

Separable Low-rank Barycentric Forms in the p-AAA Algorithm

Linus Balicki, Serkan Gugercin

Abstract

Rational approximation is a powerful tool for capturing the behavior of functions which have singularities on or near a domain of interest. This circumstance makes rational functions ubiquitous in fields such as signal processing, model reduction, and partial differential equations. In recent years the adaptive Antoulas-Anderson (AAA) algorithm [2] has established itself as a successful method for computing rational approximations from a set of sampled data. Our recent work [3] introduced the p-AAA algorithm, extending the original AAA framework to multivariate functions. In order to allow for a clear presentation, we first discuss the two variable case, where the goal is to approximate a function $\mathbf{f} : \mathbb{C}^2 \rightarrow \mathbb{C}$. In this case, p-AAA is given a set of samples

$$\mathbf{F} = \{\mathbf{f}(x_1, x_2) \mid x_1 \in \mathbf{X}_1, x_2 \in \mathbf{X}_2\} \subset \mathbb{C}$$

with the corresponding sampling points

$$\mathbf{X}_1 = \{X_{11}, \dots, X_{1N_1}\} \subset \mathbb{C} \quad \text{and} \quad \mathbf{X}_2 = \{X_{21}, \dots, X_{2N_2}\} \subset \mathbb{C},$$

where X_{ij} denotes the j -th sampling point of the i -th variable. Then the goal is to approximate this data via a rational function represented as a multivariate barycentric form, i.e.,

$$\mathbf{r}(x_1, x_2) = \frac{\mathbf{n}(x_1, x_2)}{\mathbf{d}(x_1, x_2)} = \frac{\sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \frac{\alpha_{i_1 i_2} \mathbf{f}(\xi_{1i_1}, \xi_{2i_2})}{(x_1 - \xi_{1i_1})(x_2 - \xi_{2i_2})}}{\sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \frac{\alpha_{i_1 i_2}}{(x_1 - \xi_{1i_1})(x_2 - \xi_{2i_2})}} \quad (1)$$

where $\alpha_{i_1 i_2} \neq 0$. We note that \mathbf{r} is interpolatory in the sense that

$$\mathbf{r}(x_1, x_2) = \mathbf{f}(x_1, x_2) \quad \text{for} \quad x_1 \in \boldsymbol{\xi}_1 \quad \text{and} \quad x_2 \in \boldsymbol{\xi}_2,$$

where

$$\boldsymbol{\xi}_1 = \{\xi_{11}, \dots, \xi_{1n_1}\} \subset \mathbf{X}_1 \quad \text{and} \quad \boldsymbol{\xi}_2 = \{\xi_{21}, \dots, \xi_{2n_2}\} \subset \mathbf{X}_2$$

are the respective sets of interpolation nodes. Similar to before, ξ_{ij} denotes the j -th interpolation node of the i -th variable. The p-AAA algorithm follows an iterative procedure to choose the interpolation nodes as well as the matrix of barycentric coefficients

$$\alpha \in \mathbb{C}^{n_1 \times n_2}, \quad \text{i.e.,} \quad \alpha(i_1, i_2) = \alpha_{i_1 i_2}.$$

Each iteration consists of first performing a greedy selection where we determine

$$(x_1^*, x_2^*) = \arg \max_{(x_1, x_2) \in \mathbf{X}_1 \times \mathbf{X}_2} |\mathbf{r}(x_1, x_2) - \mathbf{f}(x_1, x_2)| \quad (2)$$

and update the interpolation sets via

$$\boldsymbol{\xi}_1 \leftarrow \boldsymbol{\xi}_1 \cup \{x_1^*\} \quad \text{and} \quad \boldsymbol{\xi}_2 \leftarrow \boldsymbol{\xi}_2 \cup \{x_2^*\}.$$

As a second step, the barycentric coefficients are computed by solving a linear least-squares (LS) problem of the form

$$\min_{\|\alpha\|_{\mathbf{F}}=1} \|\mathbb{L}_2 \text{vec}(\alpha)\|_2^2, \quad (3)$$

where $\mathbb{L}_2 \in \mathbb{C}^{N_1 N_2 \times n_1 n_2}$ is the 2D Loewner matrix. The LS problem above arises as the minimization of the approximation error

$$\sum_{i_1=1}^{N_1} \sum_{i_2=1}^{N_2} |\mathbf{f}(X_{1i_1}, X_{2i_2}) - \mathbf{r}(X_{1i_1}, X_{2i_2})|^2 = \sum_{i_1=1}^{N_1} \sum_{i_2=1}^{N_2} \left| \frac{1}{\mathbf{d}(X_{1i_1}, X_{2i_2})} (\mathbf{d}(X_{1i_1}, X_{2i_2}) \mathbf{f}(X_{1i_1}, X_{2i_2}) - \mathbf{n}(X_{1i_1}, X_{2i_2})) \right|^2,$$

which is linearized by dropping the $1/\mathbf{d}(X_{1i_1}, X_{2i_2})$ terms. In other words, the linearized LS problem minimizes the expression

$$\|\mathbb{L}_2 \text{vec}(\alpha)\|_2^2 = \sum_{i_1=1}^{N_1} \sum_{i_2=1}^{N_2} |(\mathbf{d}(X_{1i_1}, X_{2i_2}) \mathbf{f}(X_{1i_1}, X_{2i_2}) - \mathbf{n}(X_{1i_1}, X_{2i_2}))|^2.$$

This procedure is repeated until the approximation error indicated in (2) drops below a desired error tolerance. Solving the LS problem in (3) is the dominant cost of p-AAA and is done via a singular value decomposition of \mathbb{L}_2 . More precisely, (3) has a closed form solution which is given in terms of the right singular vector of \mathbb{L}_2 which corresponds to the smallest singular value.

While we only outlined the two variable case so far, the algorithm can easily be formulated as an approximation procedure for functions $\mathbf{f} : \mathbb{C}^d \rightarrow \mathbb{C}$ that depend on $d > 2$ variables. The key adjustments that need to be taken into account are that the multivariate approximant $\mathbf{r}(x_1, x_2, \dots, x_d)$ will depend on a tensor

$$\alpha \in \mathbb{C}^{n_1 \times \dots \times n_d} \quad (4)$$

of barycentric coefficients (rather than a matrix) and the p-AAA LS problem will be based on the higher-order Loewner matrix $\mathbb{L}_d \in \mathbb{C}^{N_1 \dots N_d \times n_1 \dots n_d}$. In this case solving this dense LS problem via SVD requires $\mathcal{O}(N_1 \dots N_d n_1^2 \dots n_d^2)$ operations and thus computing α or even forming \mathbb{L}_d may become an infeasible task. We note that this growth in complexity is a common issue in multivariate approximation algorithms and is typically referred to as the ‘‘curse of dimensionality’’. While there exist approaches to overcome these obstacles in multivariate function approximation (e.g., sparse grids, radial basis schemes), we focus here on a method that leverages a separable representation of the denominator \mathbf{d} of the rational approximant \mathbf{r} . As we will point out in the following, such a representation is directly connected to low-rank representations of higher-order tensors and allows for partially overcoming the curse of dimensionality associated with the p-AAA LS problem.

In order to introduce our proposed approach, we consider a canonical (CP) [1] decomposition of the tensor α in (4) which we write in its vectorized form as

$$\text{vec}(\alpha) = \sum_{\ell=1}^r \beta_{1\ell} \otimes \dots \otimes \beta_{d\ell} \in \mathbb{C}^{n_1 \dots n_d},$$

where $\beta_{i\ell} \in \mathbb{C}^{n_i}$ for $\ell = 1, \dots, r$. The matrices $\beta_1 = [\beta_{11}, \dots, \beta_{1r}] \in \mathbb{C}^{n_1 \times r}$, ..., $\beta_d = [\beta_{d1}, \dots, \beta_{dr}] \in \mathbb{C}^{n_d \times r}$ are called CP factors and the smallest r for which such a decomposition exists defines the tensor rank of α . The CP decomposition is particularly useful if it is able to represent (or approximate) α with a small number of terms $r \ll n_1, \dots, n_d$. In this case the storage requirement for the CP factors is merely $\mathcal{O}(r(n_1 + \dots + n_d))$ rather than $\mathcal{O}(n_1 \dots n_d)$ for the full tensor. We

propose to take advantage of this reduction in the degrees of freedom in the representation of α within the p-AAA algorithm.

To make this idea more clear, we revisit the two variable case. There the CP decomposition is analogous to a rank- r outer product representation given by

$$\alpha = \beta_1 \beta_2^\top, \quad (5)$$

Plugging this representation for α into the denominator in (1) gives the separable representation

$$\mathbf{d}(x_1, x_2) = \sum_{\ell=1}^r \left(\sum_{i_1=1}^{n_1} \frac{(\beta_{1\ell})_{i_1}}{x_1 - \xi_{1i_1}} \right) \left(\sum_{i_2=1}^{n_2} \frac{(\beta_{2\ell})_{i_2}}{x_2 - \xi_{2i_2}} \right),$$

where $(\beta_{1\ell})_{i_1} \in \mathbb{C}$ is the i_1 -th entry of the vector $\beta_{1\ell} \in \mathbb{C}^{n_1}$ and $(\beta_{2\ell})_{i_2} \in \mathbb{C}$ is the i_2 -th entry of the vector $\beta_{2\ell} \in \mathbb{C}^{n_2}$. Our main idea for incorporating such separable representations and the associated low-rank decomposition for the barycentric coefficients into the p-AAA algorithm is to add the decomposition introduced in (5) as a constraint to the LS problem in (3). In this case we obtain the LS problem

$$\min_{\beta_1, \beta_2} \left\| \mathbb{L}_2 \sum_{\ell=1}^r \beta_{1\ell} \otimes \beta_{2\ell} \right\|_2^2 \quad \text{s.t.} \quad \left\| \sum_{\ell=1}^r \beta_{1\ell} \otimes \beta_{2\ell} \right\|_2 = 1. \quad (6)$$

We introduce the matrices

$$K_{\beta_1} := [\beta_{11} \otimes I_{n_2}, \dots, \beta_{1r} \otimes I_{n_2}] \in \mathbb{C}^{n_1 n_2 \times n_2 r} \quad \text{and} \quad K_{\beta_2} = [I_{n_1} \otimes \beta_{21}, \dots, I_{n_1} \otimes \beta_{2r}] \in \mathbb{C}^{n_1 n_2 \times n_1 r},$$

as well as the contracted Loewner matrices

$$\mathbb{L}_{\beta_1} := \mathbb{L}_2 K_{\beta_1} \in \mathbb{C}^{N_1 N_2 \times n_2 r} \quad \text{and} \quad \mathbb{L}_{\beta_2} := \mathbb{L}_2 K_{\beta_2} \in \mathbb{C}^{N_1 N_2 \times n_1 r},$$

which allow for writing the constrained LS problem in two distinct ways

$$\min_{\beta_1, \beta_2} \|\mathbb{L}_{\beta_2} \text{vec}(\beta_1)\|_2^2 \quad \text{s.t.} \quad \|K_{\beta_2} \text{vec}(\beta_1)\|_2 = 1, \quad (7)$$

$$\min_{\beta_1, \beta_2} \|\mathbb{L}_{\beta_1} \text{vec}(\beta_2)\|_2^2 \quad \text{s.t.} \quad \|K_{\beta_1} \text{vec}(\beta_2)\|_2 = 1. \quad (8)$$

We note that if one of the factors β_1 or β_2 is fixed, the other one can be obtained by solving an equality constrained LS problem based on the formulation above. In this case, these linear LS problems have a closed form solution in terms of the generalized SVD [4] of the matrix tuple $(\mathbb{L}_{\beta_2}, K_{\beta_2})$ or $(\mathbb{L}_{\beta_1}, K_{\beta_1})$, respectively. Hence, the constrained LS problem in (6) has a separable structure which can be tackled via an alternating least-squares (ALS) procedure. In this procedure, we start with an initial guess for β_1 and β_2 , then repeatedly solve the problem in (7) while keeping β_2 fixed, and the problem in (8) while keeping β_1 fixed. This ALS approach requires $\mathcal{O}(N_1 N_2 r^2 (n_1^2 + n_2^2))$ operations which corresponds to the cost of computing the generalized SVDs. While this change in complexity may only have a small impact in the two variable case, it can be critical when moving to $d > 2$ variables where ALS requires $\mathcal{O}(N_1 \cdots N_d r^2 (n_1^2 + \cdots + n_d^2))$ operations. Additionally, we note that the contracted Loewner matrices can be assembled efficiently by exploiting the Kronecker structure present in \mathbb{L}_d . These facts make it appealing to combine p-AAA with a separable representation for the denominator \mathbf{d} .

We conclude this abstract by considering a simple example where our proposed low-rank version of the p-AAA algorithm yields a high-fidelity rational approximant, while the standard p-AAA algorithm runs out of memory on our machine during the construction of the Loewner matrix \mathbb{L}_d . Specifically, we consider approximating the function

$$\mathbf{f}(x_1, x_2, x_3, x_4, x_5) = \frac{x_1 + x_2 + x_3 + x_4 + x_5}{10 + \sin(x_1) + \sin(x_2) + \sin(x_3) + \sin(x_4) + \sin(x_5)}$$

on the domain $[-3, 3]^5$. For each variable we choose the same sampling points corresponding to 20 linearly spaced values in the interval $[-3, 3]$. We run the proposed low-rank version of p-AAA and enforce a rank $r = 1$ constraint on the coefficient tensor α . After 6 iterations the relative maximum approximation error over the sampled data is approximately 8.514×10^{-3} and p-AAA chose 6 interpolation nodes for each variable. The standard p-AAA algorithm does not make it past the third iteration on our machine, due to running out of memory. Note that in this example the memory requirement for \mathbb{L}_5 is around 24.4 GB in double precision once 4 interpolation nodes are chosen for each variable. In order to evaluate the quality of the computed approximation for unsampled data, we validate \mathbf{r} on a set of samples obtained by sampling 50 linearly spaced points in $[-3, 3]$ for each variable. The maximum relative error on this validation set is approximately 8.704×10^{-3} , which closely matches the maximum error on the training data set.

References

- [1] T. G. Kolda and B. W. Bader. Tensor Decompositions and Applications. *SIAM Review*, 51(3):455–500, Aug. 2009. Publisher: Society for Industrial and Applied Mathematics.
- [2] Y. Nakatsukasa, O. Sète, and L. N. Trefethen. The AAA Algorithm for Rational Approximation. *SIAM Journal on Scientific Computing*, 40(3):A1494–A1522, 2018.
- [3] A. C. Rodriguez, L. Balicki, and S. Gugercin. The p-AAA Algorithm for Data-Driven Modeling of Parametric Dynamical Systems. *SIAM Journal on Scientific Computing*, 45(3):A1332–A1358, 2023.
- [4] C. F. Van Loan. Generalizing the Singular Value Decomposition. *SIAM Journal on Numerical Analysis*, 13(1):76–83, Mar. 1976. Publisher: Society for Industrial and Applied Mathematics.