Biostatistics Department Technical Report

BST2017-001

Calculation of Orthogonal Polynomials and Their Derivatives

Charles R. Katholi, PhD

May 2017

Department of Biostatistics
School of Public Health
University of Alabama at Birmingham
ckatholi@uab.edu

Calculation of Orthogonal Polynomials and Their Derivatives

Charles R. Katholi, PhD

May 27, 2017

1 Introduction

The purpose of this report is to explore the calculation of orthogonal polynomials and their derivatives. The basic method follows the approach given by Emerson (1968)[2]. Given a set of points $x_1, \cdots, x_n$, polynomials $p_j(x_i), j = 0, \cdots, m$ where $p_j(x_i)$ is of degree j, are found such that the matrix of values

$$P = \begin{bmatrix} p_0(x_1) & p_1(x_1) & p2(x_1) & \cdots & p_m(x_1) \\ p_0(x_2) & p_1(x_2) & p2(x_2) & \cdots & p_m(x_2) \\ p_0(x_3) & p_1(x_3) & p2(x_3) & \cdots & p_m(x_3) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ p_0(x_n) & p_1(x_n) & p2(x_n) & \cdots & p_m(x_n) \end{bmatrix} \tag{1.1}$$

is orthonormal; that is such that $P^T P = I_{m+1}$ where I_{m+1} is the $(m+1) \times (m+1)$ identity matrix. From this matrix of values recursion coefficients A_j, B_j and C_j, $j = 1, \cdots, m$ are found and used to calculate values of the polynomials at any point x. It will be shown that the derivatives of the polynomials can also be found recursively utilizing constants A_j, B_j and C_j defined in the next section.

2 Derivations of the methods

2.1 Calculation of the matrix P of equation (1.1)

We shall utilize the notation of Emerson's paper [2]. Thus, let $x_i, i = 1, \cdots, n$ be given values of x and let w_i be a weight associated with each x_i. We shall find the A_j, B_j and C_j such that at any x the values of the orthogonal polynomials are given by the simple recursion (Equation (6) in Emerson),

$$p_j(x) = (A_j x + B_j)p_{j-1}(x) - C_j p_{j-2}(x)\,,\; j = 2, 3, \cdots, m < n \tag{2.1}$$

where $p_{-1}(x) = 0, \forall x$ and $p_0(x) = \left(\sqrt{\sum_{i=1}^{n} w_i}\right)^{-1}, \forall x$. In the derivation of these recursion constants we make use of the following conditions

$$\sum_{i=1}^{n} w_i p_j(x_i) p_k(x_i) = \begin{cases} 1 & , \text{ if } j = k \\ 0 & , \text{ if } j \neq k \end{cases} \tag{2.2}$$

The values of A_j, B_j and C_j are then found recursively from the following equations

$$\begin{aligned} A_j &= \left\{ \sum_{i=1}^{n} w_i x_i^2 p_{j-1}^2(x_i) - \left[\sum_{i=1}^{n} w_i x_i p_{j-1}^2(x_i) \right]^2 - \left[\sum_{i=1}^{n} w_i x_i p_{j-1}(x_i) p_{j-2}(x_i) \right]^2 \right\}^{-1/2} \\ B_j &= -A_j \sum_{i=1}^{n} w_i x_i p_{j-1}^2(x_i) \\ C_j &= A_j \sum_{i=1}^{n} w_i x_i p_{j-1}(x_i) p_{j-2}(x_i) \end{aligned} \tag{2.3}$$

The steps in the calculation of the A_j, B_j and C_j given w_i and $x_i, i = 1, \cdots, n$ can be summarized in the following steps

1. For the weights, w_i calculate $s_w = \left[\sum_{i=1}^{n} w(i)\right]^{-1/2}$. For $i = 1, \cdots, n$ set $p_{-1}(x_i) = 0$ and $p_0(x_i) = s_w$.

2. For $j = 1, 2, \cdots, n$

(i) Calculate

$$s_1 = \sum_{i=1}^{n} w_i x_i^2 p_{j-1}^2(x_i)$$
$$s_2 = \sum_{i=1}^{n} w_i x_i p_{j-1}(x_i)$$
$$s_3 = \sum_{i=1}^{n} w_i x_i p_{j-1}(x_i) p_{j-2}(x_i)$$

(ii) Calculate

$$A_j = \left\{ s_1 - s_2^2 - s_3^2 \right\}^{-1/2}$$
$$B_j = -A_j s_2$$
$$C_j = A_j s_3$$

(iii) For $i = 1, 2, \cdots, n$ calculate

$$p_j(x_i) = (A_j x_i + B_j) p_{j-1}(x_i) - C_j p_{j-2}(x_i)$$

3. End of loop started at step 2.

2.2 Approximating a function using the orthogonal polynomials

At this point the $n \times (m+1)$ orthogonal matrix P has been calculated. Next given observations of a function $y = f(x)$ for $x = x_1, x_2, \cdots, x_n$ we determine coefficients α_j, $j = 0, \cdots, m$ such that $f(x)$ is approximated on the range of the x_i by $\hat{f}(x) \approx \sum_{j=0}^{m} \alpha_j p_j(x)$. This is accomplished by finding the least squares solution of the $n \times (m+1)$ system of linear equations

$$P\boldsymbol{\alpha} = \boldsymbol{y}\,,\ \boldsymbol{y} = (y_1, y_2, \cdots, y_n)^T$$

Because of the orthogonality of P the solution is trivially found to be $\boldsymbol{\alpha} = P^T \boldsymbol{y}$. Noting that C_1 is arbitrary and so can be set to zero, the approximation of the function $f(x)$ can be found for any x by setting $p_0(x) = s_w$ and using the recursion of equation(2.1) to calculate $p_1(x), \cdots, p_m(x)$ and then form the linear combination $\hat{f(x)} = \sum_{j=0}^{m} \alpha_j p_j(x)$.

2.3 The derivative of the model function

If we wish to use the model to estimate the derivative of the approximating polynomial function we require the derivatives $p_j^{'}(x)$, $j = 0, 1, \cdots, m$. Noting that $p_0(x)$ is a constant independent of x we see that $p_0^{'}(x) = 0$ for all x and that $C_1 = 0$ we can find $p_j^{'}(x)$ by differentiating equation (2.1) to obtain the recursion

$$p_j^{'}(x) = A_j p_{j-1}(x) + (A_j x + B_j) p_{j-1}^{'}(x) - C_j p_{j-2}^{'}(x) \, , \; j = 1, 2, \cdots, m \quad (2.4)$$

Since the recurrence of equation(2.4) requires $p_{j-1}(x)$ it is necessary for any j to first utilize the recursion of equation (2.1) and then the recursion of equation (2.4). The justification for equation (2.4) is given in Appendix A.

3 Some examples

3.1 Approximating the Runge Function:

A well known example from elementary Numerical Analysis is the interpolation of the Runge Function

$$f(x) = \frac{1}{1 + 25x^2} \, , \; x \in [-1, 1] \quad (3.1)$$

Standard divided difference interpolation based on equally spaced points fails completely due to rapid oscillation of the polynomial approximation near -1 and 1.[1],[3] The problem is due to the equal spacing of the interpolation data points and becomes worse as the degree of the polynomial increases. The problem can be remedied by making the interpolation table based on the Chebyshev points on the interval $[a, b]$.

$$x_j = \left(a + b - (a - b) \frac{\cos[(2j - 1)\pi]}{2n} \right) /2 \quad (3.2)$$

In this case $a = -1$ and $b = 1$. Rather than approach this problem by means of approximation by an interpolating polynomial we shall use the methods of least squares for approximation by a set of orthogonal polynomials based on function evaluations at the Chebyshev points. We consider two approximations based on $n = 51$ Chebyshev points on $[-1, 1]$ and polynomials of degree 10 and 20 respectively. Noting that the Runge function is symmetric about

$x = 0$ we expect that only the polynomials of even degree will contribute to the approximation and that proves to be true. Figures 1 and 2 below illustrates the fit based on polynomials up to degree 10 and 20 respectively. The ANOVA table associated with the approximation based on $n = 51$ points and polynomials up to degree 10 is

Source	df	SS	MS	F
Regression	11	5.13254	0.46659	307.155
Error	40	0.06763	0.00159	
Total	51	5.19331		
$R^2 = 0.9883$				

Table 1: Analysis of variance summary for approximating the Runge function by a set of orthogonal polynomials up to degree 10 for x values taken to be the Chebyshev points on $[-1, 1]$ based on $n = 51$ values.The values of the function are not assumed to be subject to any additive errors beyond normal computational rounding errors.

Similarly, taking $n = 51$ and the maximum degree polynomial as $m = 20$ leads to Table 2

Source	df	SS	MS	F
Regression	21	5.19216	0.2474590	6491.4
Error	30	0.00114	0.0000381	
Total	51	5.19331		
$R^2 = 0.9998$				

Table 2: Analysis of variance summary for approximating the Runge function by a set of orthogonal polynomials up to degree 20 for x values taken to be the Chebyshev points on $[-1, 1]$ based on $n = 51$ values.The values of the function are not assumed to be subject to any additive errors beyond normal computational rounding errors.

In Figure 1 we note that the polynomial model of degree 10 fails to capture the peak of the function at $x = 0$ and shows significant oscillation in the tails of the function. These differences are made clear in the second figure that plots the derivatives. The large and rapid changes of the derivative of the polynomial model near -1 and 1 clarify the extent of the poor fit. On the other hand, the 20^{th} degree polynomial approximation shown in Figure 2, is a great improvement. The oscillation in the tails is greatly reduced and the fit at $x = 0$ is much better. Again, the derivatives display the problems in the tail of the function. This example deals with a very hard problem and the results of the approximation by the polynomials with maximum degree 20 does much to tame the bad behavior.

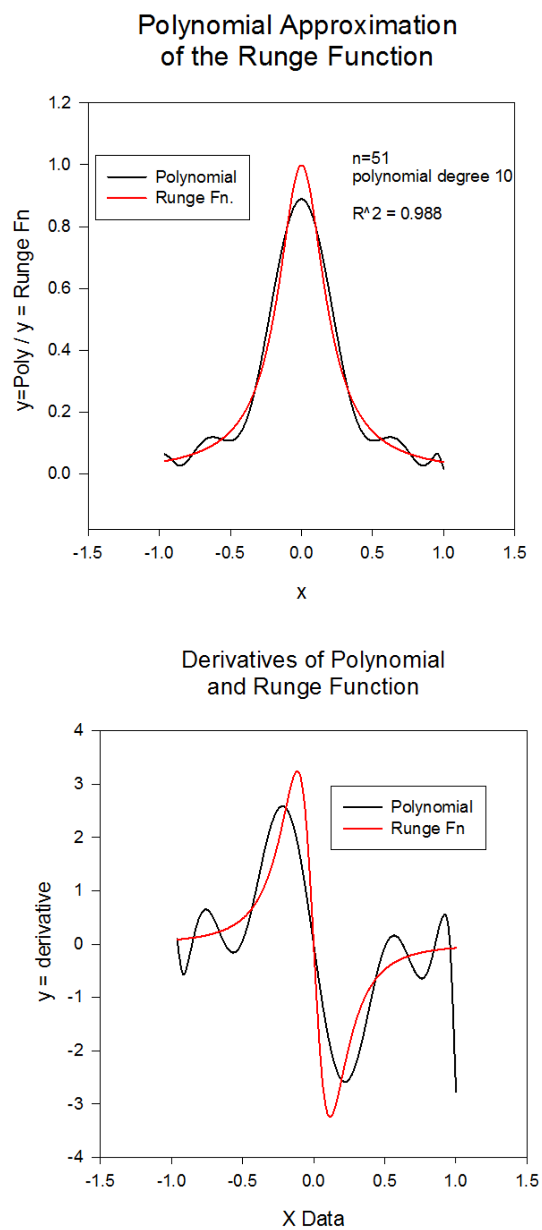

Figure 1: Approximation of the Runge function, $f(x) = (1 + 25x^2)^{-1}$ by orthogonal polynomials of maximum degree 10 by the method of Least squares.

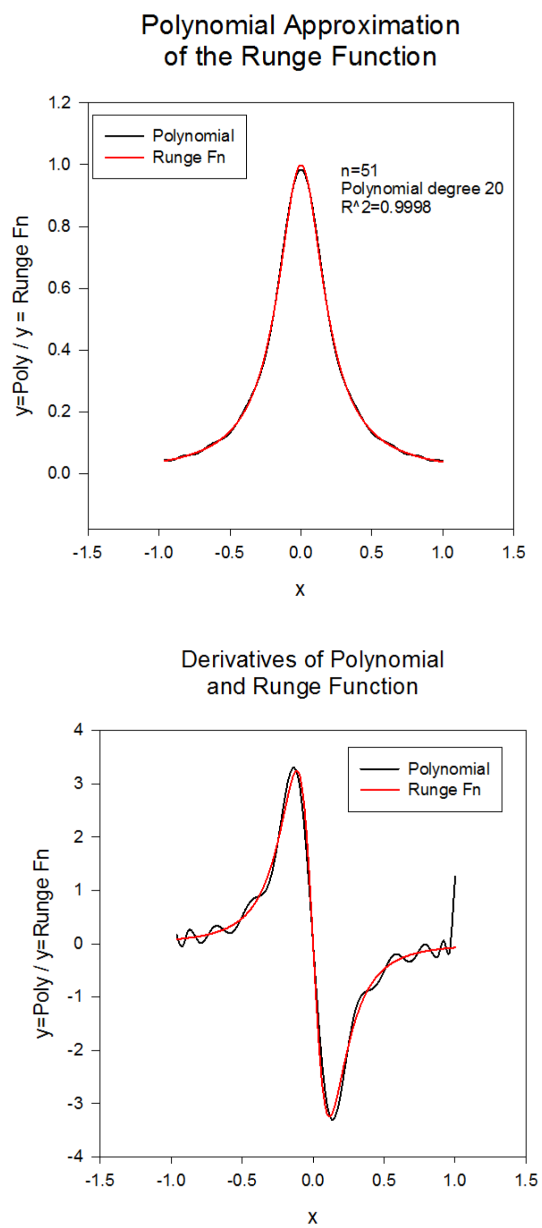

Figure 2: Approximation of the Runge function, $f(x) = (1 + 25x^2)^{-1}$ by orthogonal polynomials of maximum degree 20 by the method of Least squares.

3.2 A true polynomial data set

For this example we choose the polynomial

$$f(x) = 1 - x + x^2 + 3x^3$$

and generate pseudo data with the following SAS code,

```
seed=12345;
call streaminit(seed);
do i=1 to 30;
j=i-1;
x=mod(j,5) - 0.25+0.5*rand('uniform');
y=1-x+x**2+3x**3+0.5*rand('normal');
output;
end;
run;
```

Note that in this example, the values of X have been subjected to a random element. This is a device for generating values that are different but more or less the same. The y values are are generated subject to an error that is distributed $N(0, 0.25)$. We would expect to be able to capture the third degree polynomial nearly exactly and this precisely what we find. The results of the regress are summarized in the following Analysis of Variance table,

Source	df	SS	MS	F
Regression	5	1328517.6	265703.52	1434650.0
Error	25	4.6301195	0.1852044	
Total	30	1328522.3		
$R^2 = 0.9999997$				

Table 3: Analysis of Variance summary for fitting a simple cubic polynomial using orthogonal polynomials up to and including degree four. Although not shown, the coefficient for the fourth degree polynomial is small relative to the other coefficients and is not significant. In this case, the values of the observations are subject to an additive error distributed as $N(0, 0.25)$.

For this example the plots of the fit and of the derivative are so close to the true values as to be indistinguishable.

4 Summary

We have shown how to calculate the derivatives of a set of orthogonal polynomials given the constants for the three term recursion which generates the polynomial values. We have seen that these same constants can be used in a recursion to calculate the derivatives of the polynomials and hence of any model function based on a linear combination of the polynomials. A down side of this approach for SAS users is that the procedure ORPOL does not return the recurrence coefficients that are calculated by the methods described in this report [2]. The computations for this report were done using a program written in FORTRAN2003. A subroutine called ORPOLY.F95 is available from the author.

Appendices

A Derivation of equation(2.4)

Note that in the construction of the recursion given in equation(2.1), the constants A_j, B_j and C_j, $j = 1, 2, \cdots, m$ are found recursively and once they are found, the value of the polynomials at any point x can be found using equation(2.1),

$$p_j(x) = (A_j x + B_j)p_{j-1}(x) - C_j p_{j-2}(x)\,,\ j = 1, 2, 3, \cdots,,\ m < n$$

In particular, given any x we can use the recursion to find values of the polynomials at the points $x+h$ where h is a small number, the by elementary calculus we know that

$$p_j^{'}(x) = \lim_{h\to 0}\frac{p_j(x+h) - p_j(x)}{h}$$

and from application of the Taylor expansion we can write,

$$p_j(x+h) = p_j(x) + hp_j^{'}(x) + \circ(h) \tag{A.1}$$

From equation(2.1) we have that

$$p_j(x+h) = (A_j(x+h) + B_j)p_{j-1}(x+h) - C_j p_{j-2}(x+h)$$

if we now replace $p_k(x+h)$ for $k = j, j-1, j-2$ with the appropriate expressions from equation(A.1) in equation(2.1) , subtracting equation(2.1) leads (after considerable algebra) to

$$\frac{p_j(x+h) - p_j(x)}{h} = (A_j x + b_j)p^{'}_{j-1}(x) - C_j p^{'}_{j-2}(x) + A_j p_{j-1} \\ + hA_j p^{'}_{j-1} + \circ(h) \tag{A.2}$$

Taking the limit of both sides of this equation leads to the result given in equation(2.4).

References

[1] de Boor, Carl (2001),*A Proactial Guide to Splines: revised edition*, Springer, New York.

[2] Emerson, Phillip L.,(1968) Numerical Construction of Orthogonal Polynomials From a General Recurrence Formula, *Biometrics*, Vol 24, pp 695-701.

[3] Forsythe, G. ,Malcolm, M. and Moler, C.,*Computer Methods for Mathematical Computations*, Prentice-Hall series in automaticd computation, Prentice Hall, 1977.