

Statistical Machine Learning

Lecture 4: Basis Expansion and Regularization

W.Q.Cui Research Group

Department of Statistics and Finance
University of Science and Technology of China

2018 Autumn

Outline

1 Introduction

- Moving beyond linearity
- Controlling the complexity of the model

2 Piecewise Polynomials and Splines

- Piecewise polynomial function
- Splines: Cubic Spline, Regression Splines
- Natural Cubic Splines

3 Smoothing Splines

- regularization, Smoothing parameter
- Degrees of Freedom and Smoother Matrices
- Effective dof, eigen-analysis

Outline

4 Nonparametric Logistic Regression

- Back to logistic regression
- Penalized log-likelihood criterion

5 Regularization and Reproducing Kernel Hilbert Spaces

- Types of Kernels
- Back to Regularization
- Regularization and SVM

Moving beyond linearity

- Regularization is a process of introducing additional information in order to solve an ill-posed problem or to prevent overfitting.
- In mathematics, statistics, and computer science, particularly in the fields of machine learning and inverse problems.
- A regularization term (or regularizer) $J(f)$ is added to a loss function:

$$\min_f \sum_{i=1}^n L(y_i, f(x_i)) + \lambda J(f).$$

- λ is a parameter which controls the importance of the regularization term.
- $J(f)$ is typically chosen to impose a penalty on the complexity of f .
- Concrete notions of complexity used include restrictions for smoothness and bounds on the vector space norm.

Moving beyond linearity

Main idea

- Augment the vector of inputs X with additional variables.
- These are transformations of X

$$h_m(X) : \mathbb{R}^p \rightarrow \mathbb{R}$$

with $m = 1, 2, \dots, M$.

- Then model the relationship between X and Y

$$f(X) = \sum_{m=1}^M \beta_m h_m(X) = \sum_{m=1}^M \beta_m Z_m$$

as a **linear basis expansion** in X .

- Have a linear model *w.r.t.* Z . Can use the same methods as before.

Which transformations?

Some examples

- Linear:

$$h_m(X) = X_m, \quad m = 1, 2, \dots, p$$

- Polynomial:

$$h_m(X) = X_j^2 \quad \text{or} \quad h_m(X) = X_j X_k$$

- Non-linear transformation of single inputs:

$$h_m(X) = \log(X_j), \sqrt{X_j}, \dots$$

- Non-linear transformation of multiple inputs:

$$h_m(X) = \|X\|$$

- Use of Indicator functions:

$$h_m(X) = \text{Ind}(L_m \leq X_k \leq U_m)$$

Pros and Cons of this augmentation

Pros

- Can model more complicated decision boundaries.
- Can model more complicated regression relationships.

Cons

- Lack of locality in global basis functions.
 - **Solution** Use local polynomial representations such as piecewise-polynomials and splines.
- How should one find the correct complexity in the model?
- There is the danger of over-fitting.

Controlling the complexity of the model

Common approaches taken:

- **Restriction Methods**

Limit the class of functions considered. Use additive models

$$f(X) = \sum_{j=1}^p \sum_{m=1}^{M_j} \beta_{jm} h_{jm}(X_j)$$

- **Selection Methods**

Scan the set of h_m and only include those that contribute significantly to the fit of the model - Boosting, CART.

- **Regularization Methods**

Let

$$f(X) = \sum_{j=1}^M \beta_j h_j(X)$$

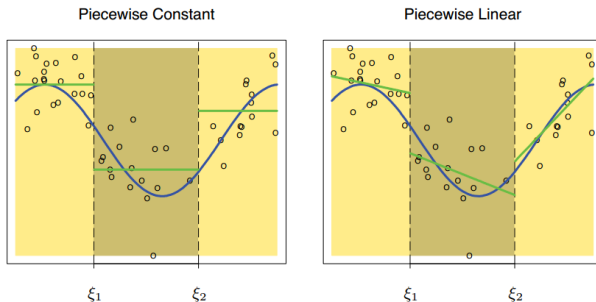
but when learning the β'_j s restrict their values in the manner of *ridge regression* and *lasso*.

Piecewise polynomial function

To obtain a **piecewise polynomial function** $f(x)$

- Divide the domain of X into contiguous intervals.
- Represent f by a separate polynomial in each interval.

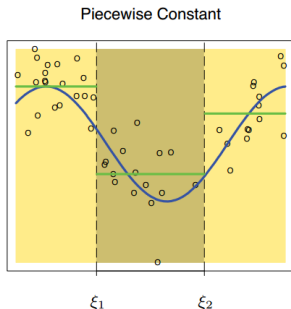
Example



Blue curve - ground truth function.

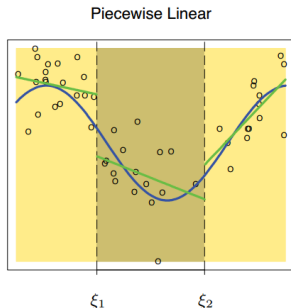
Green curve - piecewise constant/linear fit to the training data.

Example: Piecewise constant function



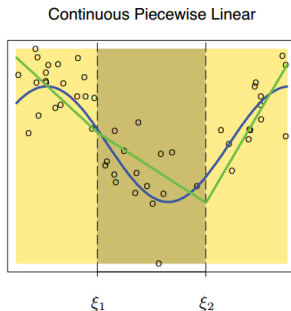
- Divide $[a, b]$, the domain of X , into three regions
 $[a, \xi_1)$, $[\xi_1, \xi_2)$, $[\xi_2, b]$ with $\xi_1 < \xi_2 < \xi_3$
 ξ'_i 's are referred to as **knots**
- Define three basis functions
 $h_1(X) = I(X < \xi_1)$, $h_2(X) = I(\xi_1 \leq X < \xi_2)$, $h_3(X) = I(\xi_2 \leq X)$
- The model $f(X) = \sum_{m=1}^3 \beta_m h_m(X)$ is fit using least-squares.
- As basis functions don't overlap $\Rightarrow \hat{\beta}_m = \text{mean of } y'_i \text{ in the } m\text{th region.}$

Example: Piecewise linear function



- In this case define 6 basis functions
 $h_1(X) = I(X < \xi_1)$, $h_2(X) = I(\xi_1 \leq X < \xi_2)$, $h_3(X) = I(\xi_2 \leq X)$
 $h_4(X) = Xh_1(X)$, $h_5(X) = Xh_2(X)$, $h_6(X) = Xh_3(X)$
- The model $f(X) = \sum_{m=1}^6 \beta_m h_m(X)$ is fit using least-squares.
- As basis functions don't overlap \implies fit a separate linear model to the data in each region.

Example: Continuous piecewise linear function



- Additionally impose the constraint that $f(X)$ is continuous as ξ_1 and ξ_2 .
- This means

$$\beta_1 + \beta_2\xi_1 = \beta_3 + \beta_4\xi_1, \text{ and}$$

$$\beta_3 + \beta_4\xi_2 = \beta_5 + \beta_6\xi_2$$

- This reduces the # of dof of $f(X)$ from 6 to 4.

A more compact set of basis functions

- To impose the continuity constraints directly can use this basis instead:

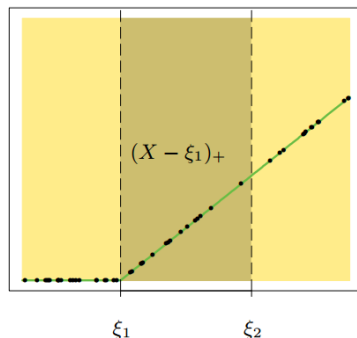
$$h_1(X) = 1$$

$$h_2(X) = X$$

$$h_3(X) = (X - \xi_1)_+$$

$$h_4(X) = (X - \xi_2)_+$$

Piecewise-linear Basis Function

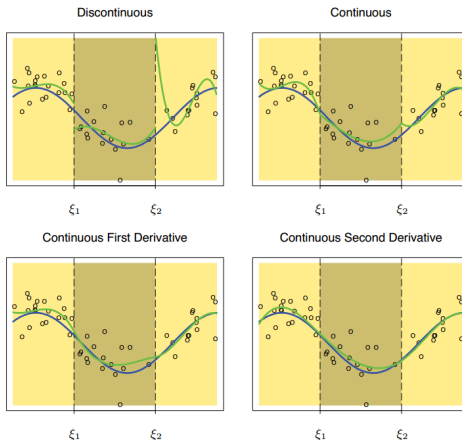


Smoother $f(X)$

Can achieve a smoother $f(X)$ by increasing the order

- of the local polynomials
- of the continuity at the knots

Piecewise-cubic polynomials with increasing orders of continuity



Cubic Spline

$f(X)$ is a **cubic spline** if

- it is a piecewise cubic polynomial **and**
- has 1st and 2nd continuity at the knots

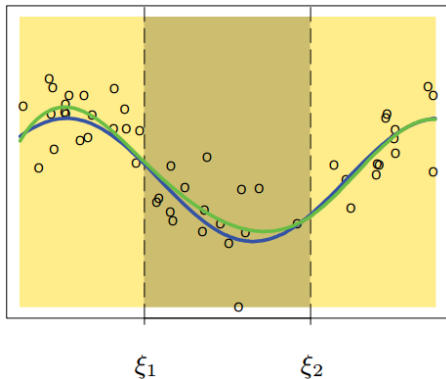
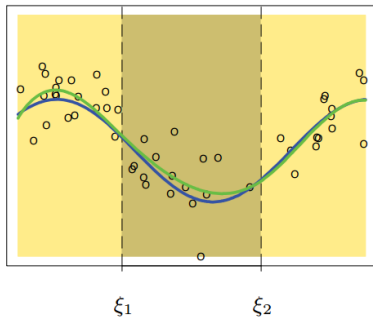


Figure: A Cubic spline

Cubic Spline

Figure: A Cubic spline



The following basis represents a cubic spline with knots at ξ_1 and ξ_2 :

$$h_1(X) = 1, \quad h_3(X) = X^2, \quad h_5(X) = (X - \xi_1)_+^3$$

$$h_2(X) = X, \quad h_4(X) = X^3, \quad h_6(X) = (X - \xi_2)_+^3$$

Order M spline

- An order M spline with knots $\xi_1, \xi_2, \dots, \xi_K$ is
 - a piecewise-polynomial of order M **and**
 - has continuous derivatives up to order $M - 2$
- The general form for the truncated-power basis set is

$$h_j(X) = X^{j-1} \quad , \quad j = 1, 2, \dots, M$$

$$h_{M+l}(X) = (X - \xi_l)_+^{M-1} \quad , \quad l = 1, 2, \dots, K$$

- In practice the most widely used orders are $M = 1, 2, 4$.

Regression Splines

- Fixed-knot splines are known as **regression splines**.
- For a regression spline one needs to select
 - the order of the spline,
 - the number of knots and
 - the placement of the knots.
- One common approach is to set a knot at each observation x_i
- There are many equivalent bases for representing splines and the **truncated power basis** is **intuitively attractive** but **not computationally attractive**.
- A better basis set for implementation is the B-spline basis set.

Natural Cubic Splines

Problem

The polynomials fit beyond the boundary knots behave wildly.

Solution : Natural Cubic Splines

- Have the additional constraints that the function is linear beyond the boundary knots.
- This frees up 4 dof which can be used by having more knots in the interior region.
- Near the boundaries one has reduced the variance of the fit but increased its bias!

Smoothing Splines

- Avoid knot selection problem by using a maximal set of knots.
- **Complexity of the fit** is controlled by **regularization**.
- Consider the following problem:
Find the function $f(x)$ with continuous second derivative which minimizes

$$\text{RSS}(f, \lambda) = \sum_{i=1}^n (y_i - f(x_i))^2 + \lambda \int (f''(t))^2 dt$$

closeness to data

smoothing parameter

curvature penalty

Smoothing Splines: Smoothing parameter

$$\text{RSS}(f, \lambda) = \sum_{i=1}^n (y_i - f(x_i))^2 + \lambda \int (f''(t))^2 dt$$

- λ establishes a trade-off between predicting the training data and minimizing the curvature of $f(x)$
- The two special cases are
 - * $\lambda = 0$: \hat{f} is any function which interpolates the data.
 - * $\lambda = \infty$: \hat{f} is the simple least squares line fit.
- In these two cases go from very rough to very smooth $\hat{f}(x)$.
- Hope is $\lambda \in (0, \infty)$ indexes an interesting class of functions in between.

Smoothing Splines: Form of the solution

$$\text{RSS}(f, \lambda) = \sum_{i=1}^n (y_i - f(x_i))^2 + \lambda \int (f''(t))^2 dt$$

- **Amazingly** the above equation has an explicit, finite-dimensional unique minimizer for a fixed λ .
- It is a natural cubic spline with knots as the unique values of the x_i , $i = 1, 2, \dots, n$.
- That is

$$\hat{f}(x) = \sum_{j=1}^n N_j(x) \theta_j$$

where the $N_j(x)$ are an N -dimensional set of basis functions for representing this family of natural splines.

Smoothing Splines: Estimating the coefficients

The criterion to be optimized thus reduces to

$$\text{RSS}(\theta, \lambda) = (y - \mathbf{N}\theta)^t(y - \mathbf{N}\theta) + \lambda \theta^t \Omega_N \theta$$

where

$$N = \begin{pmatrix} N_1(x_1) & N_2(x_1) & \cdots & N_n(x_1) \\ N_1(x_2) & N_2(x_2) & \cdots & N_n(x_2) \\ \vdots & \vdots & \cdots & \vdots \\ N_1(x_n) & N_2(x_n) & \cdots & N_n(x_n) \end{pmatrix}$$

$$\Omega_N = \begin{pmatrix} \int N_1''(t)N_1''(t)dt & \int N_1''(t)N_2''(t)dt & \cdots & \int N_1''(t)N_n''(t)dt \\ \int N_2''(t)N_1''(t)dt & \int N_2''(t)N_2''(t)dt & \cdots & \int N_2''(t)N_n''(t)dt \\ \vdots & \vdots & \cdots & \vdots \\ \int N_n''(t)N_1''(t)dt & \int N_n''(t)N_2''(t)dt & \cdots & \int N_n''(t)N_n''(t)dt \end{pmatrix}$$

$$y = (y_1, y_2, \dots, y_n)^t$$

Smoothing Splines: Estimating the coefficients

The criterion to be optimized thus reduces to

$$\text{RSS}(\theta, \lambda) = (y - \mathbf{N}\theta)^t(y - \mathbf{N}\theta) + \lambda \theta^t \Omega_N \theta$$

and its solution is given by

$$\hat{\theta} = (\mathbf{N}^t \mathbf{N} + \lambda \Omega_N)^{-1} \mathbf{N}^t y$$

The fitted smoothing spline is then given by

$$\hat{f}(x) = \sum_{j=1}^n N_j(x) \hat{\theta}_j$$

A smoothing spline is a linear smoother

- Assume that λ has been set.
- Remember the estimated coefficients $\hat{\theta}$ are a linear combination of the y_i 's

$$\hat{\theta} = (\mathbf{N}^T \mathbf{N} + \lambda \Omega_N)^{-1} \mathbf{N}^T y$$

- Let $\hat{\mathbf{f}}$ be the n -vector of the fitted values $\hat{f}(x_i)$ then

$$\hat{\mathbf{f}} = \mathbf{N} \hat{\theta} = \mathbf{N} (\mathbf{N}^T \mathbf{N} + \lambda \Omega_N)^{-1} \mathbf{N}^T y = S_\lambda y$$

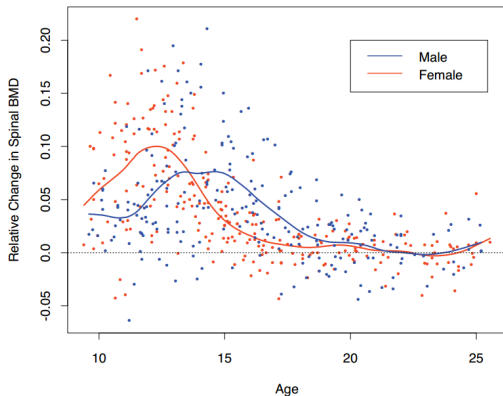
where $S_\lambda = (\mathbf{N}^T \mathbf{N} + \lambda \Omega_N)^{-1} \mathbf{N}^T$.

Properties of S_λ

- S_λ is symmetric and positive semi-definite.
- $S_\lambda S_\lambda \preceq S_\lambda$
- S_λ has rank n .
- The book defines the effective degrees of freedom of a smoothing spline to be

$$\text{df}_\lambda = \text{trace}(S_\lambda)$$

Effective dof of a smoothing spline



Both curves were fit with $\lambda \approx 0.00022$. This choice corresponds to about 12 degrees of freedom.

The eigen-decomposition of $S_\lambda: S_\lambda$ in Reinsch form

- Let $N = USV^T$ be the **SVD** of N .
- Using this decomposition it is straightforward to re-write

$$S_\lambda = (\mathbf{N}^T \mathbf{N} + \lambda \Omega_N)^{-1} \mathbf{N}^T$$

as

$$S_\lambda = (1 + \lambda K)^{-1}$$

where

$$K = US^{-1}V^T \Omega_N VS^{-1}U^T$$

- It is also easy to show that $\hat{\mathbf{f}} = S_\lambda \mathbf{y}$ is the solution to the optimization problem

$$\min_{\mathbf{f}} (\mathbf{y} - \mathbf{f})^T (\mathbf{y} - \mathbf{f}) + \lambda \mathbf{f}^T K \mathbf{f}$$

The eigen-decomposition of S_λ

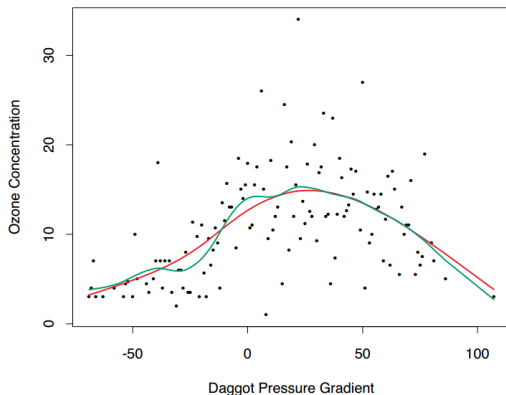
- Let $K = PDP^{-1}$ be the real eigen-decomposition of K possible as K symmetric and positive semi-definite.
- Then

$$\begin{aligned} S_\lambda &= (I + \lambda K)^{-1} = (I + \lambda PDP^{-1})^{-1} \\ &= (PP^{-1} + \lambda PDP^{-1})^{-1} \\ &= (P(I + \lambda D)P^{-1})^{-1} \\ &= P(I + \lambda D)^{-1}P^{-1} \\ &= \sum_{i=1}^n \frac{1}{1 + \lambda d_k} p_k p_k^T \end{aligned}$$

where d_k are the elements of diagonal D and e-values of K and p_k are the e-vectors of K .

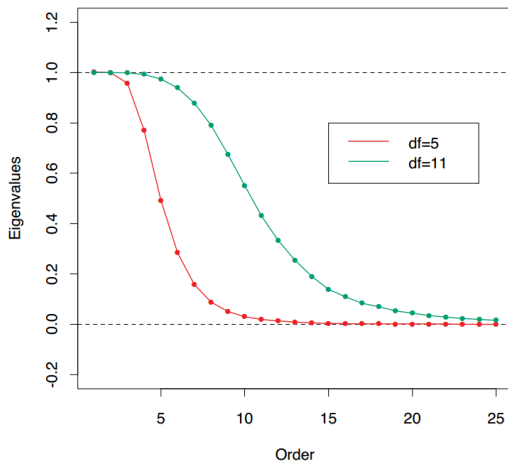
- p_k are also the e-vectors of S_λ and $1/(1 + \lambda d_k)$ its e-values.

Example: Cubic spline smoothing to air pollution data



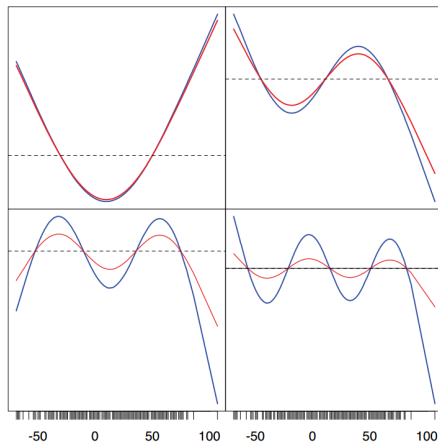
- **Green curve** smoothing spline with $df_{\lambda} = \text{trace}(S_{\lambda}) = 11$
- **Red curve** smoothing spline with $df_{\lambda} = \text{trace}(S_{\lambda}) = 5$

Example: Eigenvalues of S_λ



- Green curve eigenvalues of S_λ with $df_\lambda = 11$
- Red curve eigenvalues of S_λ with $df_\lambda = 5$

Example: Eigenvectors of S_λ



- Each **blue curve** is an eigenvector of S_λ plotted against x . Top left has highest e-value, bottom right smallest.
- **Red curve** is eigenvector damped by $1/(1 + \lambda d_k)$

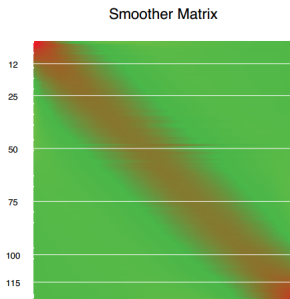
Highlights of the eigenrepresentation

- The eigenvectors of S_λ do not depend on λ .
- The smoothing spline decomposes y w.r.t. the basis $\{p_k\}$ and shrinks the contributions using $1/(1 + \lambda d_k)$ as

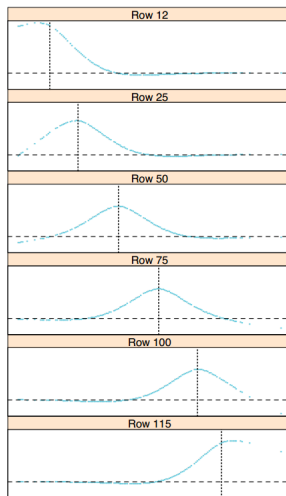
$$S_\lambda y = \sum_{k=1}^n \frac{1}{1 + \lambda d_k} p_k(p_k^t y)$$

- The first two e-values are always 1 of S_λ and correspond to the eigenspace of functions linear in x .
- The sequence of p_k , ordering by decreasing $1/(1 + \lambda d_k)$, appear to increase in complexity.
- $df_\lambda = \text{trace}(S_\lambda) = \sum_{k=1}^n 1/(1 + \lambda d_k)$.

Visualization of a S_λ



Equivalent Kernels



Choosing λ

- This is a crucial and tricky problem.
- Will deal with this problem in Chapter 7 when we consider the problem of **Model Selection**.

Back to logistic regression

- Previously considered a binary classifier s.t.

$$\log \frac{P(Y = 1|X = x)}{P(Y = 0|X = x)} = \beta_0 + \beta^t x$$

- However, consider the case when

$$\log \frac{P(Y = 1|X = x)}{P(Y = 0|X = x)} = f(x)$$

which in turn implies

$$P(Y = 1|X = x) = \frac{e^{f(x)}}{1 + e^{f(x)}}$$

- Fitting $f(x)$ in a smooth fashion leads to a smooth estimate of $P(Y = 1|X = x)$.

The penalized log-likelihood criterion

Construct the penalized log-likelihood criterion

$$\begin{aligned} l(f; \lambda) &= \sum_{i=1}^n \left[y_i \log P(Y = 1|x_i) + (1 - y_i) \log(1 - P(Y = 1|x_i)) \right] \\ &\quad - \frac{1}{2} \lambda \int (f''(t))^2 dt \\ &= \sum_{i=1}^n \left[y_i f(x_i) - \log(1 + e^{f(x_i)}) \right] - \frac{1}{2} \lambda \int (f''(t))^2 dt \end{aligned}$$

General class of regularization problems

There is a class of generalization problems which have the form

$$\min_{f \in \mathcal{H}} \left[\sum_{i=1}^n L(y_i, f(x_i)) + \lambda J(f) \right]$$

where

- $L(y_i, f(x_i))$ is a loss function,
- $J(f)$ is a penalty functional,
- \mathcal{H} is a space of functions on which $J(f)$ is defined.

Important subclass of problems of this form

- These are generated by a **positive definite kernel** $K(x, y)$ and
- the corresponding space of functions \mathcal{H}_K called a **reproducing kernel Hilbert space**(RKHS),
- the penalty functional J is defined in terms of the kernel as well.

What does all this mean?

What follows is mainly based on the notes of [Nuno Vasconcelos](#)

Types of Kernels

Definition

A kernel is a mapping $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$.

These three types of kernels are equivalent

dot-product kernel



positive definite kernel



Mercer kernel

Dot-product kernel

Definition

A mapping

$$k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$$

is a **dot-product kernel** if and only if

$$k(x, y) = \langle \Phi(x), \Phi(y) \rangle$$

where

$$\Phi : \mathcal{X} \rightarrow \mathcal{H}$$

and \mathcal{H} is a vector space and $\langle \cdot, \cdot \rangle$ is an inner-product on \mathcal{H} .

Positive definite kernel

Definition

A mapping

$$k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$$

is a **positive semi-definite kernel** on $\mathcal{X} \times \mathcal{X}$ if $\forall m \in \mathbb{N}$ and $\forall x_1, \dots, x_m$ with each $x_i \in \mathcal{X}$ the *Gram matrix*

$$K = \begin{pmatrix} k(x_1, x_1) & k(x_1, x_2) & \cdots & k(x_1, x_m) \\ k(x_2, x_1) & k(x_2, x_2) & \cdots & k(x_2, x_m) \\ \vdots & \vdots & \cdots & \vdots \\ k(x_m, x_1) & k(x_m, x_2) & \cdots & k(x_m, x_m) \end{pmatrix}$$

is positive semi-definite.

Mercer kernel

Definition

A symmetric mapping $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ such that

$$\int \int k(x, y) f(x) f(y) dx dy \geq 0$$

for all functions f s.t.

$$\int f^2(x) dx < \infty$$

is a **Mercer kernel**.

Two different pictures

These different definitions lead to different interpretations of what the kernel does:

Interpretation II

Mercer kernel map:

$$\mathcal{H}_M = \ell_2 = \left\{ x \mid \sum_i x_i^2 < \infty \right\}$$

$$\langle f, g \rangle_* = f^t g$$

$$\Phi : \mathcal{X} \rightarrow (\sqrt{\lambda_1} \phi_1(x), \sqrt{\lambda_2} \phi_2(x), \dots)^t$$

where λ_i, ϕ_i are the e-values and eigenfunctions of $k(x, y)$ with $\lambda_i > 0$.

where ℓ_2 is the space of vectors s.t. $\sum_i a_i^2 < \infty$.

Interpretation I: The dot-product picture

When a Gaussian kernel $k(x, x_i) = \exp(\|x - x_i\|^2 / \sigma)$ is used

- the point $x_i \in \mathcal{X}$ is mapped into the Gaussian $G(\cdot, x_i, \sigma I)$
- \mathcal{H}_k is the space of all functions that are linear combinations of Gaussians.
- the kernel is a dot product in \mathcal{H}_k and a non-linear similarity on \mathcal{X} .

The reproducing property

- With the definition of \mathcal{H}_k and $\langle \cdot, \cdot \rangle_*$ one has

$$\langle k(\cdot, x), f(\cdot) \rangle_* = f(x) \quad \forall f \in \mathcal{H}_k$$

- This is called the **reproducing property**.
- Leads to the **reproducing Kernel Hilbert Spaces**

Definition

A **Hilbert Space** is a complete dot-product space.

(vector space + dot product + limit points of all Cauchy sequences)

Reproducing kernel Hilbert spaces

Definition

Let \mathcal{H} be a Hilbert space of functions $f : \mathcal{X} \rightarrow \mathbb{R}$. \mathcal{H} is a **Reproducing Kernel Hilbert Space** (RKHS) with inner-product $\langle \cdot, \cdot \rangle_*$ if there exists a

$$k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$$

s.t.

- $k(\cdot, \cdot)$ spans \mathcal{H} that is

$$\mathcal{H} = \overline{\{f(\cdot) | f(\cdot) = \sum_i \alpha_i k(\cdot, x_i) \text{ for } \alpha_i \in \mathbb{R} \text{ and } x_i \in \mathcal{X}\}}$$

- $k(\cdot, \cdot)$ is a *reproducing kernel* of \mathcal{H}

$$f(x) = \langle f(\cdot), k(\cdot, x) \rangle_* \quad \forall f \in \mathcal{H}$$

Interpretation II: Mercer Kernels

Theorem

Let $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ be a Mercer kernel. Then there exists an orthonormal set of functions

$$\int \phi_i(x) \phi_j(x) dx = \delta_{ij}$$

and a set of $\lambda_i \geq 0$ such that

- $\sum_i \lambda_i^2 = \int \int k^2(x, y) dx dy < \infty$ **and**
- $k(x, y) = \sum_{i=1}^{\infty} \lambda_i \phi_i(x) \phi_i(y)$

Transformation induced by a Mercer kernel

This eigen-decomposition gives another way to design the feature transformation induced by the kernel $k(\cdot, \cdot)$.

- Let

$$\Phi : \mathcal{X} \rightarrow l_2$$

be defined by

$$\Phi(x) = \left(\sqrt{\lambda_1} \phi_1(x), \sqrt{\lambda_2} \phi_2(x), \dots \right)$$

where l_2 is the space of square summable sequences.

- Clearly

$$\begin{aligned} \langle \Phi(x), \Phi(y) \rangle &= \sum_{i=1}^{\infty} \sqrt{\lambda_i} \phi_i(x) \sqrt{\lambda_i} \phi_i(y) \\ &= \sum_{i=1}^{\infty} \lambda_i \phi_i(x) \phi_i(y) = k(x, y) \end{aligned}$$

Issues

Therefore there is a vector space l_2 other than \mathcal{H}_k such that $k(x, y)$ is a dot product in that space.

- Have two very different interpretations of what the kernel does
 - Reproducing kernel map
 - Mercer kernel map
- They are in fact more or less the same.

RKHS Vs Mercer maps

- For \mathcal{H}_M we write

$$\Phi(x) = \sum_i \sqrt{\lambda_i} \phi_i(x) \mathbf{e}_i$$

- As the ϕ_i 's are orthonormal there is a 1 – 1 map

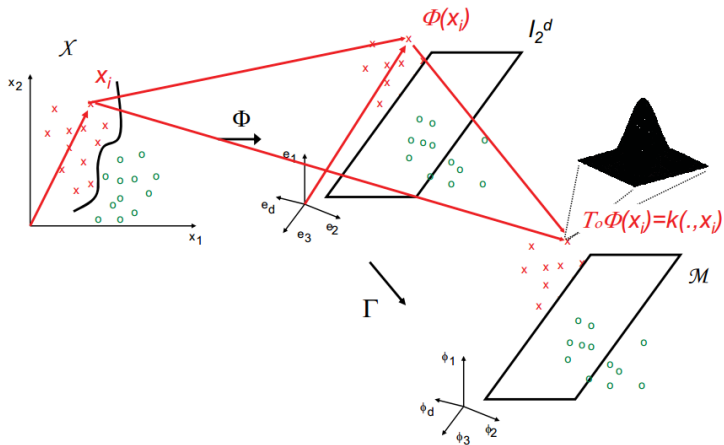
$$\Gamma : \ell_2 \rightarrow \text{span}\{\phi_k\} \quad \mathbf{e}_k = \sqrt{\lambda_k} \phi_k(\cdot)$$

- Can write

$$(\Gamma \circ \Phi)(x) = \sum_i \sqrt{\lambda_i} \phi_i(x) \phi_i(\cdot) = k(\cdot, x)$$

- Hence $k(\cdot, x)$ maps x into $\mathcal{M} = \text{span}\{\phi_k(\cdot)\}$

The Mercer picture



Mercer map

Define the inner-product in \mathcal{M} as

$$\langle f, g \rangle_m = \int f(x)g(x)dx$$

Note we will normalize the eigenfunctions ϕ_l such that

$$\int \phi_l(x)\phi_k(x)dx = \frac{\delta_{lk}}{\lambda_l}$$

Any function $f \in \mathcal{M}$ can be written as

$$f(x) = \sum_{k=1}^{\infty} \alpha_k \phi_k(x)$$

then

Mercer map

$$\begin{aligned}\langle f(\cdot), k(\cdot, y) \rangle_m &= \int f(x) k(x, y) dx \\&= \int \sum_{k=1}^{\infty} \alpha_k \phi_k(x) \sum_{l=1}^{\infty} \lambda_l \phi_l(x) \phi_l(y) dx \\&= \sum_{k=1}^{\infty} \sum_{l=1}^{\infty} \lambda_k \lambda_l \phi_l(y) \int \phi_k(x) \phi_l(x) dx \\&= \sum_{l=1}^{\infty} \lambda_l \lambda_l \phi_l(y) \frac{1}{\lambda_l} \\&= \sum_{l=1}^{\infty} \lambda_l \phi_l(y) = f(y)\end{aligned}$$

$\therefore k$ is a reproducing kernel on \mathcal{M} .

Mercer map Vs Reproducing kernel map

We want to check if

- the space $\mathcal{M} = \mathcal{H}_k$
- $\langle f, g \rangle_m$ and $\langle f, g \rangle_*$ are equivalent.

To do this will involve the following steps

- Show $\mathcal{H}_k \subset \mathcal{M}$.
- Show $\langle f, g \rangle_m = \langle f, g \rangle_*$ for $f, g \in \mathcal{H}_k$.
- Show $\mathcal{M} \subset \mathcal{H}_k$.

$$\mathcal{H}_k \subset \mathcal{M}$$

If $f \in \mathcal{H}_k$ then there exists $m \in \mathbb{N}$, $\{\alpha_i\}$ and $\{x_i\}$ such that

$$\begin{aligned} f(\cdot) &= \sum_{i=1}^m \alpha_i k(\cdot, x_i) \\ &= \sum_{i=1}^m \alpha_i \sum_{l=1}^{\infty} \lambda_l \phi_l(x_i) \phi_l(\cdot) \\ &= \sum_{l=1}^{\infty} \left(\sum_{i=1}^m \alpha_i \lambda_l \phi_l(x_i) \right) \phi_l(\cdot) \\ &= \sum_{l=1}^{\infty} \gamma_l \phi_l(\cdot) \end{aligned}$$

Thus f is a linear combination of the ϕ_i 's and $f \in \mathcal{M}$.

This shows that if $f \in \mathcal{H}$ then $f \in \mathcal{M}$ and therefore $\mathcal{H} \subset \mathcal{M}$.

Equivalence of the inner-products

Let $f, g \in \mathcal{H}$ with

$$f(\cdot) = \sum_{i=1}^n \alpha_i k(\cdot, x_i) \quad g(\cdot) = \sum_{j=1}^m \beta_j k(\cdot, y_j)$$

Then by definition

$$\langle f, g \rangle_* = \sum_{i=1}^n \sum_{j=1}^m \alpha_i \beta_j k(x_i, y_j)$$

While

$$\begin{aligned} \langle f, g \rangle_m &= \int f(x) g(x) dx \\ &= \int \sum_{i=1}^n \alpha_i k(x, x_i) \sum_{j=1}^m \beta_j k(x, y_j) dx \\ &= \sum_{i=1}^n \sum_{j=1}^m \alpha_i \beta_j \int k(x, x_i) k(x, y_j) dx \end{aligned}$$

Equivalence of the inner-products ctd

$$\begin{aligned}\langle f, g \rangle_m &= \sum_{i=1}^n \sum_{j=1}^m \alpha_i \beta_j \int \sum_{l=1}^{\infty} \lambda_l \phi_l(x) \phi_l(x_i) \sum_{s=1}^{\infty} \lambda_s \phi_s(x) \phi_s(y_j) dx \\ &= \sum_{i=1}^n \sum_{j=1}^m \alpha_i \beta_j \sum_{l=1}^{\infty} \lambda_l \phi_l(x_i) \phi_l(y_j) \\ &= \sum_{i=1}^n \sum_{j=1}^m \alpha_i \beta_j k(x_i, y_j) \\ &= \langle f, g \rangle_*$$

Thus for all $f, g \in \mathcal{H}$

$$\langle f, g \rangle_m = \langle f, g \rangle_*$$

$$\mathcal{M} \subset \mathcal{H}$$

- Can also show that if $f \in \mathcal{M}$ then also $f \in \mathcal{H}_k$.
- Will not prove that here.
- But it implies $\mathcal{M} \subset \mathcal{H}_k$

Summary

The **reproducing kernel map** and the **Mercer Kernel map** lead to the same RKHS, Mercer gives us an orthonormal basis.

Interpretation I

Reproducing kernel map:

$$\mathcal{H}_k = \left\{ f(\cdot) \mid f(\cdot) = \sum_{j=1}^m \alpha_j k(\cdot, x_j) \right\}$$

$$\langle f, g \rangle_* = \sum_{i=1}^m \sum_{j=1}^{m'} \alpha_i \beta_j k(x_i, x'_j)$$

$$\Phi_r : \mathcal{X} \rightarrow k(\cdot, x)$$

Summary

The **reproducing kernel map** and the **Mercer Kernel map** lead to the same RKHS, Mercer gives us an orthonormal basis.

Interpretation II

Mercer kernel map:

$$\mathcal{H}_M = \ell_2 = \left\{ x \mid \sum_i x_i^2 < \infty \right\}$$

$$\langle f, g \rangle_* = f^t g$$

$$\Phi_M : \mathcal{X} \rightarrow (\sqrt{\lambda_1} \phi_1(x), \sqrt{\lambda_2} \phi_2(x), \dots)^t$$

$$\Gamma : \ell_2 \rightarrow \text{span}\{\phi_k(\cdot)\}$$

$$\Gamma \circ \Phi_M = \Phi_r$$

Back to Regularization

We to solve

$$\min_{f \in \mathcal{H}_k} \left[\sum_{i=1}^n L(y_i, f(x_i)) + \lambda J(f) \right]$$

where \mathcal{H}_k is the RKHS of some appropriate Mercer kernel $k(\cdot, \cdot)$.

What is a good regularizer?

- **Intuition:** wigglier functions have larger norm than smoother functions.
- For $f \in \mathcal{H}_k$ we have

$$\begin{aligned} f(x) &= \sum_i \alpha_i k(x, x_i) \\ &= \sum_i \alpha_i \sum_l \lambda_l \phi_l(x) \phi_l(x_i) \\ &= \sum_l \left[\lambda_l \sum_i \alpha_i \phi_l(x_i) \right] \phi_l(x) \\ &= \sum_l c_l \phi_l(x) \end{aligned}$$

What is a good regularizer?

- and therefore

$$\|f(x)\|^2 = \sum_{lk} c_l c_k \langle \phi_l(x), \phi_k(x) \rangle_m = \sum_{lk} \frac{1}{\lambda_l} c_l c_k \delta_{lk} = \sum_l \frac{c_l^2}{\lambda_l}$$

with $c_l = \lambda_l \sum_i \alpha_i \phi_l(x_i)$.

- Hence
 - $\|f\|^2$ grows with the number of c_i different than zero.
 - functions with large e-values get penalized less and vice versa
 - more coefficients means more **high frequencies** or **less smoothness**.

Representer Theorem

Theorem

Let

- $\Omega : [0, \infty) \rightarrow \mathbb{R}$ be a strictly monotonically increasing function
- \mathcal{H} is the RKHS associated with a kernel $k(x, y)$
- $L(y, f(x))$ be a loss function

then

$$\hat{f} = \arg \min_{f \in \mathcal{H}_k} \left[\sum_{i=1}^n L(y_i, f(x_i)) + \lambda \Omega(\|f\|^2) \right]$$

has a representation of the form

$$\hat{f}(x) = \sum_{i=1}^n \alpha_i k(x, x_i)$$

Relevance

- The remarkable consequence of the theorem is that
 - Can reduce the minimization over the **infinite dimensional space of functions** to a minimization over a **finite dimensional space**.
- This is because as $\hat{f} = \sum_{i=1}^n \alpha_i k(\cdot, x_i)$ then

$$\begin{aligned}\|\hat{f}\|^2 &= \langle \hat{f}, \hat{f} \rangle = \sum_{ij} \alpha_i \alpha_j \langle k(\cdot, x_i), k(\cdot, x_j) \rangle \\ \sum_{ij} \alpha_i \alpha_j k(x_i, x_j) &= \alpha^T \mathbf{K} \alpha\end{aligned}$$

and

$$\hat{f}(x_i) = \sum_j \alpha_j k(x_i, x_j) = \mathbf{K}_i \alpha$$

where $\mathbf{K} = (k(x_i, x_j))$, Gram matrix, and \mathbf{K}_i is its i th row.

Representer Theorem

Theorem

Let

- $\Omega : [0, \infty) \rightarrow \mathbb{R}$ be a strictly monotonically increasing function
- \mathcal{H} is the RKHS associated with a kernel $k(x, y)$
- $L(y, f(x))$ be a loss function

then

$$\hat{f} = \arg \min_{f \in \mathcal{H}_k} \left[\sum_{i=1}^n L(y_i, f(x_i)) + \lambda \Omega(\|f\|^2) \right]$$

has a representation of the form

$$\hat{f}(x) = \sum_{i=1}^n \hat{\alpha}_i k(x, x_i)$$

where

$$\hat{\alpha} = \arg \min_{\alpha} \left[\sum_{i=1}^n L(y_i, \mathbf{K}_i \alpha) + \lambda \Omega(\alpha^t \mathbf{K} \alpha) \right]$$

Rejigging the formulation of the SVM

- When given linearly separable data $\{(x_i, y_i)\}$ the optimal separating hyperplane is given by

$$\min_{\beta_0, \beta} \|\beta\|^2 \quad \text{subject to} \quad y_i(\beta_0 + \beta^T x_i) \geq 1 \quad \forall i$$

- The constraints are fulfilled when

$$\max(0, 1 - y_i(\beta_0 + \beta^T x_i)) = (1 - y_i(\beta_0 + \beta^T x_i))_+ = 0 \quad \forall i$$

- Hence we can re-write the optimization problem as

$$\min_{\beta_0, \beta} \left[\sum_{i=1}^n (1 - y_i(\beta_0 + \beta^T x_i))_+ + \|\beta\|^2 \right]$$

SVM's connections to regularization

Finding the optimal separating hyperplane

$$\min_{\beta_0, \beta} \left[\sum_{i=1}^n (1 - y_i(\beta_0 + \beta^t x_i))_+ + \|\beta\|^2 \right]$$

can be seen as a regularization problem

$$\min_f \left[\sum_{i=1}^n L(y_i, f(x_i)) + \lambda \Omega(\|f\|^2) \right]$$

where

- $L(y, f(x)) = (1 - y f(x))_+$
- $\Omega(\|f\|^2) = \|f\|^2$

SVM's connections to regularization

- From the Representer theorem know the solution to the latter problem is

$$\hat{f}(x) = \sum_{i=1}^n \alpha_i x_i^T x$$

if the basic kernel $k(x, y) = x^T y$ is used.

- Therefore $\|f\|^2 = \alpha^T \mathbf{K} \alpha$
- This is the same form of the solution found via the KKT conditions

$$\hat{\beta} = \sum_{i=1}^n \alpha_i y_i x_i$$