Journal of Computational and Applied Mathematics 326 (2017) 193–203

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

Signal recovery by discrete approximation and a Prony-like method

M.-R. Skrzipek

Fernuniversität in Hagen, Fakultät für Mathematik und Informatik, 58084 Hagen, Germany

ARTICLE INFO

Article history:
Received 2 November 2016
Received in revised form 11 May 2017

MSC 2010:
41A30
41A50
49M25
65F20
90C08
94A12

Keywords:
Discrete approximation
Prony's method
Linear programming
Exponential sum
Overdetermined systems
Recovery of structured functions

ABSTRACT

We introduce an algorithm which combines ideas of Prony's approach to recover signals from given samples with approximation methods. We solve two overdetermined systems of linear equations with linear programming methods and calculate the zeros of a suitable 'Prony-like' polynomial. We get the bandwidth m, the frequencies as well as the amplitudes and some other characteristics of the signal. Especially, it is reconstructed if sufficient many (at least $2m$) samples are given.

If we have too few samples or if they are too much noised, we get an approximation of the original noiseless signal. Even if we have sufficient many but possible erroneous samples the obtained signal interpolates at least m of them (usually the low-noised or noiseless ones) and, of course, all samples if the signal is recovered.

The described method behaves well to moderate sampling errors and is resistant to outliers in the samples which can be detected, filtered off or corrected during the calculations to improve the quality of the computed signal.

1. Introduction

We consider a signal h with finite length,

$$h(x) := \sum_{j=1}^{m} \lambda_j e^{\omega_j x}, \tag{1.1}$$

with $\omega_j \in [-\alpha, 0] + \mathrm{i}\,[-\pi, \pi)$, $\omega_i \neq \omega_j$ for $i \neq j$, $\alpha > 0$, $\mathbb{C} \ni \lambda_j \neq 0$. Especially, all $z_j := e^{\omega_j}$ lie in a circular ring $\mathbb{D}_\alpha := \{z \in \mathbb{C} : e^{-\alpha} \leq |z| \leq 1\}$. Note, that $\operatorname{Re}\omega_j$ is the damping factor, $\operatorname{Im}\omega_j$ the angular frequency of $e^{\omega_j x}$ and $m < \infty$ is called the bandwidth of h.

By $h_k := h(k)$, $k \in \mathbb{N}_0$, we denote sampled values and assume that $\varrho := \limsup_{k\to\infty}(h_k)^{1/k} < \infty$. Consider the z-transform H,

$$H(z) := \sum_{k=0}^{\infty} h_k z^{-k} \tag{1.2}$$

E-mail address: Michael.Skrzipek@fernuni-hagen.de.

http://dx.doi.org/10.1016/j.cam.2017.05.029

which converges on $\{z \in \mathbb{C} : |z| > \varrho\}$. Replacing h_k by its representation from (1.1), we get

$$H(z) = \sum_{j=1}^{m} \lambda_j \frac{z}{z - z_j} = \frac{a(z)}{\rho_m(z)},$$

where $\rho_m,\ \rho_m(z) = \prod_{j=1}^{m}(z - z_j)$, is a monic polynomial of degree m called the Prony polynomial, and a is a polynomial with degree at most m and $a(0) = 0$. By construction, ρ_m has only simple zeros. If we know the z_j, we get the corresponding ω_j via $\omega_j = \operatorname{Log} z_j$ (complex logarithm).

If all ω_j are known, then the amplitudes $\lambda_j,\ j = 1, \ldots, m$, can be obtained from (1.1) as the solution of a system of linear equations, using the samples as interpolation conditions, see Section 4.

The classical way to determine all z_j calculates at first the monomial representation of ρ_m by using a finite set of given samples h_k and then the zeros of this polynomial [1]. In the real world, the measured values h_k contain usually some errors which may also distort the Prony polynomial. Since already small perturbations of the monomial coefficients may cause large changes for its zeros, even if m is small, Prony's original method is rarely taken in signal processing.

It is possible to avoid the explicit usage of the monomial representation and to reformulate the problem as an eigenvalue resp. as a singular value problem which is more suited for numerical purposes than Prony's original method (cf. [2–5] if an l_2-solution is desired, and [6] in a more general context).

Another classical way constructs the Prony polynomial not in its monomial representation, see e.g. [7–14]. Using the given samples, some authors construct moments of a weight function. By a Levinson like algorithm a finite sequence $\{\sigma_\nu\}_{\nu=0,\ldots,m}$ of monic Szegő polynomials resp. its reflection coefficients can be constructed. Then, σ_m is used as the desired approximation of the Prony polynomial; the reflection coefficients are the entries of a Hessenberg matrix from which we get the zeros of this polynomial as its eigenvalues. However, this method has some drawbacks, especially if there are unimodular zeros, and the convergence of these approximations to the 'true' zeros of the Prony polynomial resp. Szegő polynomial is not guaranteed.

Existing algorithms which use the Prony polynomial cannot overcome the drawback that noised samples cause a noised polynomial. Furthermore, it is known that already small perturbations of this polynomial may distort considerably the location of the zeros and wrong frequencies may be detected.

Our approach uses ideas from linear programming and approximation theory to overcome this lack. If we have enough samples and if there are at most moderate (sampling or computing) errors, the signal will be *reconstructed*. Otherwise, the unnoised one (resp. its frequencies, amplitudes, bandwidth) will be *approximated* by our algorithm. Furthermore, possible sample outliers can be detected and removed/corrected to improve the result.

We use the l_1-norm for the real resp. another closely related norm for the complex case. Using these norms, we will see that the calculated signal has some very useful properties to rate the quality of the samples and to detect outliers (besides its reconstruction/approximation property).

The order of approximation (i.e. the number m of recognized frequencies) depends sensitive on the samples, i.e. even low noised samples may change the bandwidth. Some frequencies may be very close together or have amplitudes with small moduli, which often indicate the presence of noised samples. Although these cause no problems for the calculations in the subsequent described algorithm, it is desirable to detect noised samples as early as possible. Such a noising filter is the phase 2 in our algorithm which often reduces the bandwidth and the size of the problem during the computations and can significantly improve the 'quality' of the obtained signal.

In the context of compressive sensing, one is interested in to find a sparse representation of the signal with few active frequencies. For this problem, it is known that in most cases an l_1-solution is also the sparsest one [15], and thus noise caused clustered frequencies are widely avoided by such a solution. Although we have here overdetermined instead of underdetermined systems, our numerical tests hypothesized that the latter behaviour retains valid for the solutions which we calculated by the algorithm we describe below.

It was e.g. already in the year 1964 in [16] noted that in the presence of outliers, wild points, the usage of the l_1-norm, appeared to be markedly superior among the other l_p-norms, $1 < p \le \infty$, at least for a subproblem (finding a best approximation) of the problem we consider here. Especially, these l_1-approximants have smaller variances than the least squares approximants when the errors follow a Laplace distribution or any long tailed distribution.

The least squares principle bases on the assumption that the errors are normally distributed which is usually not the case for given samples. The popularity of least squares methods is rather reasoned in the availability of good algorithms and a well studied underlying theory than on the knowledge of the distribution of the samples.

If the kind of noising is known, pre-filtering the samples is of course useful. However, this is a theme of own interest and we do not consider this aspect here.

We start the next section with some basics about Prony's method which we use in the following. At first, we modify the classical proceeding for the case that m is not known a priori. This leads to an overdetermined system of linear equations which can be solved by using the methods of linear programming as described in Section 3. By 'solving' we mean that we minimize the l_1-norm of the residuals for the real case; for the complex case, this will be slightly weakened. During the computations, m will be calculated and mostly the dimension of the problem can be significantly decreased. The residuals indicate where there are possible sampling errors so that they can be detected and corrected or ignored for further computations.