**Alin Tomescu**

6.867 Machine learning | Week 3, Tuesday, September 17th, 2013| Lecture 4

# Lecture 4: Non-linear classifiers

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \to \phi(x) = \begin{bmatrix} x_1^2 \\ x_2^2 \\ x_1 x_2 \sqrt{2} \\ x_1 \\ x_2 \end{bmatrix} \in \mathbb{R}^5$$

Initially we have $h(x;\theta) = sign(\theta \cdot x), \theta \in \mathbb{R}^2$, as a linear classifier.

Non-linear: $h(x;\theta) = sign\big(\theta \cdot \phi(x)\big), \theta \in \mathbb{R}^5$

We'll look at infinite dimensional feature mappings.

**Key idea:** looking at the feature vectors explicitly is not naturally feasible when they grow to high dimensions. We want to exploit the feature mapping $\phi(x)$, which may be difficult to evaluate explicitly, but the inner products $\phi(x) \cdot \phi(x')$ could be easier to evaluate.

**Intuition:** If $x$ and $x'$ are very similar then $\phi(x)$ and $\phi(x')$ will be pointing in the same direction and therefore their inner product will be larger. In contrast, if $x$ and $x'$ are very dissimilar then $\phi(x)$ and $\phi(x')$ may be pointing in different directions and so their inner product may be small. Not rigorous, but useful to think about. This isn't always true though.

How do you come up with a kernel function? Find a $K(x, x')$ that is large if $x$ and $x'$ are similar and small otherwise. This isn't always true though.

Inner products are referred to as kernels: $\phi(x) \cdot \phi(x') = (x \cdot x')^2 + (x \cdot x')$

$$\phi_A(\vec{x}) = \prod_{i \in A} x_i \,, \vec{x} \in \mathbb{R}^d, A \subset \{1 \dots d\}$$

$$\phi_A(x) = index\ into\ \phi(x)\ at\ positions\ in\ A$$

$$\phi(x) \cdot \phi(x') = \sum_{A \subset \{1 \dots d\}} \phi_A(x)\phi_A(x') = \prod_{i=1}^{d} (1 + x_i x_i')$$

Turn perceptron algorithm into a form that only relies on inner product.

$$kernel\ perceptron:$$

$$\theta^{(0)} = 0$$

$$cycle\ through\ training\ examples \dots any\ order$$

$$if\ y^{(i)}\theta^{(k)}\phi\big(x^{(i)}\big) \le 0, then\ \theta^{(k+1)} = \theta^{(k)} + y^{(i)}\phi\big(x^{(i)}\big)$$

After $k$ updates the parameter vector will be:

$$\alpha_i = \#\ of\ updates\ to\ \theta\ after\ step\ i$$

**Alin Tomescu**
6.867 Machine learning | Week 3, Tuesday, September 17th, 2013| Lecture 4

$$\theta^{(k)} = \sum_{i=1}^{n} \alpha_i y^{(i)} \phi(x^{(i)}), \alpha_i \geq 0$$

$$\sum_{i=1}^{n} \alpha_i = k$$

How do we classify $h(x; \theta^{(k)}) = sign\left(\theta^{(k)}\phi(x)\right) = sign\left(\left(\sum_{j=1}^{n} \alpha_i y^{(j)} \phi(x^{(j)})\right)\phi(x)\right) = h(x; \alpha)$

Let's rewrite **kernel perceptron** in terms of $\alpha$

$$\alpha_i = 0, \forall i$$

$$\text{if } y^{(i)}\left(\left(\sum_{j=1}^{n} \alpha_j y^{(j)} \phi(x^{(j)})\right)\phi(x^{(i)})\right) \leq 0, \text{then } \alpha_i = \alpha_i + 1$$

$$\phi(x^{(j)})\phi(x^{(i)}) = K(x^{(i)}, y^{(j)})$$

**Explicit definition:** A kernel function $K(x, x') = \phi(x)\phi(x')$

**Implicit definition:** $K(x, x')$ is a kernel if for all training examples and $\forall n$, if the $n \times n$ matrix $K_{i,j} = K(x^{(i)}, x^{(j)})$ is **positive semi-definite** ($a^T Ka \geq 0, \forall\ nonzero\ a$).

Example for $K(x, x') = xx'$

$$a^T Ka = \sum_{i=0}^{n}\sum_{j=0}^{n} a_i K_{i,j} a_i = \sum_{i=0}^{n}\sum_{j=0}^{n} a_i \phi(x^{(i)})\phi(x^{(j)}) a_i = \sum_{i=0}^{n}\sum_{j=0}^{n} a_i \sum_{k} \phi(x^{(i)})_k \phi(x^{(j)})_k a_i$$
$$= \sum_{k}\sum_{i=0}^{n}\sum_{j=0}^{n} a_i \phi(x^{(i)})_k \phi(x^{(j)})_k a_i = \sum_{k}\left(\sum_{i=0}^{n} a_i \phi(x^{(i)})_k\right)^2 \geq 0$$

**Definition:** $M$ is a symmetric matrix if it's equal to its transpose: $M^T = M$

**Mercer Theorem:** Let $K(x, z)$ e given, then $K$ is a valid kernel (a.k.a. a Mercer kernel), (i.e. $\exists \phi$ s.t. $K(x, z) = \phi(x)^T \phi(z)$) if and only if for any set of $m < \infty$ examples (or points) $\{x^{(1)}, \dots, x^{(n)}\}$, it holds true that $K \in R^{m \times n}$ is symmetric positive semi-definite.

## Composition rules for kernels

Assume $k_1(x, x'), k_2(x, x')$ are kernels and fix $f: X \to R$

(0) $k(x, x') = 1$ is a kernel
(1) $k(x, x') = f(x)k_1(x, x')f(x')$ is a kernel
(2) $k(x, x') = k_1(x, x') + k_2(x, x')$ is a kernel
(3) $k(x, x') = k_1(x, x') \times k_2(x, x')$ is a kernel

$k(x, x') = x_l x_l'$ is valid kernel by 0 and 1, $f(x) = x_l$

$k(x, x') = \sum_{l=1}^{n} x_l \cdot x_l' = xx'$ is valid kernel by (0), (1) and (2), with $f(x) = x_l$

If, the features for these kernels are

$$k_1 : \phi^{(1)}(x)$$

$$k_2 : \phi^{(2)}(x)$$

What is the feature mapping for $k(x, x') = k_1(x, x') + k_2(x, x')$? $k : \phi(x) = \begin{bmatrix} \phi^{(1)}(x) \\ \phi^{(2)}(x) \end{bmatrix}$ because:

$$k(x, x') = \begin{bmatrix} \phi^{(1)}(x) \\ \phi^{(2)}(x) \end{bmatrix}^T \begin{bmatrix} \phi^{(1)}(x') \\ \phi^{(2)}(x') \end{bmatrix} = \left(\phi^{(1)}(x)\phi^{(1)}(x')\right) + \left(\phi^{(2)}(x)\phi^{(2)}(x')\right) = k_1(x, x') + k_1(x, x')$$

## Radial basis kernel

$$k(x, x') = e^{-\frac{1}{2}\|x - x'\|^2}$$

Perceptron algorithm will always converge using this kernel.

$$K(x, x') = e^{-\frac{1}{2}\|x - x'\|^2} = e^{-\frac{1}{2}\|x\|^2} e^{xx'} e^{-\frac{1}{2}\|x'\|^2}$$

$$f(x) = e^{-\frac{1}{2}\|x\|^2}$$

$$e^{xx'} = 1 + xx' + \frac{1}{2!}xx'^2 + \cdots = is\ also\ a\ kernel$$

For any kernel:

$$K(x, x) = \|\phi(x)\|^2 \geq 0$$

$$K(x, x') = K(x', x)$$

$$h(x; \alpha) = sign\left(\left(\sum_{j=1}^{n} \alpha_i y^{(j)} \phi\left(x^{(j)}\right)\right) \phi(x)\right) = sign\left(\sum_{j=1}^{n} \alpha_i y^{(j)} K\left(x^{(j)}, x\right)\right)$$

$$Let\ F(x) = \sum_{j=1}^{n} \alpha_i y^{(j)} K\left(x^{(j)}, x\right)$$

**TODO:** Refer to **Figure 4.1** in notebook

Does there exist a $\phi$, such that $K(x, z) = e^{-\frac{\|x - x'\|^2}{2\sigma^2}}$ ?

## Support vector machines with kernels

By replacing all occurrences of $x^{(i)} \in \mathbb{R}^n$ with $\phi\left(x^{(i)}\right)$ we are mapping the training examples into a higher dimensional space $V$ and finding a linear separator $\theta$ in that higher-dimensional space $V$. If we convert back to $\mathbb{R}^n$, the separator will look non-linear. This allows us to separate non-linearly separable data, by mapping the data as linearly separable in $V$.

**Alin Tomescu**

6.867 Machine learning | Week 3, Tuesday, September 17th, 2013| Lecture 4

$\min_{\theta} \frac{1}{2}\|\theta\|^2$ such that $y^{(i)}\theta\phi\left(x^{(i)}\right) \geq 1, \forall i$

$$L(\theta, \alpha) = \frac{1}{2}\|\theta\|^2 - \sum_{i=1}^{n} \alpha_i \left(y^{(i)}\theta\phi\left(x^{(i)}\right) - 1\right)$$

$$\alpha_i \geq 0$$