CSE 151 Machine Learning
Instructor: Kamalika Chaudhuri
Announcements

- Midterm 2 is on Mon May 21
- HW2 has been graded
- Infosession on machine learning jobs at Yahoo on Monday May 14, 4-5pm, CSE 4140
Feature Maps

Data which is not linearly separable may be linearly separable in another feature space.

In the figure, data is linearly separable in z-space:

$$(Z_1, Z_2, Z_3) = (X_1^2, 2X_1X_2, X_2^2)$$
Let $\phi(x)$ be a feature map. For example: $\phi(x) = (x_1^2, x_1x_2, x_2^2)$

We want to run perceptron on the feature space $\phi(x)$
Feature Maps

Let \( \phi(x) \) be a feature map. For example: \( \phi(x) = (x_1^2, x_1, x_2) \)

We want to run perceptron on the feature space \( \phi(x) \)

Many linear classification algorithms, including perceptron, SVMs, etc can be written in terms of dot-products \( \langle x, z \rangle \)

We can simply change these dot products to \( \langle \phi(x), \phi(z) \rangle \)

For a feature map \( \phi \) we define the corresponding kernel \( K \):

\[
K(x, y) = \langle \phi(x), \phi(y) \rangle
\]

Thus we can write the perceptron algorithm in terms of \( K \). Computing \( K \) directly is often faster than computing the map.
Kernels

For a feature map $\phi$ we define the corresponding kernel $K$:

$$K(x, y) = \langle \phi(x), \phi(y) \rangle$$

Examples:

1. $K(x, z) = (\langle x, z \rangle)^2$
Kernels

For a feature map $\phi$ we define the corresponding kernel $K$:

$$K(x, y) = \langle \phi(x), \phi(y) \rangle$$

**Examples:**

1. $K(x, z) = (\langle x, z \rangle)^2$

Suppose $d=2$, and: $\phi(x) = [\phi_1(x), \phi_2(x), \phi_3(x), \ldots]$

$$\langle \phi(x), \phi(z) \rangle = \phi_1(x)\phi_1(z) + \phi_2(x)\phi_2(z) + \ldots$$
For a feature map \( \phi \) we define the corresponding kernel \( K \):

\[
K(x, y) = \langle \phi(x), \phi(y) \rangle
\]

Examples:

1. \( K(x, z) = (\langle x, z \rangle)^2 \)

Suppose \( d=2 \), and: \( \phi(x) = [\phi_1(x), \phi_2(x), \phi_3(x), \ldots] \)

\[
\langle \phi(x), \phi(z) \rangle = \phi_1(x)\phi_1(z) + \phi_2(x)\phi_2(z) + \ldots
\]

\[
(\langle x, z \rangle)^2 = (x_1 z_1 + x_2 z_2)^2
\]
Kernels

For a feature map \( \phi \) we define the corresponding kernel \( K \):

\[
K(x, y) = \langle \phi(x), \phi(y) \rangle
\]

Examples:

1. \( K(x, z) = (\langle x, z \rangle)^2 \)

Suppose \( d=2 \), and: \( \phi(x) = [\phi_1(x), \phi_2(x), \phi_3(x), \ldots] \)

\[
\langle \phi(x), \phi(z) \rangle = \phi_1(x)\phi_1(z) + \phi_2(x)\phi_2(z) + \ldots
\]

\[
(\langle x, z \rangle)^2 = (x_1z_1 + x_2z_2)^2
\]

\[
= x_1^2z_1^2 + x_1x_2z_1z_2 + x_1x_2z_1z_2 + x_2^2z_2^2
\]
Kernels

For a feature map $\phi$ we define the corresponding kernel $K$:

$$K(x, y) = \langle \phi(x), \phi(y) \rangle$$

Examples:

1. $K(x, z) = (\langle x, z \rangle)^2$

Suppose $d=2$, and:

$$\phi(x) = [\phi_1(x), \phi_2(x), \phi_3(x), \ldots]$$

$$\langle \phi(x), \phi(z) \rangle = \phi_1(x)\phi_1(z) + \phi_2(x)\phi_2(z) + \ldots$$

$$(\langle x, z \rangle)^2 = (x_1z_1 + x_2z_2)^2$$

$$= \begin{bmatrix} 2 \\ 1 \end{bmatrix} x_1^2 z_1^2 + \begin{bmatrix} 1 \end{bmatrix} x_1 x_2 z_1 z_2 + \begin{bmatrix} 1 \end{bmatrix} x_1 x_2 z_1 z_2 + \begin{bmatrix} 2 \\ 2 \end{bmatrix} x_2^2 z_2^2$$

$$\phi_1(x)\phi_1(z) \quad \phi_2(x)\phi_2(z) \quad \phi_3(x)\phi_3(z) \quad \phi_4(x)\phi_4(z)$$
Kernels

For a feature map $\phi$ we define the corresponding kernel $K$:

$$K(x, y) = \langle \phi(x), \phi(y) \rangle$$

Examples:

1. $K(x, z) = (\langle x, z \rangle)^2$

Suppose $d=2$, and: $\phi(x) = [\phi_1(x), \phi_2(x), \phi_3(x), \ldots]$

$$\langle \phi(x), \phi(z) \rangle = \phi_1(x)\phi_1(z) + \phi_2(x)\phi_2(z) + \ldots$$

$$\langle x, z \rangle^2 = (x_1z_1 + x_2z_2)^2$$

$$= x_1^2z_1^2 + x_1x_2z_1z_2 + x_1x_2z_1z_2 + x_2^2z_2^2$$

$\phi_1(x)\phi_1(z) \quad \phi_2(x)\phi_2(z) \quad \phi_3(x)\phi_3(z) \quad \phi_4(x)\phi_4(z)$

Feature map for $K$: $\phi(x) = [x_1^2, x_1x_2, x_1x_2, x_2^2]$
Kernels

For a feature map $\phi$ we define the corresponding kernel $K$:

$$K(x, y) = \langle \phi(x), \phi(y) \rangle$$

Examples:

1. $K(x, z) = (\langle x, z \rangle)^2$

   For more general $d$:

   $$K(x, z) = \left( \sum_{i=1}^{d} x_i z_i \right) \left( \sum_{i=1}^{d} x_i z_i \right) = \sum_{i=1}^{d} \sum_{j=1}^{d} x_i x_j z_i z_j = \sum_{i,j=1}^{d} (x_i x_j)(z_i z_j)$$

   For $d=3, \phi(x) = [x_1^2, x_1 x_2, x_1 x_3, x_2 x_1, x_2^2, x_2 x_3, x_3 x_1, x_3 x_2, x_3^2]^T$

   Time to compute $K$ directly = $O(d)$
   Time to compute $K$ though feature map = $O(d^2)$
Kernels

For a feature map \( \phi \) we define the corresponding kernel \( K \):

\[
K(x, y) = \langle \phi(x), \phi(y) \rangle
\]

Examples:

1. \( K(x, z) = (\langle x, z \rangle)^2 \)

2. \( K(x, z) = (\langle x, z \rangle + c)^2 \)

For \( d=2 \):

\[
(\langle x, z \rangle + c)^2 = (\langle x, z \rangle)^2 + 2c\langle x, z \rangle + c^2
\]
Kernels

For a feature map $\phi$ we define the corresponding kernel $K$:

$$K(x, y) = \langle \phi(x), \phi(y) \rangle$$

Examples:

1. $K(x, z) = (\langle x, z \rangle)^2$

2. $K(x, z) = (\langle x, z \rangle + c)^2$

For $d=2$:

$$(\langle x, z \rangle + c)^2 = (\langle x, z \rangle)^2 + 2c\langle x, z \rangle + c^2$$

$$= x_1^2z_1^2 + x_1x_2z_1z_2 + x_1x_2z_1z_2 + x_2^2z_2^2 + 2cx_1z_1 + 2cx_2z_2 + c^2$$
Kernels

For a feature map $\phi$ we define the corresponding kernel $K$:

$$K(x, y) = \langle \phi(x), \phi(y) \rangle$$

Examples:

1. $K(x, z) = (\langle x, z \rangle)^2$

2. $K(x, z) = (\langle x, z \rangle + c)^2$

For $d=2$:

$$(\langle x, z \rangle + c)^2 = (\langle x, z \rangle)^2 + 2c\langle x, z \rangle + c^2$$

$$= x_1^2z_1^2 + x_1x_2z_1z_2 + x_1x_2z_1z_2 + x_2^2z_2^2 + 2cx_1z_1 + 2cx_2z_2 + c^2$$

Like previous example
For a feature map $\phi$ we define the corresponding kernel $K$:

$$K(x, y) = \langle \phi(x), \phi(y) \rangle$$

**Examples:**

1. $K(x, z) = (\langle x, z \rangle)^2$

2. $K(x, z) = (\langle x, z \rangle + c)^2$

For $d=2$:

$$(\langle x, z \rangle + c)^2 = (\langle x, z \rangle)^2 + 2c\langle x, z \rangle + c^2$$

$$= x_1^2 z_1^2 + x_1 x_2 z_1 z_2 + x_1 x_2 z_1 z_2 + x_2^2 z_2^2 + 2c x_1 z_1 + 2c x_2 z_2 + c^2$$

Like previous example  $\phi_5(x)\phi_5(z)$  $\phi_6(x)\phi_6(z)$  $\phi_7(x)\phi_7(z)$
Kernels

For a feature map $\phi$ we define the corresponding kernel $K$:

$$K(x, y) = \langle \phi(x), \phi(y) \rangle$$

**Examples:**

1. $K(x, z) = (\langle x, z \rangle)^2$

2. $K(x, z) = (\langle x, z \rangle + c)^2$

For $d=2$:

$$(\langle x, z \rangle + c)^2 = (\langle x, z \rangle)^2 + 2c\langle x, z \rangle + c^2$$

$$= x_1^2z_1^2 + x_1x_2z_1z_2 + x_1x_2z_1z_2 + x_2^2z_2^2 + 2cx_1z_1 + 2cx_2z_2 + c^2$$

Like previous example

$\phi(x) = [x_1^2, x_1x_2, x_1x_2, x_2^2, \sqrt{2}cx_1, \sqrt{2}cx_2, c]$
Kernels

For a feature map $\phi$ we define the corresponding kernel $K$:

$$K(x, y) = \langle \phi(x), \phi(y) \rangle$$

Examples:

1. $K(x, z) = (\langle x, z \rangle)^2$

2. $K(x, z) = (\langle x, z \rangle + c)^2$

More general $d$:

$$\phi(x) = [x_ix_j, 1 \leq i, j \leq d, \sqrt{2}cx_i, 1 \leq i \leq d, c]$$
Kernels

For a feature map $\phi$ we define the corresponding kernel $K$:

$$K(x, y) = \langle \phi(x), \phi(y) \rangle$$

Examples:

1. $K(x, z) = (\langle x, z \rangle)^2$

2. $K(x, z) = (\langle x, z \rangle + c)^2$

3. $K(x, z) = (\langle x, z \rangle + c)^d$
Kernels

For a feature map $\phi$ we define the corresponding kernel $K$:

$$K(x, y) = \langle \phi(x), \phi(y) \rangle$$

Examples:

1. $K(x, z) = (\langle x, z \rangle)^2$

2. $K(x, z) = (\langle x, z \rangle + c)^2$

3. $K(x, z) = (\langle x, z \rangle + c)^d$

4. $K(x, z) = \exp \left( -\|x - z\|^2 / c^2 \right)$  
   Corresponds to an infinite dimensional feature map!
Kernels

For a feature map \( \phi \) we define the corresponding kernel \( K \):
\[
K(x, y) = \langle \phi(x), \phi(y) \rangle
\]

Examples:

1. \( K(x, z) = (\langle x, z \rangle)^2 \)

2. \( K(x, z) = (\langle x, z \rangle + c)^2 \)

3. \( K(x, z) = (\langle x, z \rangle + c)^d \)

4. \( K(x, z) = \exp \left( -\|x - z\|^2 / c^2 \right) \)  
   Corresponds to an infinite dimensional feature map!

5. String kernels(#)common words in a string), graph kernels, etc
Kernels

For a feature map \( \phi \) we define the corresponding kernel \( K \):

\[
K(x, y) = \langle \phi(x), \phi(y) \rangle
\]

All functions are not valid kernels though!
Given a function $K(x, z)$, how can we tell if it is a valid kernel?

Two properties:

1. **Symmetry:** For all $x, z$, $K(x, z) = K(z, x)$

   For any set $x^1, \ldots, x^m$, define kernel matrix $K_{m \times m}$ as:
   
   $$K_{ij} = K(x^i, x^j)$$

2. **Condition:** For all $x^1, \ldots, x^m$, and all $m$-dimensional vectors $t$,
   
   $$t^\top K t \geq 0$$

These conditions are **necessary and sufficient**!

Verify that conditions are necessary
Verify conditions hold for: \[ K(x, z) = (\langle x, z \rangle)^2 \]
Not A Kernel

\[ K(x, z) = -\langle x, z \rangle \] is not a kernel. Why?

Let \( x \) be any non-zero vector.
Kernel matrix of \( x \) is a single scalar:
\[ K = [-\langle x, x \rangle] = [-\|x\|^2] \]

For any 1x1 vector \( t \),
\[ t^\top K t = -t^2\|x\|^2 < 0 \]
Kernelizing Perceptron

**Perceptron Algorithm:**

1. Initially, \( w_1 = y_1 x_1 \)
2. For \( t = 2, 3, \ldots \)
   
   If \( y_t \langle w_t, x_t \rangle \leq 0 \) then:
   
   \[ w_{t+1} = w_t + y_t x_t \]

   Else:
   
   \[ w_{t+1} = w_t \]

Initially, store \((x_1, y_1)\)

If \( w_t \) makes a mistake then store \((x_t, y_t)\)

Calculate \( K(w_t, x) \) as:

\[ y_1 K(x_1, x) + y_{i1} K(x_{i1}, x) + \ldots + y_{ir} K(x_{ir}, x) \]

If we make mistakes on \((x_{i1}, y_{i1}), \ldots, (x_{ir}, y_{ir})\), then:

\[ w_t = y_1 x_1 + y_{i1} x_{i1} + \ldots + y_{ir} x_{ir} \]
Summary

- Linear Classification
- Perceptron algorithm
- Kernels (for non-linearity)
Ensemble Learning

How to combine multiple classifiers into a single one

Works well if the classifiers are complementary

This class: two types of ensemble methods:
  Bagging
  Boosting
Bagging (Bootstrap AGgregatING)

**Input:** $n$ labelled training examples $(x_i, y_i), i = 1,..,n$

**Algorithm:**

Repeat $k$ times:
- Select $m$ samples out of $n$ **with replacement** to get training set $S_i$
- Train classifier (decision tree, $k$-NN, perceptron, etc) $h_i$ on $S_i$

**Output:** Classifiers $h_1, .., h_k$

**Classification:** On test example $x$, output majority($h_1, .., h_k$)
Example

**Input:** n labelled training examples \((x_i, y_i), i = 1, \ldots, n\)

**Algorithm:**

Repeat k times:

- Select m samples out of n with replacement to get training set \(S_i\)
- Train classifier (decision tree, k-NN, perceptron, etc) \(h_i\) on \(S_i\)

**How to pick m?**

- Popular choice: \(m = n\)
- Still different from working with entire training set. Why?
Bagging

**Input:** n labelled training examples $S = \{(x_i, y_i)\}, i = 1, \ldots, n$

Suppose we select n samples out of n **with replacement** to get training set $S_i$

Still different from working with entire training set. Why?

$$\Pr(S_i = S) = \frac{n!}{n^n} \quad \text{(tiny number, exponentially small in n)}$$

$$\Pr((x_i, y_i) \text{ not in } S_i) = \left(1 - \frac{1}{n}\right)^n \approx e^{-1}$$

For large data sets, about 37% of the data set is left out!