Birla Institute of Technology and Science, Pilani

BITS F464: Machine Learning

2nd Semester 2019-20

Labsheet-04 : Basis function

Basis Functions: Introduction

We need flexible method for constructing a function f(t) that can track local curvature.

- We pick a system of K basis functions $\Psi_k(t)$, and call this the basis for f(t).
- We express f(t) as a weighted sum of these basis functions (also referred to as stacking of basis functions):

$$f(t) = a_1\Psi_1(t) + a_2\Psi_2(t) + \dots + a_K\Psi_K(t)$$

The coefficients $a_1, \dots, a_K$ determine the shape of the function.

Commonly used basis functions?

- Powers: 1, t, t^2, and so on. They are the basis functions for polynomials. These are not very flexible, and are used only for simple problems.
- Fourier series: 1, sin(ωt), cos(ωt), sin(2ωt), cos(2ωt), and so on for a fixed known frequency ω. These are used for periodic functions.
- Gaussian functions: Polynomials are global basis functions, each affecting the prediction over the whole input space. Often, local basis functions are more appropriate. One possibility is to use functions proportional to Gaussian probability densities.
- B-spline functions: These have now more or less replaced polynomials for non-periodic problems.

How do we construct basis systems?

- We start with a prototype basis function Ψ (t). Let's call it the mother function.
- We apply three operations to it:
- Lateral shift Ψ (t+a) to focus fitting power near a position a.
- Scale change Ψ (bt), to increase resolving power, or the capacity for local curvature.
- Smoothing, to increase differentiability and smoothness.

Example: Fourier Series

Consider the Fourier series:

• Mother functions: Ψ (t) = sin(ωt), and Ψ (t) = 1

• Lateral Shift: sin[ω(t+π/2)] = cos(ωt)

• Scale change: sin(kωt), cos(kωt), k=1,2,…

• Smoothing: None; these are already infinitely differentiable functions.

• Strengths: Orthogonality makes computing coefficients very fast

• Limitations: Fourier series is only natural if f(t) is periodic with period ω.

Five Fourier Basis Functions

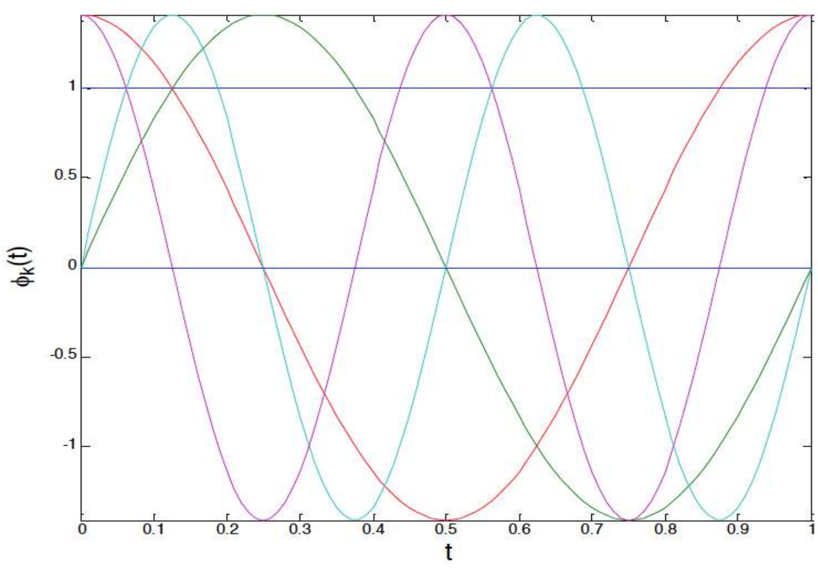
• Ψ1(t) = 1

• A sine/cosine pair with period 1:

Captures low frequency variation

• A sine/cosine pair with period ó:

Captures higher frequency Variation



How to work with Basis Functions in R?

Polynomial function

Install "basefun" package for polynomial basis function.

Syntax : polynomial_basis(var, coef, ui = NULL, ci = NULL)

Parameter description:

var a numeric_var object

coef a logical defining the order of the polynomial

ui a matrix defining constraints

ci a vector defining constraints

```
### set-up basis of order 3 ommiting the quadratic term

pb <- polynomial_basis(numeric_var("x", support = c(0, pi)), coef = c(TRUE, TRUE, FALSE, TRUE))

### generate data + coefficients

x <- as.data.frame(mkgrid(pb, n = 100))

cf <- c(1, 2, 0, 1.75)

### evaluate basis (in two equivalent ways)

pb(x[1:10,,drop = FALSE])

model.matrix(pb, data = x[1:10, ,drop = FALSE])

### evaluate and plot polynomial defined by,

### basis and coefficients

plot(x$x, predict(pb, newdata = x, coef = cf), type = "l")
```

Fourier basis function

The Fourier basis is a system that is usually used for periodic functions. It has the advantages of very fast computation and great flexibility. If the data are considered to be nonperiod, the Fourier basis is usually preferred. The first Fourier basis function is the constant function. The remainder are sine and cosine pairs with integer multiples of the base period. The number of basis functions generated is always odd. Install “fda” package for “create.fourier.basis”

Syntax: create.fourier.basis(rangeval=c(0, 1), nbasis=3, period=diff(rangeval), dropind=NULL, quadvals=NULL, values=NULL, basisvalues=NULL, names=NULL, axes=NULL)

```
#  using 3 basis functions

yearbasis3 <- create.fourier.basis(c(0,365), axes=list("axesIntervals"))

  #  plot the basis

plot(yearbasis3)
```

Gaussian Basis functions

Algebraically, Gaussian basis functions are defined as follows:

$$\phi_k(t; \mu_k, \sigma_k^2) = \exp\left(-\frac{||t-\mu_k||^2}{2\sigma_k^2}\right),\ k = 1, \ldots, K$$

where μ_k is a parameter determining the center of the basis function, σ_k^2 is a parameter that determines the width and ||.|| is the Euclidian norm. The basis functions overlap with each other to capture the information about t, and the width parameter play an essential role to capture the structure in the data over the region of input data. The parameters featuring in each basis function are often determined heuristically based on the structure of the observed data.

Radial basis function

A radial basis function (RBF) is a real-valued function Ψ whose value depends only on the distance between the input and some fixed point, either the origin, so that Ψ (x) = Ψ (||x||), or some other fixed point c, called a center, so that Ψ (x) = Ψ (||x-c||). Any function Ψ that satisfies the property Ψ (x) = Ψ (||x||), is a radial function.

Let's use a one-dimensional dataset as an illustration of the gaussian influence:

```
rbf.gauss <- function(gamma=1.0)
 {
  function(x) {
    exp(-gamma * norm(as.matrix(x),"F")^2 )
  }
}
D <- matrix(c(-3,1,4), ncol=1) # 3 datapoints
N <- length(D[,1])
xlim  <- c(-5,7)
plot(NULL,xlim=xlim,ylim=c(0,1.25),type="n")
points(D,rep(0,length(D)),col=1:N,pch=19)
x.coord = seq(-7,7,length=250)
gamma <- 1.5
for (i in 1:N) {
  points(x.coord, lapply(x.coord - D[i,], rbf.gauss(gamma)), type="l", col=i)
}
```

The value of gamma controls how far or how little the influnce of each datapoint is felt:

```
plot(NULL,xlim=xlim,ylim=c(0,1.25),type="n")

points(D,rep(0,length(D)),col=1:N,pch=19)

x.coord = seq(-7,7,length=250)

gamma <- 0.25

for (i in 1:N) {

  points(x.coord, lapply(x.coord - D[i,], rbf.gauss(gamma)), type="l", col=i)

}
```

Exercise:

- Read about B spline and compare them qualitatively and quantitatively with polynomial basis function.