Rational Function Interpolation
Barycentric Rational Interpolation
Coefficients of Interpolating Polynomial

Christian Zelenka
Michael Phipps
02/06/13
Summary of Previous Interpolation Methods

Linear Interpolation
Polynomial Interpolation
Cubic Spline Interpolation
Linear Interpolation

- Concatenation of linear interpolants between a pair of data points
- Piecewise linear function
- Cheap . . . $O(N \log(N))$
- Connect data points in a table (e.g. given population in 1990 and 2000, what was population in 1995?)
- Historically used with astronomical data
Polynomial Interpolation

- Given a data set, find a polynomial that goes exactly through each point
- Neville's algorithm: $O(N^2)$
- Basis for algorithms solving numerical ODEs and numerical integration
- Unstable on equidistant grid
Cubic Spline

- Interpolation interval divided into subintervals and each subinterval interpolated using 3rd degree polynomial
- Piecewise cubic functions with continuous first and second derivatives
- Requires function continuity and passing through all data points
- $O(N)$ complexity . . . stable and simplistic calculation
Cubic Spline Continued

- Known boundary first derivatives $\ldots O(h^4)$
- Natural spline (2nd derivatives $= 0$) $\ldots O(h^2)$
- Preferable to polynomials because the interpolation error can be made small even when using low degree polynomials
- Avoid Runge's Phenomenon in which boundaries oscillate wildly for functions like

$$f(x) = \frac{1}{1 + 25x^2}.$$
Why Rational Function Interpolation?

- Express more diverse behavior than polynomials
- Solves disadvantages of polynomial interpolation but polynomial can be found at any point; rational function cannot
- Ability to model equations with poles (while polynomials fail) ... Good if modeling a function with poles. Bad if goal is numerical stability.
- Higher orders give higher accuracy
Example of Unstable Rational Interpolation

Numerator and Denominator of 2nd degree

Pole at $x = -0.5$

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>-2</td>
<td>1</td>
</tr>
<tr>
<td>-1</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>-1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>
Rational Functions Continued

- Historically, the rational interpolant was constructed by solving a set of equations with unknown coefficients.
  - However, the larger the data set the larger the error in calculating coefficients.

- Neville's algorithm solves this problem by setting degree of numerator and denominator equal to N/2.

- Main disadvantage: no mechanism to find poles.
Why Barycentric Rational Interpolation?

- Suppresses all nearby poles
- Experimentation with higher orders encouraged
- Favorable comparison to splines but with smaller error and infinitely smooth curves
- If spacing of points is $O(h)$, error is $O(h^{d+1})$ as $h \to 0$. The complexity is order $O(Nd)$
Runge’s example with barycentric rational interpolation

Figure 1: Interpolating Runge’s example with \( d = 3 \) and \( n = 10, 20, 40, 80. \)
Sin(x) and errors for Sin(x) and Runge's Eq

Figure 2: Interpolating the sine function with \( d = 4 \) and \( n = 10, 20, 40, 80 \).

<table>
<thead>
<tr>
<th>( n )</th>
<th>Runge, ( d = 3 )</th>
<th>order</th>
<th>sine, ( d = 4 )</th>
<th>order</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>6.9e-02</td>
<td></td>
<td>1.7e-02</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>2.8e-03</td>
<td>4.6</td>
<td>3.9e-04</td>
<td>5.5</td>
</tr>
<tr>
<td>40</td>
<td>4.3e-06</td>
<td>9.4</td>
<td>7.1e-06</td>
<td>5.8</td>
</tr>
<tr>
<td>80</td>
<td>5.1e-08</td>
<td>6.4</td>
<td>1.3e-07</td>
<td>5.7</td>
</tr>
<tr>
<td>160</td>
<td>3.0e-09</td>
<td>4.1</td>
<td>2.7e-09</td>
<td>5.6</td>
</tr>
<tr>
<td>320</td>
<td>1.8e-10</td>
<td>4.0</td>
<td>6.0e-11</td>
<td>5.5</td>
</tr>
<tr>
<td>640</td>
<td>1.1e-11</td>
<td>4.0</td>
<td>1.5e-12</td>
<td>5.3</td>
</tr>
</tbody>
</table>

Table 1: Error in rational interpolant.
Optimal Order for $y = \text{Abs}(x)$

Figure 3: Interpolating $|x|$ over $[-5, 5]$ with $d = 3$ and $n = 21$.

<table>
<thead>
<tr>
<th>$n$</th>
<th>best $d$ value</th>
<th>error</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>$d = 0$</td>
<td>3.6e-02</td>
</tr>
<tr>
<td>20</td>
<td>$d = 1$</td>
<td>1.5e-03</td>
</tr>
<tr>
<td>40</td>
<td>$d = 3$</td>
<td>4.3e-06</td>
</tr>
<tr>
<td>80</td>
<td>$d = 7$</td>
<td>2.0e-10</td>
</tr>
<tr>
<td>160</td>
<td>$d = 10$</td>
<td>1.3e-15</td>
</tr>
</tbody>
</table>

Table 2: Error in Runge’s example, varying $d$.  


Errors in barycentric vs cubic spline interpolation for Runge's equation and the sine function

<table>
<thead>
<tr>
<th></th>
<th>rational, ( d = 3 )</th>
<th>cubic spline</th>
<th></th>
<th>rational, ( d = 3 )</th>
<th>cubic spline</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>6.9e-02</td>
<td>2.2e-02</td>
<td>10</td>
<td>1.3e-02</td>
<td>3.3e-03</td>
</tr>
<tr>
<td>20</td>
<td>2.8e-03</td>
<td>3.2e-03</td>
<td>20</td>
<td>1.2e-03</td>
<td>1.7e-04</td>
</tr>
<tr>
<td>40</td>
<td>4.3e-06</td>
<td>2.8e-04</td>
<td>40</td>
<td>8.4e-05</td>
<td>1.0e-05</td>
</tr>
<tr>
<td>80</td>
<td>5.1e-08</td>
<td>1.6e-05</td>
<td>80</td>
<td>5.4e-06</td>
<td>6.4e-07</td>
</tr>
<tr>
<td>160</td>
<td>3.0e-09</td>
<td>9.5e-07</td>
<td>160</td>
<td>3.4e-07</td>
<td>4.0e-08</td>
</tr>
<tr>
<td>320</td>
<td>1.8e-10</td>
<td>5.9e-08</td>
<td>320</td>
<td>2.1e-08</td>
<td>2.5e-09</td>
</tr>
<tr>
<td>640</td>
<td>1.1e-11</td>
<td>3.7e-09</td>
<td>640</td>
<td>1.3e-09</td>
<td>1.6e-10</td>
</tr>
</tbody>
</table>

Table 3: Error in rational and spline interpolation of Runge's (left) and the sine function (right)
Rational Function Algorithm

\[
R_{i(i+1)\ldots(i+m)} = \frac{P_\mu(x)}{Q_\nu(x)} = \frac{p_0 + p_1 x + \cdots + p_\mu x^\mu}{q_0 + q_1 x + \cdots + q_\nu x^\nu}
\]  (3.4.1)

Since there are \( \mu + \nu + 1 \) unknown \( p \)'s and \( q \)'s (\( q_0 \) being arbitrary), we must have

\[
m + 1 = \mu + \nu + 1
\]  (3.4.2)
Recurrence Relations

Polynomial Approximations

\[ P_{i(i+1)\ldots(i+m)} = \frac{(x - x_{i+m})P_{i(i+1)\ldots(i+m-1)} + (x_i - x)P_{(i+1)(i+2)\ldots(i+m)}}{x_i - x_{i+m}} \]  
(3.2.3)

Rational Function Approximations

\[ R_{i(i+1)\ldots(i+m)} = R_{(i+1)\ldots(i+m)} + \frac{R_{(i+1)\ldots(i+m)} - R_{i\ldots(i+m-1)}}{x - x_i} \left(1 - \frac{R_{(i+1)\ldots(i+m)} - R_{i\ldots(i+m-1)}}{R_{(i+1)\ldots(i+m)} - R_{(i+1)\ldots(i+m-1)}}\right) - 1 \]  
(3.4.3)
Barycentric Algorithm

Barycentric form of rational interpolant

\[
R(x) = \frac{\sum_{i=0}^{N-1} \frac{w_i}{x - x_i} y_i}{\sum_{i=0}^{N-1} \frac{w_i}{x - x_i}}
\]  
(3.4.9)

Formula for the weights

\[
w_k = \sum_{i=k-d}^{k} (-1)^k \prod_{j=i}^{i+d} \frac{1}{x_k - x_j}
\]  
(3.4.10)

For example,

\[
w_k = (-1)^k, \quad d = 0
\]

\[
w_k = (-1)^{k-1} \left[ \frac{1}{x_k - x_{k-1}} + \frac{1}{x_{k+1} - x_k} \right], \quad d = 1
\]  
(3.4.11)
Coefficients of Interpolating Polynomials and Vandermonde Matrix

\[ y = c_0 + c_1 x + c_2 x^2 + \cdots + c_{N-1} x^{N-1} \]  

(3.5.1)

The \( c_i \)'s are required to satisfy the linear equation

\[
\begin{bmatrix}
1 & x_0 & x_0^2 & \cdots & x_0^{N-1} \\
1 & x_1 & x_1^2 & \cdots & x_1^{N-1} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & x_{N-1} & x_{N-1}^2 & \cdots & x_{N-1}^{N-1}
\end{bmatrix}
\begin{bmatrix}
c_0 \\
c_1 \\
\vdots \\
c_{N-1}
\end{bmatrix}
= 
\begin{bmatrix}
y_0 \\
y_1 \\
\vdots \\
y_{N-1}
\end{bmatrix}
\]  

(3.5.2)

- Transforms coefficients of a polynomial to the actual values it takes at particular points.
- Vandermonde determinant is non-vanishing for these points proving that the mapping is a one-to-one correspondence between coefficients and values . . . i.e. coefficients in polynomial interpolation have a unique solution.
Coefficients continued

- Problems
  - Ill-conditioned as N increases . . . so technique only practical for small data sets
  - If coefficients are used to interpolate functions, the interpolation will not pass through data points
  - First algorithm in NR3 has $O(N^2)$; second has $O(N^3)$
  - For high degrees of interpolation, precision of coefficients is essential . . . so interpolation error compounded by inaccuracy of coefficients
Citations

http://www.alglib.net/interpolation/rational.php
http://www.alglib.net/interpolation/spline3.php
http://en.wikipedia.org/wiki/Linear_interpolation
http://en.wikipedia.org/wiki/Polynomial_interpolation