# 2. Kernels and Regularization

## GI01/M055: Supervised Learning

Mark Herbster and Massimiliano Pontil

10 October 2016

University College London
Department of Computer Science

# Today's Plan

## Feature Maps

- Ridge Regression
- Basis Functions (Explicit Feature Maps)
- Kernel Functions (Implicit Feature Maps)

## bibliography

Chapters 2 and 3 of *Kernel Methods for Pattern Analysis*, Shawe-Taylor.J, and Cristianini N., Cambridge University Press (2004)

Part I

# Feature Maps

# Overview

- We show how a linear method such as least squares may be **lifted** to a (potentially) higher dimensional space to provide a nonlinear regression.
- We consider both **explicit** and **implicit** feature maps
- A feature map is simply a function that maps the “inputs” into a new space.
- Thus the original method is now nonlinear in original “inputs” but linear in the “mapped inputs”
- Explicit feature maps are often known as the *Method of Basis Functions*
- Implicit feature maps are often known as the *(reproducing) “Kernel Trick”*

# Linear interpolation

**Problem**

We wish to find a function $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}$ which best interpolates a data set $S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\} \subseteq \mathbb{R}^n \times \mathbb{R}$

- If the data have been generated in the form $(\mathbf{x}, f(\mathbf{x}))$, the vectors $\mathbf{x}_i$ are linearly independent and $m = n$ then there is a unique interpolant whose parameter $\mathbf{w}$ solves

$$X\mathbf{w} = \mathbf{y}$$

  where, recall, $\mathbf{y} = (y_1, \ldots, y_m)^\top$ and $X = [\mathbf{x}_1, \ldots, \mathbf{x}_m]^\top$
- Otherwise, this problem is *ill-posed*

# Ill-posed problems

A problem is well-posed – in the sense of Hadamard (1902) – if

(1) a solution exists

(2) the solution is unique

(3) the solution depends continuously on the data

A problem is ill-posed if it is not well-posed

Learning problems are in general ill-posed (usually because of (2))

Regularization theory provides a general framework to solve ill-posed problems

# Ridge Regression

# Ridge Regression

Motivation:

1. Give a set of $k$ hypothesis classes $\{\mathcal{H}_r\}_{r \in \mathbb{N}_k}$ we can choose an appropriate hypothesis class with *cross-validation*
2. An alternative compatible with linear regression is to choose a single “complex” hypothesis class and then modify the error function by adding a “complexity” term which penaltizes complex functions
3. This is known as **regularization**
4. Cross-validation may still be needed to set the regularization parameter (see below) and other parameters defining the complexity term

# Ridge Regression

We minimize the regularized (penalized) empirical error

$$\mathcal{E}_{\mathrm{emp}_\lambda}(\mathbf{w}) := \sum_{i=1}^{m}(y_i - \mathbf{w}^\top \mathbf{x}_i)^2 + \lambda \sum_{\ell=1}^{n} w_\ell^2 \equiv (\mathbf{y} - X\mathbf{w})^\top(\mathbf{y} - X\mathbf{w}) + \lambda \mathbf{w}^\top \mathbf{w}$$

The positive parameter $\lambda$ defines a trade-off between the error on the data and the norm of the vector $\mathbf{w}$ (degree of regularization)

Setting $\nabla\mathcal{E}_{\mathrm{emp}_\lambda}(\mathbf{w}) = 0$, we obtain the modified normal equations

$$-2X^\top(\mathbf{y} - X\mathbf{w}) + 2\lambda\mathbf{w} = 0 \tag{1}$$

whose solution (called *regularized solution*) is

$$\mathbf{w} = (X^\top X + \lambda I_n)^{-1} X^\top \mathbf{y} \tag{2}$$

It can be shown that the regularized solution can be written as

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i \mathbf{x}_i \quad \Rightarrow \quad f(\mathbf{x}) = \sum_{i=1}^{m} \alpha_i \mathbf{x}_i^\top \mathbf{x} \qquad (*)$$

where the vector of parameters $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_m)^\top$ is given by

$$\boldsymbol{\alpha} = (XX^\top + \lambda I_m)^{-1}\mathbf{y} \tag{3}$$

• **Function representations:** we call the functional form (or representation) $f(\mathbf{x}) = \mathbf{w}^\top x$ the *primal form* and (*) the *dual form* (or representation)

The dual form is computationally convenient when $n > m$

# Dual representation (continued – 1)

We rewrite eq.(1) as

$$\mathbf{w} = \frac{X^\top(\mathbf{y} - X\mathbf{w})}{\lambda}$$

Thus we have

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i \mathbf{x}_i \tag{4}$$

with

$$\alpha_i = \frac{y_i - \mathbf{w}^\top \mathbf{x}_i}{\lambda} \tag{5}$$

Consequently, we have that

$$\mathbf{w}^\top \mathbf{x} = \sum_{i=1}^{m} \alpha_i \mathbf{x}_i^\top \mathbf{x}$$

proving eq.(*).

# Dual representation (continued – 2)

Plugging eq.(4) in eq.(5) we obtain

$$\alpha_i = \frac{y_i - (\sum_{j=1}^{m} \alpha_j \mathbf{x}_j)^\top \mathbf{x}_i}{\lambda}$$

Thus (with defining $\delta_{ij} = 1$ if $i = j$ and as 0 otherwise)

$$y_i = (\sum_{j=1}^{m} \alpha_j \mathbf{x}_j)^\top \mathbf{x}_i + \lambda \alpha_i$$

$$y_i = \sum_{j=1}^{m} (\alpha_j {\mathbf{x}_j}^\top \mathbf{x}_i + \alpha_j \lambda \delta_{ij})$$

$$y_i = \sum_{j=1}^{m} ({\mathbf{x}_j}^\top \mathbf{x}_i + \lambda \delta_{ij}) \alpha_j$$

Hence $(XX^\top + \lambda \mathbf{I}_m)\boldsymbol{\alpha} = \mathbf{y}$ from which eq.(3) follows.

# Computational Considerations

Training time:

- Solving for $\mathbf{w}$ in the primal form requires $O(mn^2 + n^3)$ operations while solving for $\boldsymbol{\alpha}$ in the dual form requires $O(nm^2 + m^3)$ (see $(*)$) operations

If $m \ll n$ it is more efficient to use the dual representation Running

(testing) time:

- Computing $f(\mathbf{x})$ on a test vector $\mathbf{x}$ in the primal form requires $O(n)$ operations while the dual form (see $(*)$) requires $O(mn)$ operations

# Sparse representation

We can benefit even further in the dual representation if the inputs are sparse!

## Example

Suppose each input $\mathbf{x} \in \mathbb{R}^n$ has most of its components equal to zero (e.g., consider images where most pixels are 'black' or text documents represented as 'bag of words')

- If $k$ denotes the number of nonzero components of the input then computing $\mathbf{x}^\top \mathbf{t}$ requires at most $O(k)$ operations.
  How do we do this?
- If $km \ll n$ (which implies $m, k \ll n$) the dual representation requires $O(km^2 + m^3)$ computations for training and $O(mk)$ for testing

# Basis Functions

# Basis Functions – Explicit Feature Map

The above ideas can naturally be generalized to nonlinear function regression

By a *feature map* we mean a function $\boldsymbol{\phi} : \mathbb{R}^n \to \mathbb{R}^N$

$$\boldsymbol{\phi}(\mathsf{x}) = (\phi_1(\mathsf{x}), \ldots, \phi_N(\mathsf{x}))^\top, \quad \mathsf{x} \in \mathbb{R}^n$$

- The $\phi_1, \ldots, \phi_N$ are called called *basis* functions
- Vector $\boldsymbol{\phi}(\mathsf{x})$ is called the *feature vector* and the space

  $$\{\boldsymbol{\phi}(\mathsf{x}) : \mathsf{x} \in \mathbb{R}^n\}$$

  the *feature space*

The non-linear regression function has the primal representation

$$f(\mathsf{x}) = \sum_{j=1}^{N} w_j \phi_j(\mathsf{x})$$

# Kernels

# Computational Considerations Revisited

Again, if $m \ll N$ it is more efficient to work with the dual representation

Key observation: in the dual representation we don't need to know $\phi$ explicitly; we just need to know the inner product between any pair of feature vectors!

Example: $N = n^2$, $\phi(\mathrm{x}) = (x_i x_j)_{i,j=1}^{n}$. In this case we have $\langle \phi(\mathrm{x}), \phi(\mathrm{t}) \rangle = (\mathrm{x}^\top \mathrm{t})^2$ which requires only $O(n)$ computations whereas $\phi(\mathrm{x})$ requires $O(n^2)$ computations

# Kernel Functions – Implicit Feature Map

Given a feature map $\phi$ we define its associated kernel function $K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ as

$$K(\mathbf{x}, \mathbf{t}) = \langle \phi(\mathbf{x}), \phi(\mathbf{t}) \rangle, \quad \mathbf{x}, \mathbf{t} \in \mathbb{R}^n$$

- Key Point: for some feature map $\phi$ computing $K(\mathbf{x}, \mathbf{t})$ is independent of $N$ (only dependent on $n$). Where *necessarily* $\phi(\mathbf{x})$ depends on $N$.

**Example (cont.)** If $\phi(\mathbf{x}) = (x_{i_1} x_{i_2} \cdots x_{i_r} : i_1, \ldots, i_r \in \{1, \ldots, n\})$ then we have that

$$K(\mathbf{x}, \mathbf{t}) = (\mathbf{x}^\top \mathbf{t})^r$$

In this case $K(\mathbf{x}, \mathbf{t})$ is computed with $O(n)$ operations, which is essentially independent of $r$ or $N = n^r$. On the other hand, computing $\phi(\mathbf{x})$ requires $O(N)$ operations – Exponential in $r$!

# Redundancy of the feature map

## Warning

The feature map is not unique! If $\boldsymbol{\phi}$ generates $K$ so does $\hat{\boldsymbol{\phi}} = \mathbf{U}\boldsymbol{\phi}$ where $\mathbf{U}$ in an (any!) $N \times N$ orthogonal matrix. Even the dimension of $\boldsymbol{\phi}$ is not unique!

## Example

If $n = 2$, $K(\mathbf{x}, \mathbf{t}) = (\mathbf{x}^\top \mathbf{t})^2$ is generated by both $\boldsymbol{\phi}(\mathbf{x}) = (x_1^2, x_2^2, x_1 x_2, x_2 x_1)$ and $\hat{\boldsymbol{\phi}}(\mathbf{x}) = (x_1^2, x_2^2, \sqrt{2} x_1 x_2)$.

# Regularization-based learning algorithms

Let us open a short parenthesis and show that the dual form of ridge regression holds true for other loss functions as well

$$\mathcal{E}_{\mathrm{emp}_\lambda}(\mathbf{w}) = \sum_{i=1}^{m} V(y_i, \langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle) + \lambda \langle \mathbf{w}, \mathbf{w} \rangle, \quad \lambda > 0 \tag{6}$$

where $V : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is a loss function

**Theorem**

If $V$ is differentiable wrt. its second argument and $\mathbf{w}$ is a minimizer of $E_\lambda$ then it has the form

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i \phi(\mathbf{x}_i) \quad \Rightarrow \quad f(\mathbf{x}) = \langle \mathbf{w}, \phi(\mathbf{x}) \rangle = \sum_{i=1}^{m} \alpha_i K(\mathbf{x}_i, \mathbf{x})$$

This result is usually called the *Representer Theorem*

# Representer theorem

Setting the derivative of $E_\lambda$ wrt. $\mathsf{w}$ to zero we have

$$-\sum_{i=1}^{m} V'(y_i, \langle \mathsf{w}, \phi(\mathsf{x}_i)\rangle)\phi(\mathsf{x}_i) + 2\lambda \mathsf{w} = 0 \Rightarrow \mathsf{w} = \sum_{i=1}^{m} \alpha_i \phi(\mathsf{x}_i) \tag{7}$$

where $V'$ is the partial derivative of $V$ wrt. its second argument and we defined

$$\alpha_i = \frac{1}{2\lambda} V'(y_i, \langle \mathsf{w}, \phi(\mathsf{x}_i)\rangle) \tag{8}$$

Thus we conclude that

$$f(\mathsf{x}) = \langle \mathsf{w}, \phi(\mathsf{x})\rangle = \sum_{i=1}^{m} \alpha_i \langle \phi(\mathsf{x}_i), \phi(\mathsf{x})\rangle = \sum_{i=1}^{m} \alpha_i K(\mathsf{x}, \mathsf{x}_i),$$

## Some remarks

- Plugging eq.(7) in the rhs. of eq.(8) we obtain a set of equations for the coefficients $\alpha_i$:

$$\alpha_i = \frac{1}{2\lambda} V' \left( y_i, \sum_{j=1}^{m} K(\mathsf{x}_i, \mathsf{x}_j)\alpha_j \right) , \quad i = 1, \ldots, m$$

  When $V$ is the square loss and $\boldsymbol{\phi}(\mathsf{x}) = \mathsf{x}$ we retrieve the linear eq.(3)

- Substituting eq.(7) in eq.(6) we obtain an objective function for the $\alpha$'s:

$$\sum_{i=1}^{m} V(y_i, (\mathsf{K}\boldsymbol{\alpha})_i) + \lambda \boldsymbol{\alpha}^\top \mathsf{K}\boldsymbol{\alpha}, \quad \text{where} : \mathsf{K} = (K(\mathsf{x}_i, \mathsf{x}_j))_{i,j=1}^{m}$$

**Remark:** the Representer Theorem holds true under more general conditions on $V$ (for example $V$ can be any continuous function)

# What functions are “kernels”?

## Question

Given a function $K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$, which properties of $K$ guarantee that there exists a Hilbert space $\mathcal{W}$ and a feature map $\boldsymbol{\phi} : \mathbb{R}^n \to \mathcal{W}$ such that $K(\mathbf{x}, \mathbf{t}) = \langle \boldsymbol{\phi}(\mathbf{x}), \boldsymbol{\phi}(\mathbf{t}) \rangle$?

## Note

We’ve generalized the definition of *finite-dimensional* feature maps

$$\boldsymbol{\phi} : \mathbb{R}^n \to \mathbb{R}^N$$

to now allow potentially *infinite-dimensional* feature maps

$$\boldsymbol{\phi} : \mathbb{R}^n \to \mathcal{W}$$

# Positive Semidefinite Kernel

## Definition

A function $K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ is **positive semidefinite** if it is symmetric and the matrix $(K(\mathbf{x}_i, \mathbf{x}_j) : i, j = 1, \ldots, k)$ is positive semidefinite for every $k \in \mathbb{N}$ and every $\mathbf{x}_1, \ldots, \mathbf{x}_k \in \mathbb{R}^n$

## Theorem

$K$ is positive semidefinite if and only if

$$K(\mathbf{x}, \mathbf{t}) = \langle \phi(\mathbf{x}), \phi(\mathbf{t}) \rangle, \quad \mathbf{x}, \mathbf{t} \in \mathbb{R}^n$$

for some feature map $\phi : \mathbb{R}^n \to \mathcal{W}$ and Hilbert space $\mathcal{W}$

# Positive semidefinite kernel (cont.)

## Proof of "$\Leftarrow$"

If $K(\mathbf{x}, \mathbf{t}) = \langle \phi(\mathbf{x}), \phi(\mathbf{t}) \rangle$ then we have that

$$\sum_{i,j=1}^{m} c_i c_j K(\mathbf{x}_i, \mathbf{x}_j) = \left\langle \sum_{i=1}^{m} c_i \boldsymbol{\phi}(\mathbf{x}_i), \sum_{j=1}^{m} c_j \boldsymbol{\phi}(\mathbf{x}_j) \right\rangle = \left\| \sum_{i=1}^{m} c_i \boldsymbol{\phi}(\mathbf{x}_i) \right\|^2 \geq 0$$

for every choice of $m \in \mathbb{N}$, $x_i \in \mathbb{R}^d$ and $c_i \in R$, $i = 1, \ldots, m$

## Note

the proof of '$\Rightarrow$' requires the notion of reproducing kernel Hilbert spaces. Informally, one can show that the linear span of the set of functions $\{K(\mathbf{x}, \cdot) : \mathbf{x} \in \mathbb{R}^n\}$ can be made into a Hilbert space $H_K$ with inner product induced by the definition $\langle K(\mathbf{x}, \cdot), K(\mathbf{t}, \cdot) \rangle := K(\mathbf{x}, \mathbf{t})$. In particular, the map $\boldsymbol{\phi} : \mathbb{R}^n \to H_K$ defined as $\boldsymbol{\phi}(\mathbf{x}) = K(\mathbf{x}, \cdot)$ is a feature map associated with $K$. Observe then with $f(\cdot) := \sum_{i=1}^{m} \alpha_i K(\mathbf{x}_i, \cdot)$ that $\|f\|^2 = \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j)$.

# Two Example Kernels

## Polynomial Kernel(s)

If $p : \mathbb{R} \to \mathbb{R}$ is a polynomial with nonnegative coefficients then $K(\mathbf{x}, \mathbf{t}) = p(\mathbf{x}^\top \mathbf{t}), \mathbf{x}, \mathbf{t} \in \mathbb{R}^n$ is a positive semidefinite kernel. For example if $a \geq 0$

- $K(\mathbf{x}, \mathbf{t}) = (\mathbf{x}^\top \mathbf{t})^r$
- $K(\mathbf{x}, \mathbf{t}) = (a + \mathbf{x}^\top \mathbf{t})^r$
- $K(\mathbf{x}, \mathbf{t}) = \sum_{i=0}^{d} \frac{a^i}{i!} (\mathbf{x}^\top \mathbf{t})^i$

are each positive semidefinite kernels.

## Gaussian Kernel

An important example of a “radial” kernel is the Gaussian kernel

$$K(\mathbf{x}, \mathbf{t}) = \exp(-\beta \|\mathbf{x} - \mathbf{t}\|^2), \quad \beta > 0, \mathbf{x}, \mathbf{t} \in \mathbb{R}^n$$

note: any corresponding feature map $\phi(\cdot)$ is $\infty$-dimensional.

# Polynomial and Anova Kernel

## Anova Kernel

$$K_a(\mathbf{x}, \mathbf{t}) = \prod_{i=1}^{n} (1 + x_i t_i)$$

Compare to the polynomial kernel $K_p(\mathbf{x}, \mathbf{t}) = (1 + \mathbf{x}^\top \mathbf{t})^d$

$$\mathbf{x} = \begin{matrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_n \end{matrix} \Rightarrow \boldsymbol{\phi}_p(\mathbf{x}) = \begin{matrix} 1 \\ \sqrt{d}x_1 \\ \sqrt{d}x_2 \\ \vdots \\ \sqrt{d}x_n \\ \sqrt{d(d-1)}x_1 x_2 \\ \vdots \\ \sqrt{\binom{d}{i_0, i_1, \ldots, i_n} x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n}} \\ \vdots \end{matrix} \qquad \mathbf{x} = \begin{matrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_n \end{matrix} \Rightarrow \boldsymbol{\phi}_a(\mathbf{x}) = \begin{matrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_n \\ x_1 x_2 \\ \vdots \\ \vdots \\ x_1 x_2 \ldots x_n \end{matrix}$$

where $\sum_{j=0}^{n} i_j = d$

# Kernel construction

Which operations/combinations (eg, products, sums, composition, etc.) of a given set of kernels is still a kernel?

If we address this question we can build more interesting kernels starting from simple ones

**Example**

We have already seen that $K(\mathbf{x}, \mathbf{t}) = (\mathbf{x}^\top \mathbf{t})^r$ is a kernel. For which class of functions $p : \mathbb{R} \to \mathbb{R}$ is $p(\mathbf{x}^\top \mathbf{t})$ a kernel? More generally, if $K$ is a kernel when is $p(K(\mathbf{x}, \mathbf{t}))$ a kernel?

# General linear kernel

If $\mathbf{A}$ is an $n \times n$ psd matrix the function $K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ defined by

$$K(\mathbf{x}, \mathbf{t}) = \mathbf{x}^\top \mathbf{A} \mathbf{t}$$

is a kernel

**Proof**

Since $\mathbf{A}$ is psd we can write it in the form $\mathbf{A} = \mathbf{R}\mathbf{R}^\top$ for some $n \times n$ matrix $\mathbf{R}$. Thus $K$ is represented by the feature map $\phi(\mathbf{x}) = \mathbf{R}^\top \mathbf{x}$

Alternatively, note that:

$$\sum_{i,j} c_i c_j \mathbf{x}_i^\top \mathbf{A} \mathbf{x}_j = \sum_{i,j} c_i c_j (\mathbf{R}^\top \mathbf{x}_i)^\top (\mathbf{R}^\top \mathbf{x}_j) = \| \sum_i c_i \mathbf{R}^\top \mathbf{x}_i \|^2 \geq 0$$

# Kernel composition

More generally, if $K : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}$ is a kernel and $\boldsymbol{\phi} : \mathbb{R}^n \to \mathbb{R}^N$, then

$$\tilde{K}(\mathbf{x}, \mathbf{t}) = K(\boldsymbol{\phi}(\mathbf{x}), \boldsymbol{\phi}(\mathbf{t}))$$

is a kernel

**Proof**

By hypothesis, $K$ is a kernel and so, for every $\mathbf{x}_1, \ldots, \mathbf{x}_m \in \mathbb{R}^n$ the matrix $(K(\boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_j)) : i, j = 1, \ldots, m)$ is psd

In particular, the above example corresponds to $K(\mathbf{x}, \mathbf{t}) = \mathbf{x}^\top \mathbf{t}$ and $\boldsymbol{\phi}(\mathbf{x}) = \mathbf{R}^\top \mathbf{x}$

**Question**

If $K_1, \ldots, K_q$ are kernels on $\mathbb{R}^n$ and $F : \mathbb{R}^q \to \mathbb{R}$, when is the function

$$F(K_1(\mathbf{x}, \mathbf{t}), \ldots, K_q(\mathbf{x}, \mathbf{t})), \quad \mathbf{x}, \mathbf{t} \in \mathbb{R}^n$$

a kernel?

Equivalently: when for every choice of $m \in \mathbb{N}$ and $\mathsf{A}_1, \ldots, \mathsf{A}_q$ $m \times m$ psd matrices, is the following matrix psd?

$$(F(A_{1,ij}, \ldots, A_{q,ij}) : i, j = 1, \ldots m)$$

We discuss some examples of functions $F$ for which the answer to these question is YES

# Nonnegative combination of kernels

If $\lambda_j \geq 0, j = 1, \ldots, q$ then $\sum_{j=1}^{q} \lambda_j K_j$ is a kernel

This fact is immediate (a non-negative combination of psd matrices is still psd)

**Example:** Let $q = n$ and $K_i(\mathbf{x}, \mathbf{t}) = x_i t_i$.

In particular, this implies that

- $aK_1$ is a kernel if $a \geq 0$
- $K_1 + K_2$ is a kernel

# Product of kernels

The pointwise product of two kernels $K_1$ and $K_2$

$$K(\mathbf{x}, \mathbf{t}) := K_1(\mathbf{x}, \mathbf{t}) K_2(\mathbf{x}, \mathbf{t}), \quad \mathbf{x}, \mathbf{t} \in \mathbb{R}^d$$

is a kernel

## Proof

We need to show that if $\mathbf{A}$ and $\mathbf{B}$ are psd matrices, so is $C = (A_{ij} B_{ij} : i, j = 1, \ldots, m)$ ($C$ is also called the Schur product of $\mathbf{A}$ and $\mathbf{B}$). We write $\mathbf{A}$ and $\mathbf{B}$ in their singular value form, $\mathbf{A} = \mathbf{U} \boldsymbol{\Sigma} \mathbf{U}^\top$, $\mathbf{B} = \mathbf{V} \boldsymbol{\Lambda} \mathbf{V}^\top$ where $\mathbf{U}, \mathbf{V}$ are orthogonal matrices and $\boldsymbol{\Sigma} = diag(\sigma_1, \ldots, \sigma_m)$, $\Lambda = diag(\lambda_1, \ldots, \lambda_m)$, $\sigma_i, \lambda_i \geq 0$. We have

$$\begin{aligned}
\sum_{i,j=1}^{m} a_i a_j C_{ij} &= \sum_{ij} a_i a_j \sum_r \sigma_r U_{ir} U_{jr} \sum_s \lambda_s V_{is} V_{js} \\
&= \sum_{rs} \sigma_r \lambda_s \sum_i a_i U_{ir} V_{is} \sum_j a_j U_{jr} V_{js} \\
&= \sum_{rs} \sigma_r \lambda_s (\sum_i a_i U_{ir} V_{is})^2 \geq 0
\end{aligned}$$

# Summary of constructions

**Theorem**

If $K_1, K_2$ are kernels, $a \geq 0$, $A$ is a symmetric positive semi-definite matrix, $K$ a kernel on $\mathbb{R}^N$ and $\boldsymbol{\phi} : \mathbb{R}^n \to \mathbb{R}^N$ then the following functions are positive semidefinite kernels on $\mathbb{R}^n$

1. $\mathsf{x}^\top A \mathsf{t}$
2. $K_1(\mathsf{x}, \mathsf{t}) + K_2(\mathsf{x}, \mathsf{t})$
3. $aK_1(\mathsf{x}, \mathsf{t})$
4. $K_1(\mathsf{x}, \mathsf{t})K_2(\mathsf{x}, \mathsf{t})$
5. $K(\phi(\mathsf{x}), \phi(\mathsf{t}))$

# Polynomial of kernels

Let $F = p$ where $p : \mathbb{R}^q \to \mathbb{R}$ is a polynomial in $q$ variables with nonnegative coefficients. By properties 1,2 and 3 above we conclude that $p$ is a valid function

In particular if $q = 1$,

$$\sum_{i=1}^{d} a_i(K(\mathrm{x}, \mathrm{t}))^i$$

is a kernel if $a_1, \ldots, a_d \geq 0$

# Polynomial kernels

The above observation implies that if $p : \mathbb{R} \to \mathbb{R}$ is a polynomial with nonnegative coefficients then $p(\mathsf{x}^\top \mathsf{t}), \mathsf{x}, \mathsf{t} \in \mathbb{R}^n$ is a kernel on $\mathbb{R}^n$. In particular if $a \geq 0$ the following are valid polynomial kernels

- $(\mathsf{x}^\top \mathsf{t})^r$
- $(a + \mathsf{x}^\top \mathsf{t})^r$
- $\sum_{i=0}^{d} \frac{a^i}{i!} (\mathsf{x}^\top \mathsf{t})^i$

# 'Infinite polynomial' kernel

If in the last equation we set $r = \infty$ the series

$$\sum_{i=0}^{r} \frac{a^i}{i!}(\mathbf{x}^\top \mathbf{t})^i$$

converges everywhere uniformly to $\exp(a\mathbf{x}^\top \mathbf{t})$ showing that this function is also a kernel.

Assume for simplicity that $n = 1$. A feature map corresponding to the kernel $\exp(axt)$ is

$$\phi(x) = \left(1,\ \sqrt{a}x,\ \sqrt{\frac{a}{2}}x^2,\ \sqrt{\frac{a^3}{6}}x^3, \ldots\right) = \left(\sqrt{\frac{a^i}{i!}}x^i : i \in \mathbb{N}\right)$$

- The feature space has an infinite dimensionality!

# Translation invariant and radial kernels

We say that a kernel $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is

- *Translation invariant* if it has the form

$$K(\mathbf{x}, \mathbf{t}) = H(\mathbf{x} - \mathbf{t}), \quad \mathbf{x}, \mathbf{t} \in \mathbb{R}^d$$

  where $H : \mathbb{R}^d \to \mathbb{R}$ is a differentiable function

- *Radial* if it has the form

$$K(\mathbf{x}, \mathbf{t}) = h(\|\mathbf{x} - \mathbf{t}\|), \quad \mathbf{x}, \mathbf{t} \in \mathbb{R}^d$$

  where $h : [0, \infty) \to [0, \infty)$ is a differentiable function

# The Gaussian kernel

An important example of a radial kernel is the Gaussian kernel

$$K(\mathbf{x}, \mathbf{t}) = \exp(-\beta \|\mathbf{x} - \mathbf{t}\|^2), \quad \beta > 0, \mathbf{x}, \mathbf{t} \in \mathbb{R}^d$$

It is a kernel because it is the product of two kernels

$$K(\mathbf{x}, \mathbf{t}) = (\exp(-\beta(\mathbf{x}^\top \mathbf{x} + \mathbf{t}^\top \mathbf{t}))) \exp(2\beta \mathbf{x}^\top \mathbf{t})$$

(We saw before that $\exp(2\beta \mathbf{x}^\top \mathbf{t})$ is a kernel. Clearly $\exp(-\beta(\mathbf{x}^\top \mathbf{x} + \mathbf{t}^\top \mathbf{t}))$ is a kernel with one-dimensional feature map $\phi(\mathbf{x}) = \exp(-\beta \mathbf{x}^\top \mathbf{x})$)

**Exercise:**

Can you find a feature map representation for the Gaussian kernel?

# The min kernel

We give another example of a kernel.

$$K_{\min}(x,t) := \min(x,t)$$

with $x, t \in [0, \infty)$. We argue informally that this is the kernel associated with the Hilbert space $\mathcal{H}_{\min}$ of all functions with the following four properties.

1. $f : [0,\infty) \to \mathbb{R}$
2. $f(0) = 0$
3. $f$ is absolutely continuous (hence $f(b) - f(a) = \int_a^b f'(x)dx$ )
4. $\|f\| = \sqrt{\int_0^\infty [f'(x)]^2 \, dx}$

# Proof sketch

## Proof sketch

Our argument is simplified as follows,

1. We argue only that the induced norms are the same.
2. We only consider $f \in \mathcal{H}_{\min}$ s.t. $f(x) = \sum_{i=1}^{m} \alpha_i \min(x_i, x)$.

Define $h_c(x) = [x \leq c]$ i.e., $h_c(x) = [\min(c, x)]'$

$$\begin{aligned}
\|f\|^2 &= \int_0^\infty [f'(x)]^2 \, dx \\
&= \int_0^\infty [(\sum_{i=1}^{m} \alpha_i \min(x_i, x))']^2 \, dx \\
&= \int_0^\infty [(\sum_{i=1}^{m} \alpha_i h_{x_i}(x))]^2 \, dx \\
&= \sum_{i,j}^{m} \alpha_i \alpha_j \int_0^\infty h_{x_i}(x) h_{x_j}(x) \, dx = \sum_{i,j}^{m} \alpha_i \alpha_j \min(x_i, x_j)
\end{aligned}$$

# Computational Summary

# Summary : Computation with Basis Functions

Data: $X, \quad (m \times n)$; $\mathbf{y}, \quad (m \times 1)$
Basis Functions: $\phi_1, \ldots, \phi_N$ where $\phi_i : \mathbb{R}^n \to \mathbb{R}$
Feature Map: $\boldsymbol{\phi} : \mathbb{R}^n \to \mathbb{R}^N$

$$\boldsymbol{\phi}(\mathbf{x}) = (\phi_1(\mathbf{x}), \ldots, \phi_N(\mathbf{x})), \quad \mathbf{x} \in \mathbb{R}^n$$

Mapped Data Matrix:

$$\Phi := \begin{pmatrix} \boldsymbol{\phi}(\mathbf{x}_1) \\ \vdots \\ \boldsymbol{\phi}(\mathbf{x}_m) \end{pmatrix} = \begin{pmatrix} \phi_1(\mathbf{x}_1) & \ldots & \phi_N(\mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ \phi_1(\mathbf{x}_m) & \ldots & \phi_N(\mathbf{x}_m) \end{pmatrix}, \quad (m \times N)$$

Regression Coefficients: $\mathbf{w} = (\Phi^\top \Phi + \lambda I_N)^{-1} \Phi^\top \mathbf{y}$
Regression Function: $\hat{y}(\mathbf{x}) = \sum_{i=1}^{N} w_i \phi_i(\mathbf{x})$

# Summary : Computation with Kernels

Data: $X, \quad (m \times n)$; $\mathbf{y}, \quad (m \times 1)$
Kernel Function: $K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$
Kernel Matrix:

$$\mathsf{K} := \begin{pmatrix} K(\mathbf{x}_1, \mathbf{x}_1) & \dots & K(\mathbf{x}_1, \mathbf{x}_m) \\ \vdots & \ddots & \vdots \\ K(\mathbf{x}_m, \mathbf{x}_1) & \dots & K(\mathbf{x}_m, \mathbf{x}_m) \end{pmatrix}, \quad (m \times m)$$

Regression Coefficients: $\boldsymbol{\alpha} = (\mathsf{K} + \lambda I_m)^{-1}\mathbf{y}$
Regression Function: $\hat{y}(\mathbf{x}) = \sum_{i=1}^{m} \alpha_i K(\mathbf{x}_i, \mathbf{x})$