Jared Yu

Computational Statistics

Problem Set 10

*NOTE: The code for the assignment is in the Appendix.*

1. Show that the Fourier Trigonometric Family

is an orthogonal family over with respect to the weight function .

The Fourier Trigonometric Family, denoted , is orthogonal over w.r.t. i.f.f.

To show this it will first be noted that there are three possible cases of based combinations of the sine and cosine functions within . Next, it will be noted that the first element of , , can be represented with for since . Then all the pairwise combinations of members of can be denoted as follows:

Below, each of these equations and their cases will be shown:

Equation

Case 1: ,

Case 2: ;

Equation

Case 1:

Case 2: ;

Case 3: ;

Equation

Case 1:

In the above 3 equations, the property of holds and so it could be said that the Fourier Trigonometric Family is orthogonal.

1. (a) Show that if is a set of orthogonal functions, then it is a linearly independent set.

A set is linearly independent if

only when .

To show that if is a set of orthogonal function, then it is a linearly independent set, suppose then that isn’t linearly independent. Then for some ,

However, this goes against the rule that for orthogonal functions, when . Therefore, in this case can’t be a set of orthogonal functions. Having completed a proof by contrapositive, it’s possible to infer that the original statement that if is orthogonal then it must also be linearly independent is true.

(b) Prove that the integrated mean squared error is the sum of the integrated variance and the integrated squared bias, that is,

The part in the center within the integral can be written as follows:

1. Let be a set of orthogonal functions. Show that

What is the common value of the expression above if the are orthonormal? In these expressions, represents an norm. Would a similar equation hold for a general norm?

Given that are orthonormal, then . Therefore, the value would be . If there was an alternate norm such as , there’s no guarantee that the above situation would hold. The reason is that the inner product cancelling out only works for the product of two different and functions. If there’s the product of for example, the solution is not certain since there’s no rule saying whether or not they cancel out properly the way that two orthogonal functions do.

1. (a) Use the Gram-Schmidt orthogonalization process as described in Lecture 10C to derive the first four orthonormal Chebychev polynomials. Note that the range is and the weight function is . (Please note that the set is to be orthonormal and not just orthogonal.)

Let , denote the orthogonalized set of functions. Also, let the linearly independent set , be and

Then from the Gram-Schmidt orthogonalization process these are the first four polynomials.

The above derivations were calculated from the following:

The above inner products were calculated using an online calculator: [www.integral-calculator.com](http://www.integral-calculator.com) Permission was given by the professor to use this tool as long as credit is given. The reason is that the above calculations require the use of trigonometric identities and so the entire process would be highly time consuming to complete by hand.

Now that the orthogonal set , have been calculated, they need to be normalized to create the orthonormal set. This calculation is performed by dividing by the magnitude, Denote the orthonormal set to be , .

NOTE: The same online integral calculator was used for the above equations.

(b) Posted on the course Blackboard is the data set Orthogonal.txt containing 1000 observations from a distribution (standard deviation of ). Use the polynomials derived in part (a) to estimate the density . Do you think that this is a good estimate of ? Be sure to explain why or why not.

In this problem, to approximate it was necessary to find the values of in

These values were approximated using,

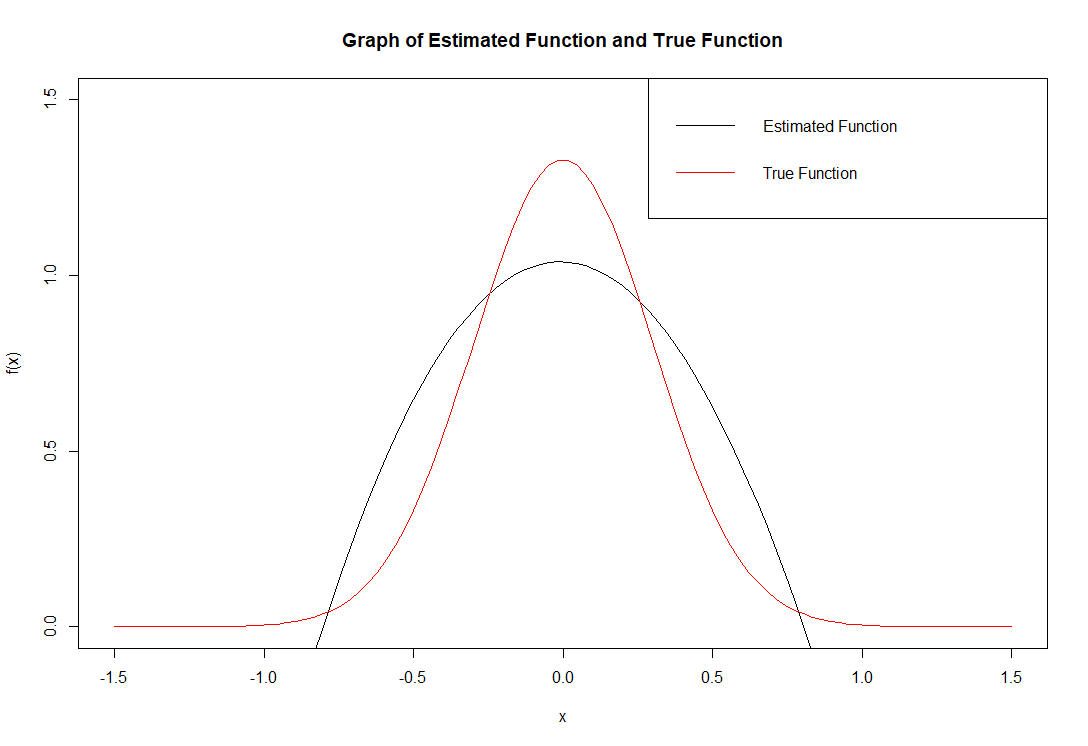
Here, since the data comes from a normal density without any other function attached to it. Using R, below are the estimated values of

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

Below in Figure 1 is a plot of the estimated function in black and the true function in red. The true function was plot by using the dnorm function in R with the corresponding parameters of and for the mean and standard deviation. It makes sense to think that it wouldn’t estimate perfectly since there are only 4 orthonormal polynomials compared to the theoretical limit of infinity.

Looking at the graph, it seems that the estimated function does have part of the characteristic bell curve appearance, without rounding off to the sides. Instead, the values continue towards around -15. This is understandable since this is an estimated function based off some sample data. Therefore, it doesn’t necessarily follow that it would perfectly fit the shape of a probability density function that has a bell shape going from to which are based off expressing extreme outliers that are unseen in the data.

Also, the estimated function does happen to cover most of the area under the true function. However, it’s slightly shorter in the center and higher on the sides, giving it properties that would make it seem that it has something like having heavier tails. Overall, it’s a decent estimate of the true function , however, with more orthonormal polynomials and a larger dataset it would make sense to think that the estimate would improve.



Figure

1. (a) Which properties of a natural cubic spline does the following function possess and which does it not possess?

A natural cubic spline has the following properties:

Additionally, the cubic spline is a polynomial of degree and is continuous.

The above function is a polynomial of less than or equal to 3.

Also, the only break point in the function is at , and the limits of the function as approaches from left and right are both . Therefore, it’s a continuous function. So, the first property holds.

Let and . Then the following are the set of first and second derivatives for and ,

From here, it’s clear to see that and don’t hold, because and . So here the second and third property don’t hold.

It was mentioned too that the function is a polynomial with degree . To be specific, and are both polynomials of degree . Therefore, and are linear functions (another property of cubic splines). However, doesn’t exist since the second derivatives are different. Then it follows that and is irrelevant.

(b) Find a natural cubic spline function whose knots are and that takes values , , , and .

Then the natural cubic spline is as follows:

1. (Extra Credit) Prove that the Fourier coefficients form the finite expansion in basis functions with the minimum mean squared error, that is prove

where are the Fourier coefficients and are any other constants. Assume that the basis set is orthonormal and note that is the norm. Also, you may find it useful to use the results of Problem 3. (Hint: Write a function of , , differentiate, set to zero for the minimum, and determine . This same approach can be done in multidimensions for or else induction can be used from on.)

Therefore, the value of for that minimizes is . This proves the statement that .

Appendix

### Q4 part(b)  
orthogonal <- read.csv('Orthogonal.txt', header = FALSE)  
orthogonal <- orthogonal[[1]]  
hist(orthogonal)  
plot(1:length(orthogonal), orthogonal)  
  
q0 <- function(x) {  
 (pi / 2)^(-1/2)  
}  
  
q1 <- function(x) {  
 x \* ((pi / 8)^(-1 / 2))  
}  
  
q2 <- function(x) {  
 ((x^2) - (1 / 4)) / (sqrt(pi / 32))  
}  
  
q3 <- function(x) {  
 ((x^3) - (x / 2)) / sqrt(pi / 128)  
}  
  
c0 <- mean(q0(orthogonal))  
c1 <- mean(q1(orthogonal))  
c2 <- mean(q2(orthogonal))  
c3 <- mean(q3(orthogonal))  
  
f <- function(x) {  
 # c0 \* q0(x) + c1 \* q1(x) + c2 \* q2(x)  
 c0 \* q0(x) + c1 \* q1(x) + c2 \* q2(x) + c3 \* q3(x)  
}  
  
xs <- seq(-1.5, 1.5, length.out = 1e2)  
plot(xs, f(xs), type = 'l', ylim = c(0, 1.5),  
 main = 'Graph of Estimated Function and True Function',  
 xlab = 'x', ylab = 'f(x)')  
lines(xs, dnorm(xs, mean = 0, sd = 0.3), col = 'red')  
legend("topright", legend = c('Estimated Function', 'True Function'),  
 col = c('black', 'red'), lty = c(1,1))  
  
### Q5  
# part (a)  
f <- function(x) {  
 if ((x >= -1) & (x <= 0)) {  
 (x + 1) + (x + 1)^3  
 } else if ((x > 0) & (x <= 1)) {  
 4 + (x - 1) + (x - 1)^3  
 } else {  
 0  
 }  
}  
  
xs <- seq(-1.1, 1.1, length.out = 1e4)  
f\_vec <- Vectorize(f)  
plot(xs, f\_vec(xs), type = 'l')  
  
# part (b)  
s0 <- function(x) {  
 (1 / 12) \* ((x - 1)^3) + (5 / 12) \* (x - 1) + (2 - x)  
}  
  
s1 <- function(x) {  
 (1 /12) \* ((3 - x)^3) + (1 / 3) \* (x - 2) + (5 / 12) \* (3 - x)  
}  
  
s2 <- function(x) {  
 (1 / 4) \* (x - 3) + (1 / 3) \* (4 - x)  
}  
  
s0(1)  
s0(2) == s1(2)  
s1(3) == s2(3)  
s2(4)  
  
b = 0.01  
x = c(seq(1,2,b), seq(2,3,b), seq(3,4,b))  
y = c(s0(seq(1,2,b)), s1(seq(2,3,b)), s2(seq(3,4,b)))  
plot(x,y, type = 'l')  
abline(v = 1, col='red')  
abline(v = 2, col='red')  
abline(v = 3, col='red')  
abline(v = 4, col='red')