CS:E4830 Kernel Methods in Machine Learning

Lecture 1 : Basics and Introduction to Kernel Methods

Rohit Babbar

9th January, 2019

Course Format

Format and Logistics

- Location - Lecture hall T1 for atleast first four lectures
- 12 lectures in total, 5 Credits
- 4 assignments and one final exam
- Two ways to pass the course
  - Assignments + Exam
  - Exam only (more details on mycourses page)

Personnel

- Solutions sessions with TAs
  - TAs - Sandor Szedmak, Viivi Uurtio, and Eric Bach
  - Python tutorial with Eric Bach on 17th January
- Doubts related to
  - Assignments - Please use Mycourses forum to send your doubts related to assignments, TAs will get back to you
  - Lecture material - Send me an email rohit.babbar@aalto

Mandatory

- Basics and Theory
  - Kernel Methods - Introduction
  - Reproducing kernel Hilbert Space
  - Learning theory and Generalization

Mandatory

- Basics and Theory
  - Kernel Methods - Introduction
  - Reproducing kernel Hilbert Space
  - Learning theory and Generalization
- Optimization and Support vector Machines
  - Convex Optimization and Duality
  - Optimization for supervised Kernel based algorithms
  - Optimization for Large-scale linear classification

About the Course - 2

- Algorithms
  - Unsupervised Learning and and kernel variants
  - (Kernel) Canonical Correlation Analysis (CCA)
  - Specific problem settings - Ranking, Structured prediction, Learning on graphs

About the Course - 2

- Algorithms
  - Unsupervised Learning and and kernel variants
  - (Kernel) Canonical Correlation Analysis (CCA)
  - Specific problem settings - Ranking, Structured prediction, Learning on graphs
- Advanced topics
  - Speeding up kernel Methods
  - Advanced topics 1/Paper reading
  - Advanced topics 2/Paper reading

Nature of the course

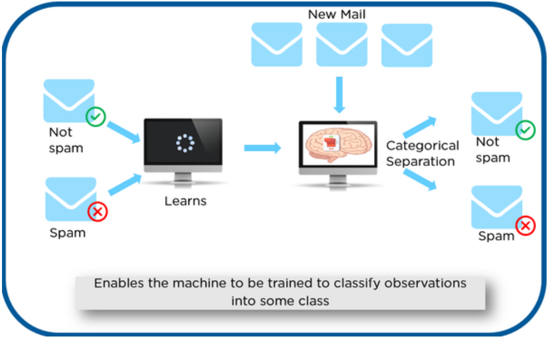
- Nature of the course - **Theoretical** (Mathematical) and **Algorithmic**
- Prerequisites
  - Machine Learning Basic Principles (or an equivalent course) - **required**
  - Linear Algebra and Basics of Probability/Statistics - **required**
  - Optimization or Signal Processing - desirable
  - Programming in Python - desirable

Outline for Today

Introduction

Supervised Learning refers to a learning process which looks at annotated data to then automatically annotate *similar* un-annotated data

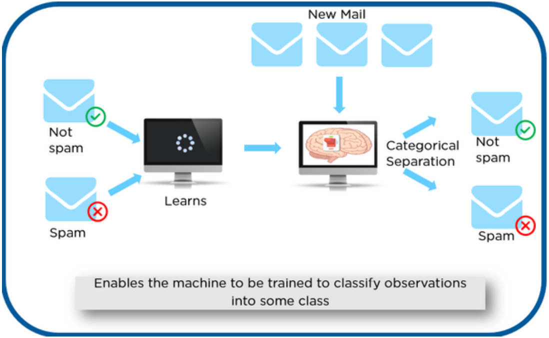
- Example - Spam vs non-spam classification by your email software [1]



[1] picture from Quora

Supervised Learning refers to a learning process which looks at annotated data to then automatically annotate *similar* un-annotated data

- Example - Spam vs non-spam classification by your email software [1]



- What makes the above statisical learning paradigm so powerful compared to building hand-crafted rules for the same task?

[1] picture from Quora

What is Machine Learning paradigm?

Learning from data

- The ML paradigm can adapt itself to data
- Furthermore, it can adapt itself to various tasks, feed different data from a different task to the **same learning algorithm**
  - Get a different classifier for a different task altogether
  - For example - Feed in dogs vs cats image data to the **same learning algorithm**

What is Machine Learning paradigm?

Learning from data

- The ML paradigm can adapt itself to data
- Furthermore, it can adapt itself to various tasks, feed different data from a different task to the **same learning algorithm**
  - Get a different classifier for a different task altogether
  - For example - Feed in dogs vs cats image data to the **same learning algorithm**
- This is not possible in hand-crafted rule based system
  - For instance, rules for spam detection are very different from those of image classification
- Elimination (or reduced dependency on domain expert) for hand-crafting rules

Shortcoming of Machine Learning paradigm?

Corner cases

- In the **classical programming** paradigm, we tell the system how to handle each (base or corner) case explicitly
- Under the **machine learning** paradigm - corner case might occur infrequently, and hence the system may be unreliable for such cases
- Arguably, a good practical and engineering system is an appropriate combination of both the approaches

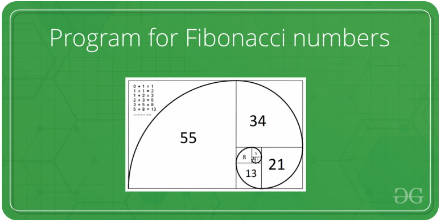


Figure: from GeeksforGeeks

Classification - Supervised Learning

Setup for first few lectures (Recall **Machine Learning Basic Principles**)

- Given a training set $\{(x_i, y_i)\}_{i=1}^{n}$ which is sampled i.i.d from a fixed distribution $\mathcal{D}$
- Learn a classifier which predicts the class $\hat{y}$ for the novel (test) instance x
- For the spam/non-spam example earlier, x_i refers to an email in the training set.
- $y_i \in \{+1, -1\}$ could be used to indicate the label spam and non-spam.

Linear vs Non-linear classification

What is the difference between the two?

Linear vs Non-linear classification

What is the difference between the two? Assuming the input data x is 3-dimensional (i.e. in $\mathbb{R}^3$)

Linear classification

- Consider the classification function f_1 below, which is linear in both the input features and weights

$$f_1(x) = w_1 x_1 + w_2 x_2 + w_3 x_3$$

Linear vs Non-linear classification

What is the difference between the two? Assuming the input data x is 3-dimensional (i.e. in $\mathbb{R}^3$)

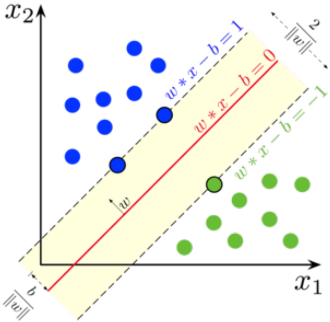
Linear classification

- Consider the classification function f_1 below, which is linear in both the input features and weights

$$f_1(x) = w_1 x_1 + w_2 x_2 + w_3 x_3$$

- In this case, the decision function $f_1(x)$ is trying to capture only **linear combination** of the input components x_1, x_2, x_3
- Linear feature map $\phi : \mathbb{R}^3 \mapsto \mathbb{R}^3$, and is given by, $\phi_1(x) = (x_1, x_2, x_3)^T$
- f_1 is parameterized as $f(.) = (w_1, w_2, w_3)^T$

Linear classification - Pictorial depiction



[2] In linear classification :

- classifier is a **straight line** in 2D (as shown above) **in the input space**
- generally called a **plane** or hyperplane in high dimensions

[2] picture from Wikiepedia

Linear vs Non-linear classification

Non-linear classification

- For the classification function f_2 below, which is linear in weights and **non-linear in input features**

$$f_2(x) = w_1x_1 + w_2x_2 + w_3x_3 + w_4x_1x_2 + w_5x_2x_3 + w_6x_1x_3 + w_7x_1x_2x_3$$

Linear vs Non-linear classification

Non-linear classification

- For the classification function f_2 below, which is linear in weights and **non-linear in input features**

$$f_2(x) = w_1x_1 + w_2x_2 + w_3x_3 + w_4x_1x_2 + w_5x_2x_3 + w_6x_1x_3 + w_7x_1x_2x_3$$

- Here, the decision function $f_2(x)$ is trying to capture **non-linear combination** of the input components as well such as $x_1x_2, x_1x_2x_3$

Linear vs Non-linear classification

Non-linear classification

- For the classification function f_2 below, which is linear in weights and **non-linear in input features**

$$f_2(x) = w_1x_1 + w_2x_2 + w_3x_3 + w_4x_1x_2 + w_5x_2x_3 + w_6x_1x_3 + w_7x_1x_2x_3$$

- Here, the decision function $f_2(x)$ is trying to capture **non-linear combination** of the input components as well such as $x_1x_2, x_1x_2x_3$
- Non-linear feature map $\phi_2 : \mathbb{R}^3 \mapsto \mathbb{R}^7$, and is given by $\phi_2(x) = (x_1, x_2, x_3, x_1x_2, x_2x_3, x_1x_3, x_1x_2x_3)^T$

Linear vs Non-linear classification

Non-linear classification

- For the classification function f_2 below, which is linear in weights and **non-linear in input features**

$$f_2(x) = w_1x_1 + w_2x_2 + w_3x_3 + w_4x_1x_2 + w_5x_2x_3 + w_6x_1x_3 + w_7x_1x_2x_3$$

- Here, the decision function $f_2(x)$ is trying to capture **non-linear combination** of the input components as well such as $x_1x_2, x_1x_2x_3$
- Non-linear feature map $\phi_2 : \mathbb{R}^3 \mapsto \mathbb{R}^7$, and is given by $\phi_2(x) = (x_1, x_2, x_3, x_1x_2, x_2x_3, x_1x_3, x_1x_2x_3)^T$
  - $\phi(x)_2 \in \mathcal{H}$, which is referred to as the feature space

Linear vs Non-linear classification

Non-linear classification

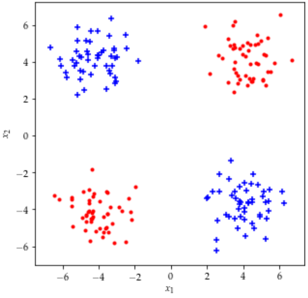
- For the classification function f_2 below, which is linear in weights and **non-linear in input features**

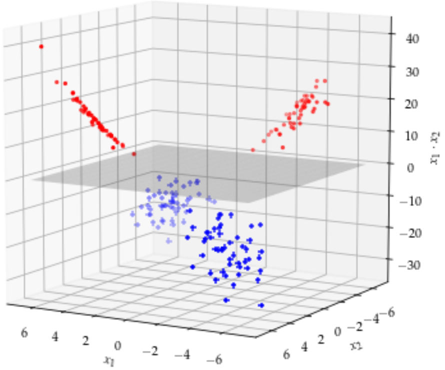
$$f_2(x) = w_1x_1 + w_2x_2 + w_3x_3 + w_4x_1x_2 + w_5x_2x_3 + w_6x_1x_3 + w_7x_1x_2x_3$$

- Here, the decision function $f_2(x)$ is trying to capture **non-linear combination** of the input components as well such as $x_1x_2, x_1x_2x_3$
- Non-linear feature map $\phi_2 : \mathbb{R}^3 \mapsto \mathbb{R}^7$, and is given by $\phi_2(x) = (x_1, x_2, x_3, x_1x_2, x_2x_3, x_1x_3, x_1x_2x_3)^T$
    - $\phi(x)_2 \in \mathcal{H}$, which is referred to as the feature space
- Note that the decision function $f_2(x)$ **is still linear in the weight vector co-efficients** w_j's and is parameterized by $(w_1, w_2, w_3, w_4, w_5, w_6, w_7)^T$

Non-linear Classification Examples

- Dataset in 2-D (left), which is not linearly separable can be separated by a plane in 3-D (third feature is the product $x_1 x_2$)





Kernel Methods - Motivation

- Data does not exhibit linear separability in the input space $\mathcal{X}$, and hence we need to apply feature transformation $\phi : \mathcal{X} \mapsto \mathcal{H}$

Kernel Methods - Motivation

- Data does not exhibit linear separability in the input space $\mathcal{X}$, and hence we need to apply feature transformation $\phi : \mathcal{X} \mapsto \mathcal{H}$
- Dimensionality of feature space $\mathcal{H} >>$ dimensionality of the input space $\mathcal{X}$

Kernel Methods - Motivation

- Data does not exhibit linear separability in the input space $\mathcal{X}$, and hence we need to apply feature transformation $\phi : \mathcal{X} \mapsto \mathcal{H}$
- Dimensionality of feature space $\mathcal{H} >>$ dimensionality of the input space $\mathcal{X}$
- As a result, we get data separability (statistical advantage) but at the cost of increased calculations (computational disadvantage)

Kernel Methods - Motivation

- Data does not exhibit linear separability in the input space $\mathcal{X}$, and hence we need to apply feature transformation $\phi : \mathcal{X} \mapsto \mathcal{H}$
- Dimensionality of feature space $\mathcal{H} >>$ dimensionality of the input space $\mathcal{X}$
- As a result, we get data separability (statistical advantage) but at the cost of increased calculations (computational disadvantage)
- Next - **Kernels to the rescue**!

Kernel Methods - Motivation

- **Kernels to the rescue**!
- Most learning algorithms such as Support Vector Machines, and Logistic regression (classification part used in deep networks) can be written in the form of Inner/dot product between vectors in the feature space $\phi(x_i)$s, i.e. $\langle \phi(x_i), \phi(x_j) \rangle$.
- The prediction function has the following form :

$$f(.) = \text{Some function of} \left(\sum_{i=1}^{n} \sum_{j=1}^{n} \langle \phi(x_i), \phi(x_j) \rangle \right)$$

- Kernels are functions which give us the dot product $\langle \phi(x_i), \phi(x_j) \rangle$ directly without explicitly computing the feature expansion $\phi(.)$

Dot Product - recall

- Dot product between feature vector of objects is a **measure of similarity between objects**
- In $\mathbb{R}^d$, dot product between two vectors x, y is given by $\langle x, y \rangle = ||x|| \times ||y|| \times cos(\theta)$, where θ is an angle between the vectors x and y.

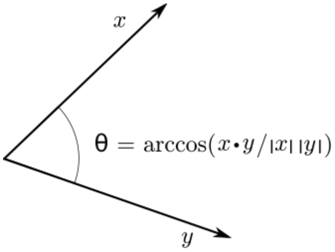


Figure: Figure from Wikipedia

Kernel Methods - Motivation

- Computing the dot product **explicitly** $\langle \phi(x_i), \phi(x_j) \rangle$ can be very computationally expensive
  - Interesting feature spaces such as those induced by the Gaussian Kernel) are infinite dimensional
  - In such a case, computing the inner product $\langle \phi(x_i), \phi(x_j) \rangle$ **explicitly** might not even be possible
- Kernel methods are a computational trick to compute the dot product **implicitly**
  - We do not have write down the explicit form of the feature maps $\phi(x_i)$, but rather compute the dot product directly using the kernel function $k(.,.)$
  - $k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$

Implcit vs Explicit Computation

- Example - Polynomial kernel. Assuming inputs $x, x' \in \mathbb{R}^d$, i.e. $x = (x_1, x_2, \ldots, x_d)$

$$k(x, x') = (\langle x, x' \rangle)^2 = \langle \phi(x), \phi(x') \rangle$$

- What is the dimensionality of the input space and feature space, $\phi(.)$ (keeping order into account. i.e., $x_1 x_2$ is different from $x_2 x_1$)?

Implcit vs Explicit Computation

- Example - Polynomial kernel. Assuming inputs $x, x' \in \mathbb{R}^d$, i.e. $x = (x_1, x_2, \ldots, x_d)$

$$k(x, x') = (\langle x, x' \rangle)^2 = \langle \phi(x), \phi(x') \rangle$$

- What is the dimensionality of the input space and feature space, $\phi(.)$ (keeping order into account. i.e., $x_1 x_2$ is different from $x_2 x_1$)?
  - $\phi : \mathbb{R}^d \mapsto \mathbb{R}^{d^2}$
  - $\phi(x) = \{x_1 x_1, x_1 x_2, \ldots x_1 x_d, \ldots, x_d x_1, x_d x_2, \ldots, x_d x_d\}$
- Computational complexity of
  - $\langle \phi(x), \phi(x') \rangle$?

Implcit vs Explicit Computation

- Example - Polynomial kernel. Assuming inputs $x, x' \in \mathbb{R}^d$, i.e. $x = (x_1, x_2, \ldots, x_d)$

$$k(x, x') = (\langle x, x' \rangle)^2 = \langle \phi(x), \phi(x') \rangle$$

- What is the dimensionality of the input space and feature space, $\phi(.)$ (keeping order into account. i.e., $x_1 x_2$ is different from $x_2 x_1$)?
  - $\phi : \mathbb{R}^d \mapsto \mathbb{R}^{d^2}$
  - $\phi(x) = \{x_1 x_1, x_1 x_2, \ldots x_1 x_d, \ldots, x_d x_1, x_d x_2, \ldots, x_d x_d\}$
- Computational complexity of
  - $\langle \phi(x), \phi(x') \rangle$? $O(d^2)$

Implcit vs Explicit Computation

- Example - Polynomial kernel. Assuming inputs $x, x' \in \mathbb{R}^d$, i.e. $x = (x_1, x_2, \ldots, x_d)$

$$k(x, x') = (\langle x, x' \rangle)^2 = \langle \phi(x), \phi(x') \rangle$$

- What is the dimensionality of the input space and feature space, $\phi(.)$ (keeping order into account. i.e., $x_1 x_2$ is different from $x_2 x_1$)?
  - $\phi : \mathbb{R}^d \mapsto \mathbb{R}^{d^2}$
  - $\phi(x) = \{x_1 x_1, x_1 x_2, \ldots x_1 x_d, \ldots, x_d x_1, x_d x_2, \ldots, x_d x_d\}$
- Computational complexity of
  - $\langle \phi(x), \phi(x') \rangle$? $O(d^2)$
  - $k(x, x')$?

Implcit vs Explicit Computation

- Example - Polynomial kernel. Assuming inputs $x, x' \in \mathbb{R}^d$, i.e. $x = (x_1, x_2, \ldots, x_d)$

$$k(x, x') = (\langle x, x' \rangle)^2 = \langle \phi(x), \phi(x') \rangle$$

- What is the dimensionality of the input space and feature space, $\phi(.)$ (keeping order into account. i.e., $x_1 x_2$ is different from $x_2 x_1$)?
    - $\phi : \mathbb{R}^d \mapsto \mathbb{R}^{d^2}$
    - $\phi(x) = \{x_1 x_1, x_1 x_2, \ldots x_1 x_d, \ldots, x_d x_1, x_d x_2, \ldots, x_d x_d\}$
- Computational complexity of
    - $\langle \phi(x), \phi(x') \rangle$? $O(d^2)$
    - $k(x, x')$? $O(d)$

Implcit vs Explicit Computation

- Example - Polynomial kernel. Assuming inputs $x, x' \in \mathbb{R}^d$, i.e. $x = (x_1, x_2, \ldots, x_d)$

$$k(x, x') = (\langle x, x' \rangle)^2 = \langle \phi(x), \phi(x') \rangle$$

- What is the dimensionality of the input space and feature space, $\phi(.)$ (keeping order into account. i.e., $x_1 x_2$ is different from $x_2 x_1$)?
  - $\phi : \mathbb{R}^d \mapsto \mathbb{R}^{d^2}$
  - $\phi(x) = \{x_1 x_1, x_1 x_2, \ldots x_1 x_d, \ldots, x_d x_1, x_d x_2, \ldots, x_d x_d\}$
- Computational complexity of
  - $\langle \phi(x), \phi(x') \rangle$? $O(d^2)$
  - $k(x, x')$? $O(d)$
- What if we consider another kernel with a higher degree such as $k(x, x') = (\langle x, x' \rangle)^{10}$

Gaussian kernel

Gaussian kernel - Closer points are more similar

- is given by

$$k(x, x') = \exp\left(-\frac{||x - x'||^2}{2\sigma^2}\right), \forall x, x' \in \mathbb{R}^d$$

  where $\sigma > 0$ is the kernel bandwidth
- What is the range of values for the Gaussian kernel $k(x, x')$?

Gaussian kernel

Gaussian kernel - Closer points are more similar

- is given by

$$k(x, x') = \exp\left(-\frac{||x - x'||^2}{2\sigma^2}\right), \forall x, x' \in \mathbb{R}^d$$

  where $\sigma > 0$ is the kernel bandwidth
- What is the range of values for the Gaussian kernel $k(x, x')$?
- For $\sigma = 1$, $k(x, x') = 1$ when $x = x'$ and $k(x, x') \approx 0$ when $||x - x'||^2 = 10$ (Since $\exp(-5) = 0.006$)
- Also, $k(x, x') = \langle \phi(x), \phi(x') \rangle$ for some feature space $\phi(.)$
  - As we will see in the next lecture, the feature space of Gaussian kernel is infinite dimensional

Towards defining a Kernel

How should a kernel be defined, which takes into account

- **Inner product** $\langle ., . \rangle$ for quantifying similarity
- The high (potentially infinite) dimensional **feature map** $\phi(.)$
- The high (potentially infinite) dimensional **feature space** $\mathcal{H}$

What is a Kernel - Definition

Definition

For a non-empty set $\mathcal{X}$, a function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is defined to be a kernel if there exists a Hilbert Space $\mathcal{H}$ and a function $\phi : \mathcal{X} \to \mathcal{H}$ such that $\forall x, x' \in \mathcal{X}$, $k(x, x') := \langle \phi(x), \phi(x') \rangle_{\mathcal{H}}$.

Vector Spaces (a slight detour)

Vector Space

Definition (Vector Space)

A vector space is non-empty set V, that is equipped or associated with two operations, (i) *addition* - For each pair of elements $v, w \in V$, there is an element $v + w \in V$, and (ii) *scalar multiplication* - For each element $v \in V$ and $\alpha \in \mathbb{R}$, there is an element $\alpha v \in V$.

If the above two operations of *addition* and *scalar multiplication* satisfy a set of (following) requirements [3], then V is called a vector space.

- For all $v, w \in V, v + w = w + v$
- A few other similar conditions ...

[3] not so important for the course

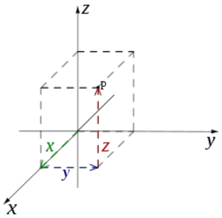


Figure: Figure from Wikipedia

1. $\mathbb{R}^d$ is a vector space
2. Space of functions can also be vector space. For example - The set W of polynomials of degree atmost 3

$$P(x) = a_3x^3 + a_2x^2 + a_1x + a_0, \text{ such that } x, a_i \in \mathbb{R}$$
$$P(x) = b_3x^3 + b_2x^2 + b_1x + b_0, \text{ such that } x, b_i \in \mathbb{R}$$

- $P(x) + Q(x) \in W$,
- For $\alpha P(x) \in W$, for $\alpha \in \mathbb{R}$

Normed Vector Spaces

Definition (Norm)

Let V be a real vector space. A norm on V is a function (denoted as $||.||$)

$$||.|| : V \to \mathbb{R}^+$$

that satisfies the following requirements :

- $||v + w|| \leq ||v|| + ||w||, \forall v, w \in V$ (Triangle Inequality)
- $||\alpha v|| = |\alpha| \times ||v||, \forall v \in V$, and $\alpha \in \mathbb{R}$
- $||v|| \geq 0, \forall v \in V$, and $||v|| = 0$ if and only if $v = \mathbf{0}$ (Non-negativity)

A vector space equipped with a norm is called a normed vector space.

Examples of Normed Vector Spaces

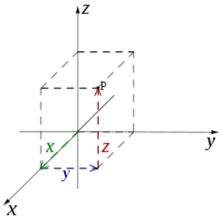


Figure: Figure from Wikipedia

Example1 - $\mathbb{R}^d$ (For $d = 3$, shown above) is a normed vector space with the ℓ_2 norm of an element $v \in \mathbb{R}^d$ given by $||v|| := \sqrt{\sum_{i=1}^{d} v_i^2}$

Examples of Normed Vector Spaces

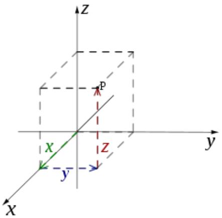


Figure: Figure from Wikipedia

Example1 - $\mathbb{R}^d$ (For $d = 3$, shown above) is a normed vector space with the ℓ_2 norm of an element $v \in \mathbb{R}^d$ given by $||v|| := \sqrt{\sum_{i=1}^{d} v_i^2}$

Example2 - $C[a, b] := \{f : [a, b] \to \mathbb{R} | f$ is a continuous function$\}$ (set of continuous functions over a bounded interval $[a, b]$) with the norm defined as

$$||f||_\infty := \max_{\{x \in [a,b]\}} |f(x)|$$

is a normed vector space

Inner Product Spaces

Definition (Inner Product)

Let V be a real vector space. An inner product on V is a function

$$\langle ., . \rangle : V \times V \rightarrow \mathbb{R}$$

that satisfies the following requirements :

- $\langle \alpha v + \beta w, u \rangle = \alpha \langle v, u \rangle + \beta \langle w, u \rangle \forall u, v, w \in V$, and $\alpha, \beta \in \mathbb{R}$ (Linearity)
- $\langle v, w \rangle = \langle w, v \rangle \forall v, w \in V$ (Symmetry)
- $\langle v, v \rangle \geq 0, \forall v \in V$, and $\langle v, v \rangle = 0$ if and only if $v = 0$

A vector space equipped with an inner product is called an inner product space.

Examples of Inner Product Spaces

Examples

- For $\mathbb{R}^d$ the inner product is given by $\langle v, w \rangle = \sum_{i=1}^{d} v_i w_i$, also called the dot-product
- For function spaces, the inner product between real valued functions over a closed interval $[a, b]$ is given by

$$\langle f, g \rangle = \int_a^b f(x) g(x) dx$$

Inner Product Spaces as Normed vector spaces

Inner proudct as a norm

Let V be a real vector space with an inner product $\langle ., . \rangle$. Then

$$||v|| := \sqrt{\langle v, v \rangle}, v \in V$$

defines a norm on V.

Proof.

Need to prove that for $v \in V$, $\sqrt{\langle v, v \rangle}$ is a norm on V, i.e. it satisfies the definition required for a function to be norm on a vector space. $\square$

Hilbert Spaces

A Hilbert Space refers to an Inner product space with some technical condition(not so important for our course) [4]

[4] i.e. it contains the limits of all Cauchy sequences

Pictorial depiction of Vector Spaces

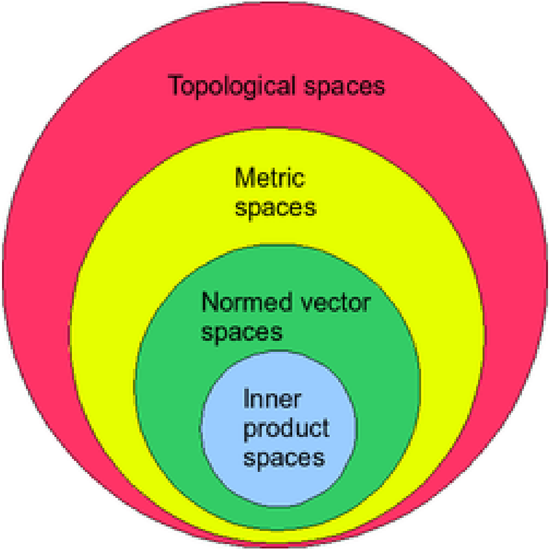


Figure: Figure from Wikipedia

Back to Kernels

What is a Kernel - Definition

Definition

For a non-empty set $\mathcal{X}$, a function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is defined to be a kernel if there exists a Hilbert Space $\mathcal{H}$ and a function $\phi : \mathcal{X} \to \mathcal{H}$ such that $\forall x, x' \in \mathcal{X}$, $k(x, x') := \langle \phi(x), \phi(x') \rangle_{\mathcal{H}}$.

- $\phi(.)$ is called the feature map, and $\mathcal{H}$ is called the feature space
- $\mathcal{X}$ is only required to be a non-empty set
- No structure (such as a vector space) is required over $\mathcal{X}$, it can just be raw documents or pictures

Example of Kernel functions

- Polynomial kernel. Assuming inputs $x, x' \in \mathbb{R}^d$, i.e. $x = (x_1, x_2, \ldots, x_d)$

  $$k(x, x') = (\langle x, x' \rangle + c)^m$$

  for $c \geq 0, m \in \mathbb{N}$.
  - For $m = 1, c = 0, k(x, x') = \langle x, x' \rangle$ is called linear kernel

Example of Kernel functions

- Polynomial kernel. Assuming inputs $x, x' \in \mathbb{R}^d$, i.e. $x = (x_1, x_2, \ldots, x_d)$

$$k(x, x') = (\langle x, x' \rangle + c)^m$$

for $c \geq 0, m \in \mathbb{N}$.

  - For $m = 1, c = 0, k(x, x') = \langle x, x' \rangle$ is called linear kernel

- Gaussian kernel

$$k(x, x') = \exp\left(-\frac{||x - x'||^2}{2\sigma^2}\right), \forall x, x' \in \mathbb{R}^d$$

where $\sigma > 0$ is the kernel bandwidth

Non-uniqueness of Feature Map and Hilbert Space

For a given Kernel, the feature map $\phi(.)$ and the Hilbert space $\mathcal{H}$ is non-unique.

- For the linear kernel $k(x, x') := \langle x, x' \rangle$, two of the many possible choices of feature maps and Hilbert space are :
  - $\phi_1(x) = x$, and $\mathcal{H}_1 = \mathbb{R}$

Non-uniqueness of Feature Map and Hilbert Space

For a given Kernel, the feature map $\phi(.)$ and the Hilbert space $\mathcal{H}$ is non-unique.

- For the linear kernel $k(x, x') := \langle x, x' \rangle$, two of the many possible choices of feature maps and Hilbert space are :
  - $\phi_1(x) = x$, and $\mathcal{H}_1 = \mathbb{R}$
  - $\phi_2(x) = \frac{1}{\sqrt{2}}(x, x)$, and $\mathcal{H}_2 = \mathbb{R}^2$

Positive Scalar Multiple of a kernel

Positive scalar multiple

For any $\alpha > 0$, if $k(.,.)$ is a kernel, then $\alpha k(.,.)$ is also a kernel.

Positive Scalar Multiple of a kernel

Positive scalar multiple

For any $\alpha > 0$, if $k(.,.)$ is a kernel, then $\alpha k(.,.)$ is also a kernel.

What do we need to do to prove it is a kernel

Positive Scalar Multiple of a kernel

Positive scalar multiple

For any $\alpha > 0$, if $k(.,.)$ is a kernel, then $\alpha k(.,.)$ is also a kernel.

What do we need to do to prove it is a kernel

- Find a feature map $\phi(.)$
- Find a feature space $\mathcal{H}$

such that $k(x, x') = \langle \phi(x), \phi(x') \rangle$

Positive Scalar Multiple of a kernel

Positive scalar multiple

For any $\alpha > 0$, if $k(.,.)$ is a kernel, then $\alpha k(.,.)$ is also a kernel.

What do we need to do to prove it is a kernel

- Find a feature map $\phi(.)$
- Find a feature space $\mathcal{H}$

such that $k(x, x') = \langle \phi(x), \phi(x') \rangle$

Proof.

$$\alpha k(x, x')$$

Positive Scalar Multiple of a kernel

Positive scalar multiple

For any $\alpha > 0$, if $k(.,.)$ is a kernel, then $\alpha k(.,.)$ is also a kernel.

What do we need to do to prove it is a kernel

- Find a feature map $\phi(.)$
- Find a feature space $\mathcal{H}$

such that $k(x, x') = \langle \phi(x), \phi(x') \rangle$

Proof.

$$\alpha k(x, x') = \alpha \langle \phi(x), \phi(x') \rangle_{\mathcal{H}}$$

Positive Scalar Multiple of a kernel

Positive scalar multiple

For any $\alpha > 0$, if $k(.,.)$ is a kernel, then $\alpha k(.,.)$ is also a kernel.

What do we need to do to prove it is a kernel

- Find a feature map $\phi(.)$
- Find a feature space $\mathcal{H}$

such that $k(x, x') = \langle \phi(x), \phi(x') \rangle$

Proof.

$$\alpha k(x, x') = \alpha \langle \phi(x), \phi(x') \rangle_{\mathcal{H}} = \langle \sqrt{\alpha}\phi(x), \sqrt{\alpha}\phi(x') \rangle_{\mathcal{H}}$$

□

Conic Sum of Kernels

Conic Sum of Kernels

For kernels $(k_j)_{j=1}^K$, and $(\alpha_j)_{j=1}^K > 0, \sum_{j=1}^K \alpha_j k_j$ is also a kernel

Proof.

$\sum_{j=1}^K \alpha_j k_j(x, x')$

Conic Sum of Kernels

Conic Sum of Kernels

For kernels $(k_j)_{j=1}^{K}$, and $(\alpha_j)_{j=1}^{K} > 0, \sum_{j=1}^{K} \alpha_j k_j$ is also a kernel

Proof.

$\sum_{j=1}^{K} \alpha_j k_j(x, x') = \sum_{j=1}^{K} \alpha_j \langle \phi_j(x), \phi_j(x') \rangle_{\mathcal{H}_j}$

Conic Sum of Kernels

Conic Sum of Kernels

For kernels $(k_j)_{j=1}^K$, and $(\alpha_j)_{j=1}^K > 0, \sum_{j=1}^K \alpha_j k_j$ is also a kernel

Proof.

$\sum_{j=1}^K \alpha_j k_j(x, x') = \sum_{j=1}^K \alpha_j \langle \phi_j(x), \phi_j(x') \rangle_{\mathcal{H}_j} =$
$\sum_{j=1}^K \langle \sqrt{\alpha_j} \phi_j(x), \sqrt{\alpha_j} \phi_j(x') \rangle_{\mathcal{H}_j}$

Conic Sum of Kernels

Conic Sum of Kernels

For kernels $(k_j)_{j=1}^{K}$, and $(\alpha_j)_{j=1}^{K} > 0, \sum_{j=1}^{K} \alpha_j k_j$ is also a kernel

Proof.

$\sum_{j=1}^{K} \alpha_j k_j(x, x') = \sum_{j=1}^{K} \alpha_j \langle \phi_j(x), \phi_j(x') \rangle_{\mathcal{H}_j} =$
$\sum_{j=1}^{K} \langle \sqrt{\alpha_j}\phi_j(x), \sqrt{\alpha_j}\phi_j(x') \rangle_{\mathcal{H}_j} = \langle \hat{\phi}_j(x), \hat{\phi}_j(x') \rangle_{\hat{\mathcal{H}}}$ $\square$

where $\hat{\phi}_j(x) = (\sqrt{\alpha_1}\phi_1(x), \ldots, \sqrt{\alpha_K}\phi_K(x))$, and $\hat{\mathcal{H}} = \mathcal{H}_1 \oplus \ldots \oplus \mathcal{H}_K$. Here $\oplus$ denotes axes concatenation of vector spaces. For example - $\mathbb{R}^2 \oplus \mathbb{R}^1 = \mathbb{R}^3$. i.e. adding a third dimension to a plane leads to a 3D space.

Difference of Kernels

Difference of Kernels is not necessarily a kernel

- Consider two kernels k_1 and k_2. Let there be an $x \in \mathcal{X}$ such that $k_1(x, x) - k_2(x, x) < 0$. Otherwise consider $k_2(x, x) - k_1(x, x)$, the same argument holds.

Difference of Kernels

Difference of Kernels is not necessarily a kernel

- Consider two kernels k_1 and k_2. Let there be an $x \in \mathcal{X}$ such that $k_1(x, x) - k_2(x, x) < 0$. Otherwise consider $k_2(x, x) - k_1(x, x)$, the same argument holds.
- If $k_1 - k_2$ is a kernel, then $\exists \phi$ and $\mathcal{H}$ such that $\forall x, x' \in \mathcal{X}$,

$$k_1(x, x') - k_2(x, x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{H}}$$

Difference of Kernels

Difference of Kernels is not necessarily a kernel

- Consider two kernels k_1 and k_2. Let there be an $x \in \mathcal{X}$ such that $k_1(x, x) - k_2(x, x) < 0$. Otherwise consider $k_2(x, x) - k_1(x, x)$, the same argument holds.
- If $k_1 - k_2$ is a kernel, then $\exists \phi$ and $\mathcal{H}$ such that $\forall x, x' \in \mathcal{X}$,

$$k_1(x, x') - k_2(x, x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{H}}$$

- How about $x = x'$?

Properties

Mappings

For an arbitrary function $f : \mathcal{X} \mapsto \mathbb{R}$, and a kernel $k(.,.)$

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

is also a kernel

Properties

Mappings

For an arbitrary function $f : \mathcal{X} \mapsto \mathbb{R}$, and a kernel $k(.,.)$

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

is also a kernel

Proof.

$\hat{k}(x, x') =$

Properties

Mappings

For an arbitrary function $f : \mathcal{X} \mapsto \mathbb{R}$, and a kernel $k(.,.)$

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

is also a kernel

Proof.

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

Properties

Mappings

For an arbitrary function $f : \mathcal{X} \mapsto \mathbb{R}$, and a kernel $k(.,.)$

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

is also a kernel

Proof.

$\hat{k}(x, x') = f(x)k(x, x')f(x') = f(x)\langle\phi(x), \phi(x')\rangle_{\mathcal{H}} f(x') =$

Properties

Mappings

For an arbitrary function $f : \mathcal{X} \mapsto \mathbb{R}$, and a kernel $k(.,.)$

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

is also a kernel

Proof.

$\hat{k}(x, x') = f(x)k(x, x')f(x') = f(x)\langle\phi(x), \phi(x')\rangle_{\mathcal{H}} f(x') =$
$\langle f(x)\phi(x), \phi(x')f(x')\rangle_{\mathcal{H}}$

Properties

Mappings

For an arbitrary function $f : \mathcal{X} \mapsto \mathbb{R}$, and a kernel $k(.,.)$

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

is also a kernel

Proof.

$\hat{k}(x, x') = f(x)k(x, x')f(x') = f(x)\langle\phi(x), \phi(x')\rangle_{\mathcal{H}} f(x') =$
$\langle f(x)\phi(x), \phi(x')f(x')\rangle_{\mathcal{H}}$, a different kernel $\hat{k}(.,.)$ but the same Hilbert Space $\mathcal{H}$. $\square$

Properties

Mappings

For an arbitrary function $f : \mathcal{X} \mapsto \mathbb{R}$, and a kernel $k(.,.)$

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

is also a kernel

Proof.

$\hat{k}(x, x') = f(x)k(x, x')f(x') = f(x)\langle\phi(x), \phi(x')\rangle_{\mathcal{H}} f(x') =$
$\langle f(x)\phi(x), \phi(x')f(x')\rangle_{\mathcal{H}}$, a different kernel $\hat{k}(.,.)$ but the same Hilbert Space $\mathcal{H}$. $\square$

Cauchy-Schwarz Inequality $|k(x, x')| \leq \sqrt{k(x, x)}\sqrt{k(x, x')}$

Properties

Mappings

For an arbitrary function $f : \mathcal{X} \mapsto \mathbb{R}$, and a kernel $k(.,.)$

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

is also a kernel

Proof.

$\hat{k}(x, x') = f(x)k(x, x')f(x') = f(x)\langle\phi(x), \phi(x')\rangle_{\mathcal{H}} f(x') =$
$\langle f(x)\phi(x), \phi(x')f(x')\rangle_{\mathcal{H}}$, a different kernel $\hat{k}(.,.)$ but the same Hilbert Space $\mathcal{H}$. □

Cauchy-Schwarz Inequality $|k(x, x')| \leq \sqrt{k(x, x)}\sqrt{k(x, x')}$

Proof.

$|k(x, x')|$

Properties

Mappings

For an arbitrary function $f : \mathcal{X} \mapsto \mathbb{R}$, and a kernel $k(.,.)$

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

is also a kernel

Proof.

$\hat{k}(x, x') = f(x)k(x, x')f(x') = f(x)\langle\phi(x), \phi(x')\rangle_{\mathcal{H}} f(x') =$
$\langle f(x)\phi(x), \phi(x')f(x')\rangle_{\mathcal{H}}$, a different kernel $\hat{k}(.,.)$ but the same Hilbert Space $\mathcal{H}$. □

Cauchy-Schwarz Inequality $|k(x, x')| \leq \sqrt{k(x, x)}\sqrt{k(x, x')}$

Proof.

$|k(x, x')| = |\langle\phi(x), \phi(x')\rangle_{\mathcal{H}}|$

Properties

Mappings

For an arbitrary function $f : \mathcal{X} \mapsto \mathbb{R}$, and a kernel $k(.,.)$

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

is also a kernel

Proof.

$\hat{k}(x, x') = f(x)k(x, x')f(x') = f(x)\langle\phi(x), \phi(x')\rangle_{\mathcal{H}} f(x') = \langle f(x)\phi(x), \phi(x')f(x')\rangle_{\mathcal{H}}$, a different kernel $\hat{k}(.,.)$ but the same Hilbert Space $\mathcal{H}$. $\square$

Cauchy-Schwarz Inequality $|k(x, x')| \leq \sqrt{k(x, x)}\sqrt{k(x, x')}$

Proof.

$|k(x, x')| = |\langle\phi(x), \phi(x')\rangle_{\mathcal{H}}| \leq ||\phi(x)||_{\mathcal{H}} ||\phi(x')||_{\mathcal{H}}$ $\square$

Product of Kernels

Product of Kernels

For kernels k_1 on $\mathcal{X}_1$, and k_2 on $\mathcal{X}_2$, $k_1 \times k_2$ is a kernel on $\mathcal{X}_1 \times \mathcal{X}_2$.

Polynomial Kernels

Polynomial kernel

Let $x, x' \in \mathbb{R}^d$ for $d \geq 1$, and $m \geq 1$ be an integer, and $c \geq 0$ is a positive real number, then

$$k(x, x') := (\langle x, x' \rangle + c)^m$$

is a kernel

Proof.

Homework exercise
Hint : Use Binomial Theorem $\square$

Polynomial Kernels

Polynomial kernel

Let $x, x' \in \mathbb{R}^d$ for $d \geq 1$, and $m \geq 1$ be an integer, and $c \geq 0$ is a positive real number, then

$$k(x, x') := (\langle x, x' \rangle + c)^m$$

is a kernel

Proof.

Homework exercise

Hint : Use Binomial Theorem $\square$

Linear Kernel

For $m = 1$, and $c = 0$, $k(x, x') := \langle x, x' \rangle$ is called linear kernel

Infinite dimensional feature spaces

Definition

The space ℓ_2 of square summable sequences referring to sequences $a := (a_i)_{i=1}$ for which

$$||a||_{\ell_2}^2 = \sum_{i=1}^{\infty} a_i^2 < \infty$$

Inner product between infinite dimensional sequences

For a sequence of functions $(\phi_i(x))_{i \geq 1}$ in ℓ_2 where $\phi_i : \mathcal{X} \mapsto \mathbb{R}$ is the i-th co-ordinate of $\phi(x)$. Then

$$k(x, x') := \sum_{i=1}^{\infty} \phi_i(x)\phi_i(x')$$

is also a kernel

Why we need Square summability?

By Cauchy-Schwarz inequality

$$|\sum_{i=1}^{\infty} \phi_i(x)\phi_i(x')| \leq ||\phi_i(x)||_{\ell_2}||\phi_i(x')||_{\ell_2}$$

Both the terms in the product on RHS are square summable individually, and hence is the product

- Therefore, the inner product is well-defined (i.e., not infinite) $\forall x, x' \in \mathcal{X}$.

Infinite Dimensional feature space - Example

Exponential Kernel

Exponential Kernel

$$k(x, x') = \exp(\langle x, x' \rangle), x, x' \in \mathbb{R}^d$$

is a kernel

Proof.

$$\exp(\langle x, x' \rangle) =$$

Infinite Dimensional feature space - Example

Exponential Kernel

Exponential Kernel

$$k(x, x') = \exp(\langle x, x' \rangle), x, x' \in \mathbb{R}^d$$

is a kernel

Proof.

$$\exp(\langle x, x' \rangle) = \sum_{i=0}^{\infty} \frac{\langle x, x' \rangle^i}{i!} \text{ (By Taylor Series expansion of } \exp z\text{)}$$

Infinite Dimensional feature space - Example

Exponential Kernel

Exponential Kernel

$$k(x, x') = \exp(\langle x, x' \rangle), x, x' \in \mathbb{R}^d$$

is a kernel

Proof.

$$\exp(\langle x, x' \rangle) = \sum_{i=0}^{\infty} \frac{\langle x, x' \rangle^i}{i!} \text{ (By Taylor Series expansion of } \exp z)$$

Since $\langle x, x' \rangle$ is a kernel, RHS is a kernel by sum and product rule □

Gaussian Kernel

For inputs $x, x' \in \mathbb{R}^d$, **Gaussian Kernel** (given below)

$$k(x, x') = \exp\left(-\frac{||x - x'||^2}{\sigma^2}\right) \text{ for } x, x' \in \mathbb{R}^d$$

where σ is fixed, **is a kernel**

Proof.

$$\exp\left(-\frac{||x - x'||^2}{\sigma^2}\right)$$

Gaussian Kernel

For inputs $x, x' \in \mathbb{R}^d$, **Gaussian Kernel** (given below)

$$k(x, x') = \exp\left(-\frac{||x - x'||^2}{\sigma^2}\right) \text{ for } x, x' \in \mathbb{R}^d$$

where σ is fixed, **is a kernel**

Proof.

$$\exp\left(-\frac{||x - x'||^2}{\sigma^2}\right) = \exp\left(-\frac{||x||^2 + ||x'||^2 - 2\langle x, x'\rangle}{\sigma^2}\right)$$

Gaussian Kernel

For inputs $x, x' \in \mathbb{R}^d$, **Gaussian Kernel** (given below)

$$k(x, x') = \exp\left(-\frac{||x - x'||^2}{\sigma^2}\right) \text{ for } x, x' \in \mathbb{R}^d$$

where σ is fixed, **is a kernel**

Proof.

$$\begin{aligned}
\exp\left(-\frac{||x - x'||^2}{\sigma^2}\right) &= \exp\left(-\frac{||x||^2 + ||x'||^2 - 2\langle x, x'\rangle}{\sigma^2}\right) \\
&= \exp\left(-\frac{||x||^2}{\sigma^2}\right) \exp\left(-\frac{2\langle x, x'\rangle}{\sigma^2}\right) \exp\left(-\frac{||x'||^2}{\sigma^2}\right)
\end{aligned}$$

Gaussian Kernel

For inputs $x, x' \in \mathbb{R}^d$, **Gaussian Kernel** (given below)

$$k(x, x') = \exp\left(-\frac{||x - x'||^2}{\sigma^2}\right) \text{ for } x, x' \in \mathbb{R}^d$$

where σ is fixed, **is a kernel**

Proof.

$$\begin{aligned}
\exp\left(-\frac{||x - x'||^2}{\sigma^2}\right) &= \exp\left(-\frac{||x||^2 + ||x'||^2 - 2\langle x, x'\rangle}{\sigma^2}\right) \\
&= \exp\left(-\frac{||x||^2}{\sigma^2}\right)\exp\left(-\frac{2\langle x, x'\rangle}{\sigma^2}\right)\exp\left(-\frac{||x'||^2}{\sigma^2}\right) \\
&= f(x)\exp\left(-\frac{2\langle x, x'\rangle}{\sigma^2}\right)f(x')
\end{aligned}$$

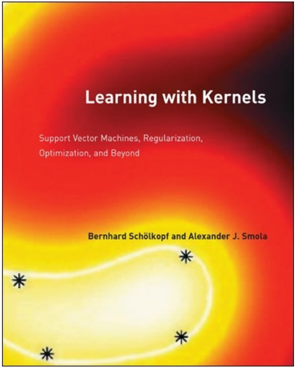
$\square$

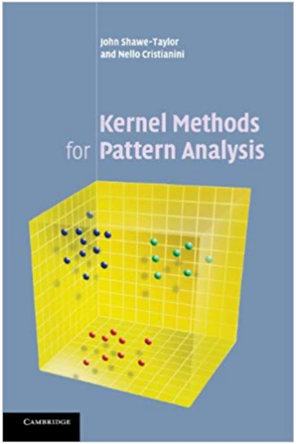
The lecture closely follows

- Slides from the lecture on kernel methods by Arthur Gretton, Machine Learning Summer School
  - Link - http://mlss.tuebingen.mpg.de/2015/slides/gretton/part_1.pdf
- More detailed notes
  - http://www.gatsby.ucl.ac.uk/~gretton/coursefiles/lecture4_introToRKHS.pdf

Books for further study

- Learning with kernels - Schoelkopf and Smola
- Kernel Methods for Pattern Analysis - Shawe-Taylor and Christianini





Summary

- Linear vs Non-linear classification
- Vector, Inner product, and Hilbert Spaces
- Kernels
  - Definition and Feature Mapping
  - Finite and infinite-dimensional feature spaces
- Kernel Properties
  - Conic combination of kernels
  - Product of kernels
- Positive-definiteness of kernel functions
- For the next lecture - recall Fourier series expansions