midterm project

LIU YU TING

November 2 2025

1 Introduction

Compared with polynomial interpolation, rational approximation can more accurately extrapolate the behavior of a real or complex function beyond the known region. However, when using rational approximations, numerical instability and spurious poles may occur. The AAA algorithm addresses these issues by automatically selecting support points through an adaptive greedy strategy and solving a least-squares problem via SVD, which makes the approximation stable and significantly reduces the occurrence of spurious poles.

2 Method Overview

The rational barycentric representation is

$$r(z) = \frac{n(z)}{d(z)} = \frac{\sum_{j=1}^{m} \frac{w_j f_j}{z-z_j}}{\sum_{j=1}^{m} \frac{w_j}{z-z_j}}$$

where $z_1, \cdots, z_m$ are support points, $\omega_1, \cdots, \omega_m$ are weights, and $f_1, \cdots, f_m$ are data values. The representation is composed from quotients $\{\frac{1}{z-z_j}\}_{j=1}^m$. Since AAA algorithm chooses the support point by the greedy algorithm, the quotients $\{\frac{1}{z-z_j}\}_{j=1}^m$ are nearly independent, resulting in a stable representation.

Let $l(z) = \prod_{j=1}^m (z - z_j)$. Then

$$r(z) = \frac{n(z)}{d(z)} = \frac{n(z) \times l(z)}{d(z) \times l(z)} = \sum_{j=1}^{m} \frac{\omega_j f_j \prod_{i=1, i\neq j}^m (z - z_i)}{\omega_j \prod_{i=1, i\neq j}^m (z - z_i)}$$

Thus for all $z_j \in (z_1, \cdots, z_m)$, $r(z_j) = f_j$. We have z_j is not a pole of r. Therefore, $r(z)$ is well-defined on the entire set of support points.

2.1 AAA algorithm

Let $Z = \{z_1, \cdots, z_M\} \subseteq \mathrm{C}$ be the sample set. Let $Z^{(m)} = Z \setminus \{z_1, \cdots, z_m\}$ be the set of points which have not been selected. Suppose that all data values $f(z)$ are known for $z \in Z$.

At each step m, the algorithm will choose a new support point z_m by greedy algorithm and calculate the corresponding weights $\omega_1, \cdots, \omega_m$ by solving a least-squares problem, then compute a new rational function r of type $(m-1, m-1)$. The process is iterated until the approximation error $\max_{z \in Z} |f(z) - r(z)|$ is less than the prescribed tolerance.

2.2 Find the support point by greedy algorithm

To choose the first support point, we compute the average of all function values $f(z)$ over Z, then select the point z_1 such that $f(z_1)$ is farthest from this average.
At each subsequent step, find the point $z \in Z^{(m-1)}$ that maximizes the approximation error

$$|f(z) - n_{m-1}(z)/d_{m-1}(z)| = \|f - n_{m-1}/d_{m-1}\|_{Z^{(m-1)}},$$

then this point will be chosen as the m-th support point z_m.

2.3 Choose the weights by solving the least-squares problem

Since we are approaching f, we want the value $\|f - r\|_Z$ as small as we can. $||f - r||_Z$ can be linearized by multiplying $d(z)$, leading to the least squares form $\|fn - d\|_{Z^{(m)}}$. We can solve which $\omega = (\omega_1, \cdots, \omega_m)^T$ with $||\omega|| = 1$ can minimize $\|fd - n\|_{Z^{(m)}}$ by SVD.

Let $Z^{(m)} = (Z_1^{(m)}, \cdots, Z_{M-m}^{(m)})^T$ and $F^{(m)} = (F_1^{(m)}, \cdots, F_{M-m}^{(m)})^T$ where $F^{(m)} = f(Z^{(m)})$.
At the step m, for all $z_i \in Z^{(m)}$,

$$f(z_i)n(z_i) - d(z_i) = \sum_{j=1}^{m} \frac{w_j F_i^{(m)}}{Z_i^{(m)} - z_j} - \sum_{j=1}^{m} \frac{w_j f_j}{Z_i^{(m)} - z_j} = \sum_{j=1}^{m} \frac{w_j (F_i^{(m)} - f_j)}{Z_i^{(m)} - z_j}$$

It can be written as a $A^{(m)}\omega$, where $A^{(m)}$ is a $(M-m) \times m$ matrix with

$$a_{ij} = \frac{F_i^{(m)} - f_j}{Z_i^{(m)} - z_j}.$$

Let $A^{(m)} = U\Sigma V^T$ where $U = [u_1, \cdots, u_m]$, $V = [v_1, \cdots, v_m]$ and $\Sigma = \mathrm{diag}(\sigma_1, \cdots, \sigma_m)$ with $\sigma_1 \geq \cdots \geq \sigma_m \geq 0$. Since $V^T V = I$, we have $A^{(m)}V = U\Sigma$.

Since σ_m is the smallest in $(\sigma_1, \cdots, \sigma_m)$, we know that

$$\|A^{(m)}v_m\|_{Z^{(m)}} = \|\omega_m u_m\|_{Z^{(m)}} = \min_{\omega \in R^m} \|A^{(m)}\omega\|_{Z^{(m)}}$$

Thus, $\omega = v_m$.
Let $S_F = \text{diag}(F_1^{(m)}, \cdots, F_{M-m}^{(m)})$, $S_f = \text{diag}(f_1, \cdots, f_m)$ and $C^{(m)}$ be a $(M-m) \times m$ matrix with

$$c_{ij} = \frac{1}{Z_i^{(m)} - z_j}$$

Then $A^{(m)}$ can rewrite as $S_F C - C S_f$ and
$N = (n(z_1), \cdots, n(z_{M-m}))^T = C(\omega f)$, $D = (d(z_1), \cdots, d(z_{M-m}))^T = C\omega$

2.4 Stop the iteration

If for all $z_i \in Z^{(m)}$ such that $|f(z_i) - r_m(z_i)| < tol \times \|f\|_\infty$, then stop the iteration.
Thus we have the rational barycentric approximation $r(z) = n(z)/d(z)$.
The zeros of d are generically the poles of r. Let $u(z) = [1, \frac{1}{z-z_1}, \cdots, \frac{1}{z-z_m}]^T$ and

$$A = \begin{pmatrix} 0 & \omega_1 & \omega_2 & \cdots & \omega_m \\ 1 & z_1 & 0 & \cdots & 0 \\ 1 & 0 & z_2 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 1 & 0 & \cdots & 0 & z_m \end{pmatrix} \qquad b = \begin{pmatrix} 0 & & & & \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & \\ & & & & 1 \end{pmatrix}$$

Then $Au(\lambda) = [d(\lambda), \frac{\lambda}{\lambda - z_1}, \cdots, \frac{\lambda}{\lambda - z_m}]^T = \lambda b u(\lambda) + [d(\lambda), 0, \cdots, 0]^T$.
Thus $Au(\lambda) = \lambda b u(\lambda)$ if and only if $d(\lambda) = 0$.

3 Experiment

We apply the AAA algorithm on [-1.5 , 1.5] to approximate the gamma function, and then extend this approximation to [-3.5 , 4.5].

We can see that the extended approximation is accurate. The error is large when $m = 1, 2$ because the chosen support points lie near the poles, which is consistent with the greedy selection rule of the algorithm.

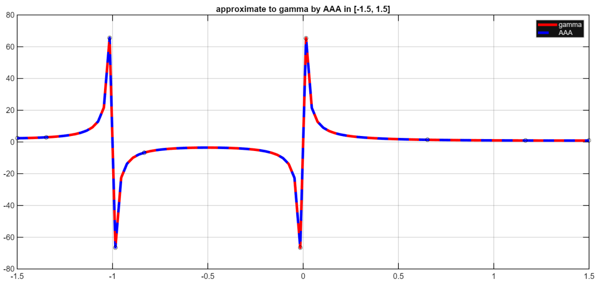

Figure 1: AAA approximation on [-1.5, 1.5]

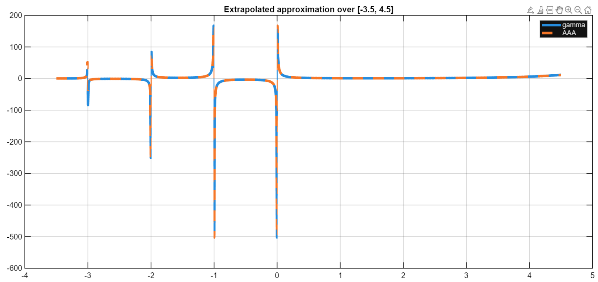

Figure 2: extrapolation to [-3.5, 4.5]

m	error
1	1.322e+02
2	6.839e+02
3	6.952e-01
4	6.283e-01
5	1.070e-03
6	6.271e-02
7	1.444e-06
8	3.167e-08
9	6.717e-10
10	1.807e-12

Table 1: Error at each iteration.

m	support point
1	-0.0151
2	-1.0151
3	1.5
4	-0.9848
5	-0.8333
6	-1.5
7	0.6515
8	0.0151
9	1.1666
10	-1.3484

Table 2: Support points.

4 Summary

The AAA algorithm provides an efficient and numerically stable way to construct rational approximations from discrete data. It achieves high accuracy with only a few lines of MATLAB code and minimal user input, thanks to its automatic selection of support points and stable barycentric formulation. The method performs well even for functions with poles, as demonstrated in the gamma function experiment.

A possible drawback of the AAA algorithm is the appearance of spurious poles, which may occur during the iteration when the tolerance is set too small or the maximum number of steps is too large. However, these spurious poles can be detected by their small residues and eliminated by solving the least-squares problem again to obtain a new rational barycentric representation without spurious poles.