Rational approximation

MATH 6610 Lecture 28

November 20, 2020

Types of approximation

We considered two types of approximation:

- Fourier Series approximation (periodic functions)
- Polynomial approximation (mostly interpolation)

Both of these methods have certain (dis)advantages.

Types of approximation

We considered two types of approximation:

- Fourier Series approximation (periodic functions)
- Polynomial approximation (mostly interpolation)

Both of these methods have certain (dis)advantages.

The last type of approximation we'll consider is *rational* approximation.

General setup: univariate scalar-valued functions, but can be complex valued.

Rational functions

A function $R : \mathbb{C} \to \mathbb{C}$ is a rational function if it is a ratio of polynomials:

$$r(z) := \frac{p(z)}{q(z)}, \qquad\qquad p, q \in P_n,$$

where P_n is the space of polynomials of degree n and less.

Terminology: r is a rational function of "type $(\deg p, \deg q)$".

We'll assume throughout that p and q have no common (non-constant) divisors.

The function r is a ("strictly") *proper rational function* if $\deg p < \deg q$.

Note that p and q are not unique without specifying a normalization.

Rational functions

A function $R : \mathbb{C} \to \mathbb{C}$ is a rational function if it is a ratio of polynomials:

$$r(z) := \frac{p(z)}{q(z)}, \qquad\qquad p, q \in P_n,$$

where P_n is the space of polynomials of degree n and less.

Terminology: r is a rational function of "type $(\deg p, \deg q)$".

We'll assume throughout that p and q have no common (non-constant) divisors.

The function r is a ("strictly") *proper rational function* if $\deg p < \deg q$.

Note that p and q are not unique without specifying a normalization.

Goal: given f, construct r such that $f \approx r$.

Why is this better (worse?) than polynomial approximation or Fourier Series?

Padè approximation

One strategy for constructing rational functions is Padè approximation.

The main idea: choose $r = p/q$ such that

$$f(z) = \frac{p(z)}{q(z)} + \mathcal{O}(x^{n+m+1}), \qquad \deg p = m, \qquad \deg q = n.$$

I.e., match Taylor coefficients to as high an order as possible.

Padè approximation

One strategy for constructing rational functions is Padè approximation.

The main idea: choose $r = p/q$ such that

$$f(z) = \frac{p(z)}{q(z)} + \mathcal{O}(x^{n+m+1}), \qquad \deg p = m, \qquad \deg q = n.$$

I.e., match Taylor coefficients to as high an order as possible.

Specifically, suppose p and q have the form,

$$r(z) = \frac{p(z)}{q(z)} = \frac{\sum_{j=0}^{m} a_j x^j}{1 + \sum_{j=1}^{n} b_j x^j},$$

for some coefficients $a_0, \ldots, a_m$ and $b_1, \ldots, b_n$.

Padè approximation

One strategy for constructing rational functions is Padè approximation.

The main idea: choose $r = p/q$ such that

$$f(z) = \frac{p(z)}{q(z)} + \mathcal{O}(x^{n+m+1}), \qquad \deg p = m, \qquad \deg q = n.$$

I.e., match Taylor coefficients to as high an order as possible.

Specifically, suppose p and q have the form,

$$r(z) = \frac{p(z)}{q(z)} = \frac{\sum_{j=0}^{m} a_j x^j}{1 + \sum_{j=1}^{n} b_j x^j},$$

for some coefficients $a_0, \ldots, a_m$ and $b_1, \ldots, b_n$. The computation can be accomplished in a two-step procedure:

- Compute $\{b_j\}_{j=1}^{n}$ with a linear system matching orders $m+1, \ldots, m+n$.
- Compute $\{a_j\}_{j=0}^{m}$ with a linear system matching orders $0, \ldots, m$.

Rational approximation practicalities

$$f(z) = \frac{p(z)}{q(z)} + \mathcal{O}(x^{n+m+1}), \qquad \deg p = m, \qquad \deg q = n.$$

In order to match coefficients, we need the Taylor expansion of f.

Rational approximation practicalities

$$f(z) = \frac{p(z)}{q(z)} + \mathcal{O}(x^{n+m+1}), \qquad \deg p = m, \qquad \deg q = n.$$

In order to match coefficients, we need the Taylor expansion of f.

This is not so practical, but it does reveal a very useful strategy: linearization.

Consider, e.g., interpolation:

$$r(z_j) = \frac{p(z_j)}{q(z_j)} = f(z_j), \qquad j = 1, \ldots, m+n+1.$$

The difficulty in imposing these conditions: they depend <u>non</u>linearly on coefficients.

Rational approximation practicalities

$$f(z) = \frac{p(z)}{q(z)} + \mathcal{O}(x^{n+m+1}), \qquad \deg p = m, \qquad \deg q = n.$$

In order to match coefficients, we need the Taylor expansion of f.

This is not so practical, but it does reveal a very useful strategy: linearization.

Consider, e.g., interpolation:

$$r(z_j) = \frac{p(z_j)}{q(z_j)} = f(z_j), \qquad j = 1, \ldots, m+n+1.$$

The difficulty in imposing these conditions: they depend <u>non</u>linearly on coefficients.

Linearization: impose these conditions in a different way:

$$q(z_j) f(z_j) = p(z_j), \qquad j = 1, \ldots, m+n+1.$$

This results in a linear system for the a_j, b_j coefficients.

Linearizations

$$f(z) = \frac{p(z)}{q(z)} \quad \longrightarrow \quad q(z)f(z) = p(z).$$

For interpolation and Padè approximation, linearization does not change formulation.

For other conditions, e.g., least-squares, linearization is different.

However, linearization provides a concrete solution strategy.

Linearizations

$$f(z) = \frac{p(z)}{q(z)} \quad \longrightarrow \quad q(z)f(z) = p(z).$$

For interpolation and Padè approximation, linearization does not change formulation.

For other conditions, e.g., least-squares, linearization is different.

However, linearization provides a concrete solution strategy.

There is one problem that linearization doesn't solve: how to ensure a good approximation?

Barycentric form

Consider an alternative "barycentric" formulation for a rational function:

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

By eliminating denominators: this is a type $(m-1, m-1)$ rational function.
(It's actually also a polynomial if w_j are chosen correctly....)

Barycentric form

Consider an alternative "barycentric" formulation for a rational function:

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

By eliminating denominators: this is a type $(m-1, m-1)$ rational function. (It's actually also a polynomial if w_j are chosen correctly....)

The coefficients f_j and w_j are freely chosen complex numbers.

There are some important properties of this approximation:

- If $w_j \neq 0$, then r does not have a pole at $z = z_j$.
- If $w_j \neq 0$, then $r(z_j) = f_j$.
- The above are true independent of how $w_j \neq 0$ are chosen.

The AAA algorithm

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

Given data,

$$(Z_1, \ldots, Z_M), \quad (F_1, \ldots, F_M),$$

with $f(Z_j) = F_j$, and $M \gg m$.

The AAA algorithm

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

Given data,

$$(Z_1, \ldots, Z_M), \quad (F_1, \ldots, F_M),$$

with $f(Z_j) = F_j$, and $M \gg m$.

AAA core ideas:

- "Intelligently" choose interpolation locations $\{z_1, \ldots, z_m\} \subset \{Z_1, \ldots, Z_M\}$ (Hence choose z_j, f_j appropriately)
- The $\{w_j\}_{j=1}^m$ can be chosen arbitrarily: choose them to minimize a least-squares residual.

The algorithm proceeds in an alternating fashion. Let $m = 0$.

1. Choose z_{m+1} (and hence f_{m+1}.
2. Compute weights $\{w_j\}_{j=1}^{m+1}$ using least-squares.
3. $m \leftarrow m + 1$ and repeat steps.

AAA algorithm interpolation

How is z_{m+1} chosen?

- If $m = 0$, choose

$$j^* = \arg\max_j |F_j|, \qquad z_1 = Z_{j*}.$$

- If $m > 1$, choose

$$j^* = \arg\max_j \left|F_j - r(Z_j)\right|, \qquad z_{m+1} = Z_{j*}.$$

The approximation r above is the m-point barycentric rational approximation from the previous step.

AAA algorithm least squares

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

How are the weights $\{w_j\}_{j=1}^m$ chosen?

First note that there is ambiguity in the normalization of the weights, so enforce

$$\|w\|_2 = 1, \qquad\qquad w = (w_1, \ldots, w_m)^T.$$

AAA algorithm least squares

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

How are the weights $\{w_j\}_{j=1}^m$ chosen?

First note that there is ambiguity in the normalization of the weights, so enforce

$$\|w\|_2 = 1, \qquad w = (w_1, \ldots, w_m)^T.$$

The weights are now chosen in the *linearized* least squares sense:

$$w^* = \underset{w \in \mathbb{C}^m}{\arg\min} \sum_{j \in S_m} |d(Z_j) F_j - n(Z_j)|^2 ,$$

where the index set S_m corresponds to the indices j such that Z_j is not an interpolation node:

$$S_m := \left\{ j \in \{1, \ldots, M\} \;\middle|\; Z_j \notin \{z_1, \ldots, z_m\} \right\}.$$

The Loewner matrix

The AAA least-squares minimization problem is equivalent to,

$$\text{Compute } w \in \mathbb{C}^m \text{ such that } \|w\|_2 = 1 \text{ and } \|L_m w\|_2 \text{ is minimized}$$

where L_m is the *Loewner matrix*. With

$$S_m = \{s_1, \ldots, s_{M-m}\},$$

then

$$L_m \in \mathbb{C}^{(M-m)\times m}, \qquad (L_m)_{k,j} = \frac{F_{s_k} - f_j}{Z_{s_k} - z_j},$$

for $k = 1, \ldots, M-m$, and $j = 1, \ldots, m$.

The Loewner matrix

The AAA least-squares minimization problem is equivalent to,

$$\text{Compute } w \in \mathbb{C}^m \text{ such that } \|w\|_2 = 1 \text{ and } \|L_m w\|_2 \text{ is minimized}$$

where L_m is the *Loewner matrix*. With

$$S_m = \{s_1, \ldots, s_{M-m}\},$$

then

$$L_m \in \mathbb{C}^{(M-m)\times m}, \qquad (L_m)_{k,j} = \frac{F_{s_k} - f_j}{Z_{s_k} - z_j},$$

for $k = 1, \ldots, M - m$, and $j = 1, \ldots, m$. I.e., w is a (unit-norm) minimal right-singular vector of L_m.

The AAA algorithm

In summary, here are steps for the AAA algorithm:

Set $m = 0$, set $r(z) = 0$.
Initialize the Loewner matrix L_0 as an $M \times 0$ matrix.

1. Compute z_{m+1} as

$$j^* = \arg\max_j |F_j - r(Z_j)|, \qquad z_{m+1} = Z_{j*}$$

   and set $f_{m+1} = F_{j*}$.
2. Construct L_{m+1} by adding a column and removing a row.
   (Columns correspond to interpolation points, rows to the rest of the points.)
3. Compute $w \in \mathbb{C}^{m+1}$ as the minimal right-singular vector of L_{m+1}.
4. Construct r for $m+1$ using the new weights w and new point (z_{m+1}, f_{m+1}).
5. Terminate if $\max_j |F_j - r(Z_j)|$ is "small enough".
6. Otherwise, $m \leftarrow m + 1$ and repeat.