Condition Expressed in α_j) $\rightarrow y^{(i)}(\theta \cdot \phi(x^{(i)})) \leq 0 \Leftrightarrow y^{(i)} \sum_{j=1}^{m} \alpha_j y^{(j)} K(x^{(j)}, x^{(i)}) \leq$

Update α_j appropriately $\rightarrow \alpha_i = \alpha_i + 1 \Leftrightarrow \theta = \theta + y^{(i)} \phi(x^{(i)})$

Kernel Composition Rules:

$K(x, x') = 1$ is a kernel fuction $\phi(x) = 1$

$\hat{K}(x, x') = f(x) K(x, x') f(x')$ is also a kernel $\hat{\phi}(x) = f(x)\phi(x)$

$K(x, x') = K_1(x, x') + K_2(x, x')$ is a Kernel

$K(x, x') = K_1(x, x') K_2(x, x')$ is a kernel

Kernel Composition Rule 1:

$\hat{K}(x, x') = f(x) K(x, x') f(x')$

if there exists $\phi(x)$ such that: $K(x, x') = \phi(x) \cdot \phi(x')$

which of the following ϕ gives $\hat{K}(x, x') = \phi(x) \cdot \phi(x')$?

$(f(x)\phi(x)) \cdot (f(x')\phi(x')) = \hat{K}(x, x') \Rightarrow \phi(x) = f(x)\phi(x)$