NEAR OPTIMAL RATIONAL APPROXIMATIONS OF LARGE DATA SETS

ANIL DAMLE, GREGORY BEYLKIN, TERRY HAUT AND LUCAS MONZÓN

Abstract. We introduce a new computationally efficient algorithm for constructing near optimal rational approximations of large (one-dimensional) data sets. In contrast to wavelet-type approximations, these new approximations are effectively shift invariant. We note that the complexity of current algorithms for computing near optimal rational approximations prevents their use for large data sets.

In order to obtain a near optimal rational approximation of a large data set, we first construct its intermediate B-spline representation. Then, by using a new rational approximation of B-splines, we arrive at a suboptimal rational approximation of the data set. We then use a recently developed fast and accurate reduction algorithm for obtaining a near optimal rational approximation

in contrast to wavelet decompositions, rational functions are closed under translations and, thus, optimal rational approximations are shift invariant. Indeed, shifting an optimal rational approximation yields the optimal approximation of the shifted function or data.

Our rational representations are optimal in the sense that, for a given accuracy of approximation, the number of poles is minimal. We say that the approximation is "near optimal" if, instead of the desired accuracy ϵ, our algorithms yield accuracy ϵ', where ϵ' is slightly smaller than ϵ. In such case the number of poles may not be minimal in the strict sense (we note that we have an *a posteriori* check to identify such situation, if needed). We use the term "suboptimal", if we know that the number of poles definitely exceeds the optimal number (for a given accuracy).

For functions given analytically or for functions described by a relatively small number of samples, there are several methods for obtaining their near optimal rational approximations [5, 6, 7]. For a large data set these methods are impractical due to their computational complexity. On the other hand, computing a wavelet decomposition of a large data set does not present a difficulty since its computational cost is linear in the number of samples; we use these facts in our approach.

We first compute a B-spline representation of the data, which provides a simple and efficient method for a transition to a suboptimal rational representation. For this purpose, we construct a new rational approximation of B-splines, where the poles are arranged on a rectangular grid aligned with the location of spline knots. We then split the data into large segments, and compute suboptimal rational approximations for each segment. Finally, we compute a near optimal rational approximation using a recently developed, fast and accurate algorithm in [10].

Although the example provided here is compression of audio recordings, the algorithm may be used to compress and analyze other types of signals, e.g., signals obtained by continuous, global seismic monitoring. In particular, we view compression via near optimal rational approximations as the first step in signal analysis since the poles carry frequency and time information. As shown in [6], poles of near optimal rational approximations concentrate near the singularities of functions. For signals, this corresponds to locations of rapid change, such as occurring when a piano key is struck or at wave arrivals in seismic recordings. The location of the poles also carries information about local frequency content of the signal in a manner similar to wavelets, i.e., the logarithmic distance of these locations from the real axis corresponds to wavelet scales.

We start in Section 2 by providing the background information on the key existing algorithms that facilitate the development of our new approach. Next, in Section 3, we construct a rational approximation (with special properties) of a B-spline to be used in intermediate computations. Then, in Section 4, we describe in detail the algorithm for constructing near optimal

Following [6], from (1) we obtain the rational representation

$$\mathbf{f}(\mathbf{x}) = -2\mathcal{R}\mathbf{e}\left(\sum_{\mathbf{j}=1}^{\mathbf{M}} \quad \mathbf{w_j}\right.$$

. In contrast to standard algorithms, the con-eigenvalues (and con-eigenvectors) are computed with high relative accuracy in $\mathcal{O}\left(M^2 M_0\right)$ operations.

Find all the roots inside the unit disk of the function

$$v(z) = \frac{1}{\sigma_M} \sum_{j=1}^{M_0} \frac{\sqrt{\overline{d_j}}\,\overline{u_j}}{1 - \overline{\mu_j} z}.$$

Note that there are exactly M roots ν_l of $v(z)$ inside the unit disk based on results from [3].

Finally solve for the residuals r_l of $r(z)$ by solving the $M \quad M$ linear system

$$\sum_{j=1}^{M} \frac{r_j}{1 - \nu_j \overline{\nu_k}} = \sum_{j=1}^{M_0} \frac{d_j}{1 - \mu_j \overline{\nu_k}}.$$

Using this algorithm, we obtain $f - r \approx \sigma_M$, which provides a near optimal representation of $f(z)$ using only M pairs of conjugate-reciprocal poles, ν_l and $\overline{\nu}_l^{-1}$. The computational complexit12Tf.4801S091Tf9obtainexactly

Given a uniformly sampled 1-periodic function f, we seek the coefficients α_j such that

$$f\left(\frac{k}{2N}\right) = \sum_{j=0}^{2N} \alpha_j \beta_m(k-j), \quad k = 0, \dots, 2N. \tag{7}$$

The algorithm in [4, 12] rapidly computes the coefficients α_j in (7) using the Fast Fourier Transform (FFT). It performs the following steps:

Set $f_k = f(\frac{k}{2N})$ and compute, for $k = 0, \dots, 2N$,

$$\widehat{f}_k = \sum_{n=0}^{2N} f_n e^{\frac{-2\pi i}{2N+1}kn}$$

using the FFT.

Compute, for $k = 0, \dots, 2N$,

$$\widehat{\alpha}_k = \frac{\widehat{f}_k}{a_m(\frac{k}{2N+1})}.$$

The B-spline coefficients are now obtained via the FFT as

$$\alpha_j = \frac{1}{2N+1} \sum_{n=0}^{2N} \widehat{\alpha}_n e^{\frac{2\pi i}{2N+1}jn}, \quad j = 0, \dots, 2N$$

This algorithm requires $\mathcal{O}(N \log N)$ operations. The details may be found in the appendix in [12].

3. Rational representation of B-splines

In this section we construct rational approximations of B-splines. In our construction we force the real parts of the poles to be integers $l \in Z$, so that the poles are aligned with the knots of the B-spline. As we explain below, this reduces the cost of intermediate computations.

Specifically, we are looking for a suboptimal rational approximation of the form (5), with poles $l \pm i\tau_k$, so that

$$\left| \beta_m(x) + 2 \sum_{l=-\frac{m+1}{2}}^{\frac{m+1}{2}} \sum_{k=1}^{R} \frac{u_{k,l}(x-l) - v_{k,l}\tau_k}{(x-l)^2 + \tau_k^2} \right| \quad \square, \tag{8}$$

where the number of rows of poles, R, will be chosen later. We note that the constraint on the real part of the poles arranges them on a rectangular grid (see Figure 2).

We start by computing a near optimal rational approximation of a B-spline following the approach in [6]. For a given m, we evaluate $\widehat{\beta}$ at a sufficient number of samples; specifically for $m =$ we have

$$h_n = \widehat{\beta}_m\left(\frac{n}{32}\right), \quad n = 0, 1, \dots, \ 00, \tag{9}$$

where

$$\widehat{\beta}_m(\xi) = \left(\frac{\sin \pi\xi}{\pi\xi}\right)^{m+1}, \tag{10}$$

and use the algorithm in Section 2.1 to construct a near optimal rational approximation.

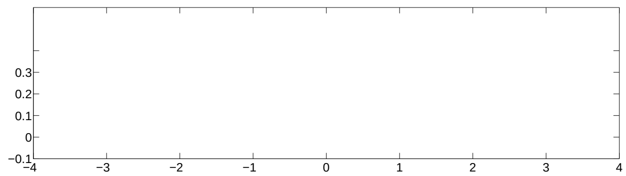
An example of a near optimal rational approximation of a B-spline of degree $m = 3$ may be found in [6]. As observed in that paper, the poles concentrate towards the locations of the knots of the B-spline since its third derivative is discontinuous at these points. In our application we would like to use a higher degree B-spline to lessen the impact of discontinuities and obtain fewer poles. In Figure 1 we present the results for a near optimal approximation of a th degree B-spline using the same approach as in [6]. Since the poles, $t_j \pm i s_j$, appear in complex conjugate pairs, in Figure 1 we display (on a $\log_{10}$ scale) only those with negative imaginary part.

We then seek a suboptimal rational representation of $\beta(x)$ with poles in the locations indicated in (8) and use the near optimal approximation to select the parameters τ_k in (8). Taking into account that the poles closer to the real line are responsible for the high frequency content of the representation, whereas those furthest away capture the lower frequency content, we limit the range for the imaginary parts of our suboptimal poles by using the corresponding maximum, $s^+,$ and minimum, $s^-,$ of the near optimal poles. We select three rows of poles, i.e., $R = 3$ in (8), by choosing imaginary parts $\tau_1 = s^+$, $\tau_3 = s^-$, and

$$\tau_2 = e^{\frac{1}{2}(\log \tau_1 + \log \tau_3)}.$$

The real part for all of these poles are at locations l, where $l = -\frac{m+1}{2}, \ldots, \frac{m+1}{2}$ (recall that m is odd). The choice of three rows of poles is based on the degree of the B-spline and our accuracy requirements (see Figure 2(b)) and may be different in other applications.

Once the location of poles is fixed, the weights in (8) are obtained by solvd[()-M45.96020r7


0.3
0.2
0.1
0
−0.1
−4
−3
−2
−1
0
1
2
3
4

the optimization package CVX [9]. The resulting absolute error is shown in Figure 2(b).

4. Near optimal rational approximations

We now briefly describe the key steps

compute the B-spline coefficients for each section of the signal. Once the B-spline coefficients for each section are found, by adding comthe support of both segments, we preserve the overall accuracy of the merged approximation. In our experiments, we reduce the set of poles parts,

$$
\begin{aligned}
\mathbf{f_k} = \mathbf{f_k^+} + \mathbf{f_k^-} + \mathbf{f_k^{local}} &= \sum_{\mathbf{x_k - t_j \geq \alpha s}} \left(\frac{\mathbf{u_j} + \mathbf{iv_j}}{\mathbf{x_k} - \mathbf{t_j} - \mathbf{is_j}} + \frac{\mathbf{u_j} - \mathbf{iv_j}}{\mathbf{x_k} - \mathbf{t_j} + \mathbf{is_j}} \right) \\
&+ \sum_{\mathbf{t_j - x_k \geq \alpha s}} \left(\frac{\mathbf{u_j} + \mathbf{iv_j}}{\mathbf{x_k} - \mathbf{t_j} - \mathbf{is_j}} + \frac{\mathbf{u_j} - \mathbf{iv_j}}{\mathbf{x_k} - \mathbf{t_j} + \mathbf{is_j}} \right) \qquad (16) \\
&+ \sum_{\mathbf{|x_k - t_j| < \alpha s}} \left(\frac{\mathbf{u_j} + \mathbf{iv_j}}{\mathbf{x_k} - \mathbf{t_j} - \mathbf{is_j}} + \frac{\mathbf{u_j} - \mathbf{iv_j}}{\mathbf{x_k} - \mathbf{t_j} + \mathbf{is_j}} \right),
\end{aligned}
$$

and evaluate $\mathbf{f_k^+}$, $\mathbf{f_k^-}$ and $\mathbf{f_k^{local}}$ separately, where that of $\mathbf{f_k^{local}}$ proceeds directly. The condition on the factor $\boldsymbol{\alpha}$ is decribed below ($\boldsymbol{\alpha} = 5$ is a typical choice). It remains to describe an algorithm for evaluating $\mathbf{f_k^+}$ since $\mathbf{f_k^-}$ is computed in a similar manner.

We have

$$
\begin{aligned}
\mathbf{f_k^+} &= \sum_{\mathbf{x_k - t_j \geq \alpha s}} \left(\frac{\mathbf{u_j} + \mathbf{iv_j}}{\mathbf{x_k} - \mathbf{t_j} - \mathbf{is_j}} + \frac{\mathbf{u_j} - \mathbf{iv_j}}{\mathbf{x_k} - \mathbf{t_j} + \mathbf{is_j}} \right) \\
&= 2 \textstyle\sum_{\mathbf{x_k - t_j \geq \alpha s}} \displaystyle\int_{-\infty}^{\infty} \mathbf{e^{-e^y (x_k - t_j) + y}} \left(\mathbf{u_j} \cos(\mathbf{e^y s_j}) - \mathbf{v_j} \sin(\mathbf{e^y s_j}) \right) \mathbf{dy}. \qquad (17)
\end{aligned}
$$

The effective range of integration in (17) is finite due to the exponential ($\mathbf{y}$ $-$) and super-exponential ($\mathbf{y}$) decay of the integrand. Our choice of the factor $\boldsymbol{\alpha}$ prevents an excessive oscillatory behavior of the integrand within that range. In order to obtain an exponential approximation of the form

$$
\mathbf{f_k^+} = \sum_{\mathbf{t_j \leq x_k}} \sum_{\mathbf{l=1}}^{\mathbf{L}} \boldsymbol{\lambda}_{\mathbf{l,j}} \mathbf{e^{-\mu_l (x_k - t_j)}}, \quad \boldsymbol{\alpha}\mathbf{s} \quad \mathbf{x_k} - \mathbf{t_j} \quad \mathbf{T}, \quad \mathcal{R}\mathbf{e}(\boldsymbol{\mu}_\mathbf{l}) > 0, \qquad (18)
$$

(where $\mathbf{T}$ is sufficiently large to accommodate a given segment of the signal), we may now proceed as in [5, 7]. Indeed, we discretize the integral in (17) to any desired precision and use an appropriate algorithm to reduce the number of terms.

In (18) we may switch the order of summation and, as a result, construct a recursion (see [13, 5]). Denoting

$$
\mathbf{q_{k,l}} = \sum_{\mathbf{t_j \leq x_k}} \boldsymbol{\lambda}_{\mathbf{l,j}} \mathbf{e^{-\mu_l (x_k - t_j)}},
$$

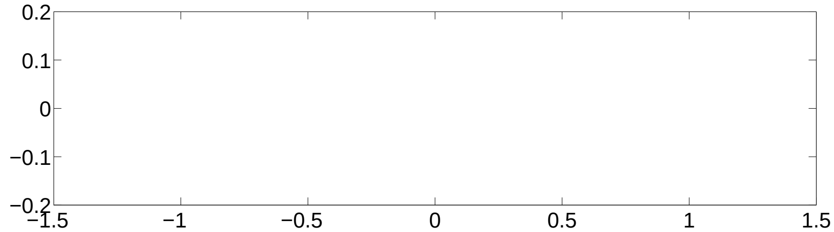
we obtain

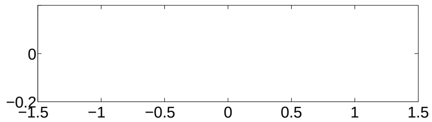
$$
\begin{aligned}
\mathbf{q_{k+1,l}} &= \sum_{\mathbf{t_j \leq x_{k+1}}} \boldsymbol{\lambda}_{\mathbf{l,j}} \mathbf{e^{-\mu_l (x_{k+1} - t_j)}} \\
&= \mathbf{e^{-\mu_l (x_{k+1} - x_k)}} \mathbf{q_{k,l}} + \sum_{\mathbf{x_k < t_j \leq x_{k+1}}} \boldsymbol{\lambda}_{\mathbf{l,j}} \mathbf{e^{-\mu_l (x_{k+1} - t_j)}}.
\end{aligned}
$$

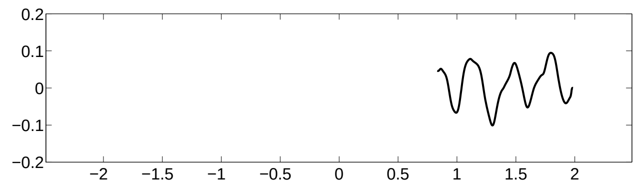
This recursion leads to an $\mathcal{O}(\mathbf{L} \cdot \mathbf{K}) + \mathcal{O}(\mathbf{L} \cdot \mathbf{M})$ algorithm for computing $\mathbf{f_k^+}$.

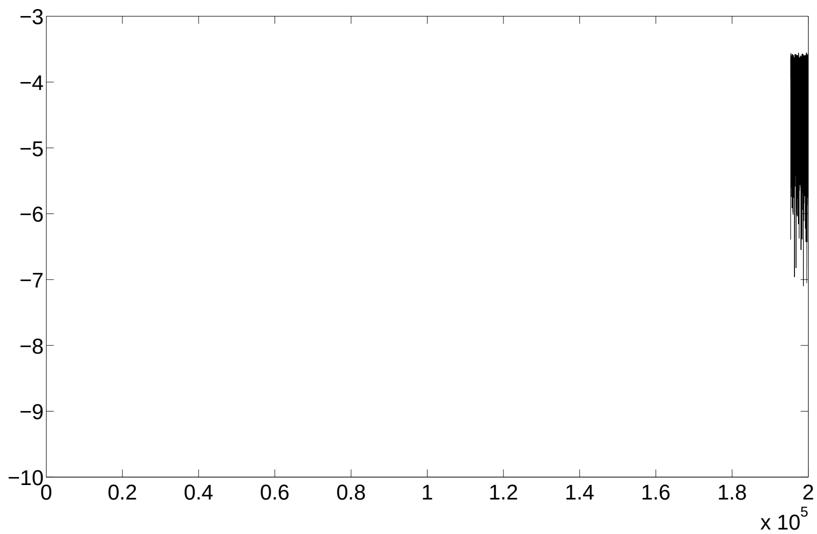
5. Numerical Examples

We have computed several approximations using the algorithm from Section 4. Since one of the potential applications for this method is a compression scheme, we illustrate our algorithm using a large data set from a high quality audio recording.


0.2
0.1
0
−0.1
−0.2
−1.5
−1
−0.5
0
0.5
1
1.5


0
−0.2
−1.5
−1
−0.5
0
0.5
1
1.5


0.2
0.1
0
−0.1
−0.2
−2
−1.5
−1
−0.5
0
0.5
1
1.5
2


−3
−4
−5
−6
−7
−8
−9
−10
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
x 10^5

[12] B. A. Jones, G. H. Born, and G. Beylkin. A Cubed Sphere Gravity Model for Fast Orbit Propagation. *AAS/AIAA Spaceflight Mechanics Meeting, Advances in the Astronautical Sciences*, 134:567–584, 2009.

[13] N. Yarvin and V. Rokhlin. An improved fast multipole algorithm for potential fields on the line. *SIAM J. Numer. Anal.*, 36(2):629–666 (electronic), 1999.

Department of Applied Mathematics, University of Colorado, Boulder, CO 80309-0526