Hierarchical alternating nonlinear least squares for nonnegative matrix factorization using rational functions

Cécile Hautecoeur
ICTEAM institute
UCLouvain
Louvain-la-Neuve, Belgium
cecile.hautecoeur@uclouvain.be

François Glineur
ICTEAM & CORE institutes
UCLouvain
Louvain-la-Neuve, Belgium
francois.glineur@uclouvain.be

Lieven De Lathauwer
ESAT-STADIUS
KU Leuven
Kortrijk, Belgium
lieven.delathauwer@kuleuven.be

***Abstract*—We present an extension of the widely used Hierarchical Alternating Least Squares (HALS) algorithm to solve Nonnegative Matrix Factorization (NMF) problems using rational functions, in order to unmix discretization of continuous signals. We observe that the use of rational functions in NMF can significantly improve the quality of the reconstruction of noisy data compared to the standard approach based on vectors, and to recent continuous signal factorization approaches using splines or polynomials. We also show that our algorithm obtains state-of-the-art results in the domain of multicomponent nanostructures spectrum image unmixing.**

***Index Terms*—nonnegative matrix factorization (over discretized signals), hierarchical alternating least squares, (projection on) nonnegative rational functions**

I. Introduction

Nonnegative Matrix Factorization (NMF) aims to approximately factorize an input data matrix Y as a product of two lower rank and nonnegative matrices A and X. Besides the dimension reduction, this operation allows to denoise data and to describe each column of Y as a linear combination (via the coefficients of the corresponding column of X) of a few characteristic elements (the columns of A) [3], [9]. The nonnegativity constraint allows to retrieve more interpretable factors A and X. Indeed, this constraint leads to a parts-based representation of input matrix Y, relying only on additive combinations [9]. As a result, NMF is successfully used in various fields and techniques such as image processing, text mining, blind signals unmixing, etc. (see e.g. [5] and references therein).

In many practical situations, input matrix Y consists of a discretization of continuous signals, i.e. each row corresponds to a sampling of input signals at a given time or value. These signals frequently present some structure, such as smoothness. Therefore, one is interested in using this prior by expressing these signals as a linear combination of signals sharing a similar structure, i.e belonging to a class such as polynomials, splines or rational functions.

This work was supported by the Fonds de la Recherche Scientifique - FNRS and the Fonds Wetenschappelijk Onderzoek - Vlaanderen under EOS Project no 30468160.

Often, this class consists of finitely parametrizable functions, i.e. functions that can be described using a finite number of parameters. The advantages of using finite parametrizable functions are two-fold: first, as these functions are entirely described via their parameters, this allows to recover data outside the sampling points of the input signals. Second, the reduction of dimension is stronger as the number of parameters is smaller than the number of sampling points. Therefore, we focus on finitely parametrizable functions in NMF which we denote as "functional NMF" (F-NMF).

Consider a matrix $Y \in \mathbb{R}^{m\times n}$ with n columns $Y_{:i}$ containing samples of continuous signals on m points $\boldsymbol{\tau} = \{\tau_i\}_{i=1}^m$. Without loss of generality, we consider in this paper signals defined on the interval $T = [\tau_1, \tau_m] = [-1, 1]$. Denote $\mathcal{F}$ the chosen set of finitely parametrizable functions, and define $\mathcal{F}_+ \subset \mathcal{F}$ to be the subset of $\mathcal{F}$ containing functions nonnegative on interval T. As input data are only sampled at points $\boldsymbol{\tau}$, we can work on $\mathcal{R}_+ \subset \mathbb{R}_+^m$, the set containing the discretization on $\boldsymbol{\tau}$ of all functions belonging to $\mathcal{F}_+$, i.e. $\mathcal{R}_+ = \{f(\boldsymbol{\tau}) | f \in \mathcal{F}_+\}$, where $f(\boldsymbol{\tau})$ is a vector containing the value of function f at each point τ_i. This leads us to the definition of F-NMF, where we consider the Frobenius norm to evaluate the reconstruction error.

Definition 1 (**F-NMF**). *Given an input matrix $Y \in \mathbb{R}^{m\times n}$, a discretized nonnegative set $\mathcal{R}_+ \in \mathbb{R}_+^m$ and a factorization rank $r \geq 1$, find a matrix $A \in \mathbb{R}_+^{m\times r}$ containing elements of $\mathcal{R}_+$ in each of its columns, i.e. $A_{:j} \in \mathcal{R}_+$, $\forall j$, and a nonnegative matrix $X \in \mathbb{R}_+^{r\times n}$ solving*

$$\min_{A_{:j}\in\mathcal{R}_+, X\in\mathbb{R}_+^{r\times n}} \sum_{i=1}^{n} \left|\left| Y_{:i} - \sum_{j=1}^{r} A_{:j} X_{ji} \right|\right|^2. \tag{1}$$

In this work, we focus on the analysis of F-NMF using rational functions of fixed degree in $\mathcal{F}$, which we denote as R-NMF. The paper is organised as follows: Section II motivates the choice of rational functions, and presents a method to solve the problem, the hierarchical alternating nonlinear least squares method, inspired from [2], which repeatedly solves problems restricted to a single

column/row of A/X. We confirm in Section III that methods based on rational functions are more accurate than polynomials, splines or standard NMF on noisy datasets containing real-world signals with some (near) singularities. Finally, we compare in section IV functional NMF to state-of-the-art methods in the field of multicomponent nanostructures spectrum images unmixing, and conclude that using rational functions in NMF helps the algorithm to be less sensitive to noise in this situation too.

II. Solve NMF using rational functions

A. Nonnegative rational functions

Univariate rational functions are by definition a ratio between two univariate polynomials $r(t) = a(t)/b(t)$. In this work, we consider fixed and known degrees for both the numerator and the denominator, denoted by d_1 and d_2 respectively. Rational functions can describe a wider range of shapes and behaviors than polynomials or splines, like signals with (near) singularities (see e.g., [12]). Moreover, in our context of matrix factorization, they can ensure that the factors are essentially unique when the poles are distinct [4], which alleviates a well-known drawback of nonnegative matrix factorization.

However, rational functions also have disadvantages that make them harder to use in practice. They cannot be expressed as linear combinations of a few basis elements. Moreover, the space of rational functions of fixed numerator and denominator degrees is not a vector space as the addition is not closed: adding two rational functions with same degrees often results in a rational function with other degrees. For the same reason, this space is not convex.

To optimize over nonnegative rational functions, we need to express in a compact way that the function is nonnegative at every single point of an interval. First, observe that nonnegative rational functions can be described without loss of generality using a nonnegative numerator and denominator.

Second, it is known that any polynomial p of degree $2d$ that is nonnegative on interval [-1,1] can be described as [11]: $p(t) = g_1(t)^2 + (1-t^2)g_2(t)^2$, where g_1 and g_2 are degree d and $d-1$ respectively. Moreover, given a polynomial basis $\left\{\Pi_i(t)\right\}_{i=1}^{2d+1}$, in our case the Chebyshev basis, the vector of polynomial values $p(\boldsymbol{\tau})$ can be computed as $V^{2d}\boldsymbol{p}$ where $\boldsymbol{p} \in \mathbb{R}^{2d+1}$ are the coefficients of the polynomial in the chosen basis, and $V^{2d} \in \mathbb{R}^{m\times(2d+1)}$ is a fixed matrix defined as $V^{2d}_{i,j} = \Pi_j(\tau_i)$.

Combining these observations, we can describe the values on discretization points $\boldsymbol{\tau}$ of a rational function nonnegative on $[-1,1]$ with even numerator degree $d_1 = 2d_1'$ and denominator degree $d_2 = 2d_2'$ in the following way:

$$R(\boldsymbol{\tau}) = \frac{(V^{d_1'}\boldsymbol{a_1})^2 + (1-\boldsymbol{\tau}^2)\cdot(V^{d_1'-1}\boldsymbol{a_2})^2}{(V^{d_2'}\boldsymbol{b_1})^2 + (1-\boldsymbol{\tau}^2)\cdot(V^{d_2'-1}\boldsymbol{b_2})^2 + \epsilon}; \quad (2)$$

with $\boldsymbol{a_1} \in \mathbb{R}^{d_1'+1}, \boldsymbol{a_2} \in \mathbb{R}^{d_1'}, \boldsymbol{b_1} \in \mathbb{R}^{d_2'+1}$, $\boldsymbol{b_2} \in \mathbb{R}^{d_2'}$, and ϵ preventing the denominator to go to 0. The square and division operators are performed element-wise, and '$\cdot$' stands for the element-wise multiplication. Similar formulations can be found for nonnegative rational functions with odd degrees and for other intervals. Note that ϵ can be fixed to any number without loss of generality. Hence equation (2) provides a description of the set $\mathcal{R}_+$ with $2d_1+2d_2+2$ parameters.

B. The Hierarchical Alternating Nonlinear Least Squares

In Hierarchical Alternating Least Squares (HALS), the columns of A and the rows of X are updated successively, considering all the other elements as fixed [2]. This is therefore a $2r$-block coordinate descent method. Thanks to the quadratic structure of the objective function, minimizing (1) when all variables are fixed except a column of A or a row of X can be done by projecting the unconstrained minimizer on the corresponding feasible region. This region is the set $\mathcal{R}_+$ of nonnegative rational functions with fixed degrees (for A), or the set $\mathbb{R}^n_+$ of nonnegative vectors (for X). The unconstrained minimizer can easily be found for both columns of A and rows of X by cancelling the gradient. Algorithm 1 sketches this approach, using $[\cdot]_S$ for the projection on set S. The projection on $\mathbb{R}^n_+$ is a simple thresholding operation, setting all negative values to 0, while the projection on the set of nonnegative rational function is not trivial and discussed in next section. Equation (4) is separable, as the value of X_{si} can be computed independently from X_{sj}, but note that this is not the case for $A_{:s}$ in equation (3).

Algorithm 1 R-HANLS

function R-HANLS($Y,\ A,\ X$)
  while Stop condition not encountered **do**
    for $A_{:s} \in A$ **do**

$$A_{:s} \leftarrow \left[\frac{Y(X_{s:})^\top - \sum_{j\neq s} A_{:j}X_{j:}(X_{s:})^\top}{||X_{s:}||^2}\right]_{\mathcal{R}_+} \quad (3)$$

    for $X_{s:} \in X$ **do**

$$X_{s:} \leftarrow \left[\frac{A_{:s}^\top Y - \sum_{j\neq s} A_{:s}^\top A_{:j}X_{j:}}{||A_{:s}||^2}\right]_{\mathbb{R}^n_+} \quad (4)$$

  return $A,\ X$

C. Projection on nonnegative rational functions

To perform step (3) in Algorithm 1 above, we first compute $\boldsymbol{z} = \frac{Y(X_{s:})^\top - \sum_{j\neq s} A_{:j}X_{j:}(X_{s:})^\top}{||X_{s:}||^2}$ which we then must project on $\mathcal{R}_+$, the set of $\boldsymbol{\tau}$-discretizations of rational functions nonnegative on T with degree-d_1 numerator and degree-d_2 denominator. In other words, we aim to find the rational function $\frac{a(t)}{b(t)}$ nonnegative on T that is closest to $\boldsymbol{z}$ at the discretization points $\boldsymbol{\tau}$, which can be written as

$$\min_{a\in\mathcal{P}_+^{d_1}, b\in\mathcal{P}_+^{d_2}} \left|\left|\boldsymbol{z} - \frac{a(\boldsymbol{\tau})}{b(\boldsymbol{\tau})}\right|\right|_2^2 \quad \text{with } b(t) > 0,\ \forall t \in T, \quad (5)$$

where $\mathcal{P}_+^d$ is the set of degree-d polynomials nonnegative on interval $T = [\tau_1, \tau_m] = [-1, 1]$.

The above problem without the nonnegativity constraints $a \in \mathcal{P}_+^{d_1}, b \in \mathcal{P}_+^{d_2}$ has been intensively studied in the literature, see for example [10] and references therein. However, to the best of our knowledge there is not much work on projection or approximation using nonnegative rational functions. In this work, we use equation (2) to write the projection problem in an unconstrained way. For example, when both numerator and denominator have even degree $d_1 = 2d_1'$ and $d_2 = 2d_2'$, the projection becomes

$$\min_{\substack{\boldsymbol{a_1},\boldsymbol{a_2},\\ \boldsymbol{b_1},\boldsymbol{b_2}}} \left|\left| y(\boldsymbol{\tau}) - \frac{(V^{d_1'}\boldsymbol{a_1})^2 + (1-\boldsymbol{\tau}^2)(V^{d_1'-1}\boldsymbol{a_2})^2}{(V^{d_2'}\boldsymbol{b_1})^2 + (1-\boldsymbol{\tau}^2)(V^{d_2'-1}\boldsymbol{b_2})^2 + \epsilon} \right|\right|^2. \quad (6)$$

(we use $\epsilon = 10^{-16}$) and solve the problem using a standard nonlinear least squares solver based on a trust-region reflective algorithm.

This completes the description of our method, which we call Hierarchical Alternating Nonlinear Least Squares (HANLS) as step (3) involves solving a nonlinear least-squares problem, problem (6).

III. Performance of R-HANLS

In this section, we first present a synthetic data set created by mixing real-world signals. Then, we analyze how well our algorithm R-HANLS performs on this data set, and compare it to previous continuous factorization approaches based on HALS over polynomials or splines from [6], and to standard HALS on vectors.

A. Dataset

For our experiments, we use the reflectance signals of Andesine, Celestine and Kaolinite, showed in Figure 1 (left). Those reflectance signals come from the U.S. Geological Survey (USGS) database [7] and are evaluated on 414 nonequally spaced data points. Those signals form the matrix $A \in \mathbb{R}_+^{414\times 3}$. We then generate matrix $X \in \mathbb{R}_+^{3\times 100}$ randomly following a normal distribution $\mathcal{N}(0,1)$ and replace the negative values by 0, so X is 50% sparse. The data provided to the algorithms is then $Y = AX+N$ where N is additive Gaussian noise. Figure 1 (right) shows some of the signals in Y with noise level 20dB. The presented dataset is the one used in this section. We used rational functions with numerator and denominator of degree $d_1 = d_2 = 10$, the number of observations n is equal to 100, and the noise level is 20dB. In all our experiments, when comparing other methods to factorization over rational functions with degrees d_1 and d_2, we consider polynomials of degree $d_1 + d_2$ and splines of degree 3 with $d_1 + d_2 - 1$ interior knots, so that all approaches share the same number of degrees of freedom (except standard HALS which operates over unstructured nonnegative vectors). The selected number of degrees of freedom was chosen to deliver the best results on polynomials and splines, and was not specifically tuned for rational functions. A more in-depth analysis of the influence of this degree is left as future work. Accuracy of the tested methods on obtained factors A', X' is evaluated trough the relative error computed as $\left(\sum_i ||Y_{:i} - \sum_j A'_{:j}X'_{ji}||^2\right)/\left(\sum_i ||Y_{:i}||^2\right)$.

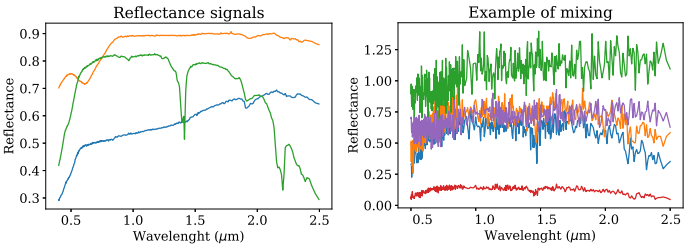


Fig. 1. Left: Reflectance signals of Andesine (blue), Celestine (orange) and Kaolinite (green). Right: Example of mixing of those signals. Each of the five signals is a column of Y.

B. Accelerate the projections in R-HANLS

R-HANLS, described in Algorithm 1, is an iterative algorithm that uses a projection at each iteration. An important characteristic of this algorithm is that the successive iterates for the same block tend to become close to each other. Therefore, we should consider exploiting knowledge from previous iterations: here, we will try to initialize the least squares solver with the previous projection.

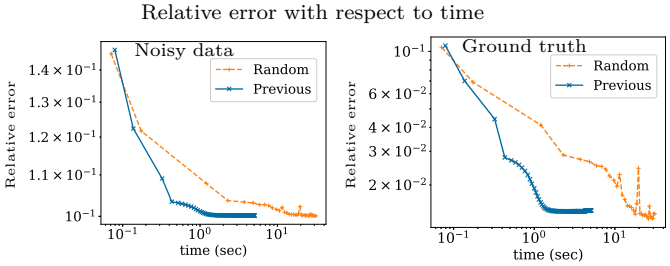


Fig. 2. Evolution of the relative error with respect to time, using R-HANLS. Left figure contains error w.r.t Y, the noisy data, while on right figure it is computed w.r.t the ground truth AX (not provided to the algorithm). First iteration is omitted to improve readability.

Figure 2 compares the performance of R-HANLS using or not information about previous projection. We consider a tolerance of 10^{-4} in the least squares solver. Using information about previous projection appears to significantly accelerate the convergence of the algorithm and reduces its oscillations, but is slightly less accurate. This behavior is probably due to the fact that the set of rational functions of fixed degree is not convex, and using random initialization can allow to escape from a local minimum from time to time. Nevertheless, we initialize the projection algorithm with its previous value in our further experiments. Note that the nonconvexity of the constraint also prevents us from concluding about the convergence of the algorithm to a stationary point, but the following tests highlight that in practice this does not prevent the algorithm to obtain interesting results.

Figure 2 also shows that our algorithm can efficiently denoise data. Indeed, relative error obtained w.r.t. the ground truth (right figure) are smaller than the relative error w.r.t. the noisy data (left) provided to the algorithm.

C. Comparison of the methods

Here we compare R-HANLS to existing continuous factorization approaches based on polynomials and splines, and to standard HALS. Figure 3 shows boxplots of the relative error (computed with respect to the exact data AX, not provided to the algorithm) and the computation time for different methods. The boxplots summarize the results of 10 tests where each algorithm is run 10 times with different initializations. We observe that using functional NMF is more accurate than using standard NMF, and among functional approaches the use of rational functions leads to the most accurate results. This is remarkable as the real-world signals used to generate the mixed inputs were not particularly close to rational functions. In terms of computational time, standard HALS is the fastest method, and R-HANLS is a bit slower than HALS using splines or polynomials. Nevertheless, all methods are able to solve the problem in less than 30 seconds.

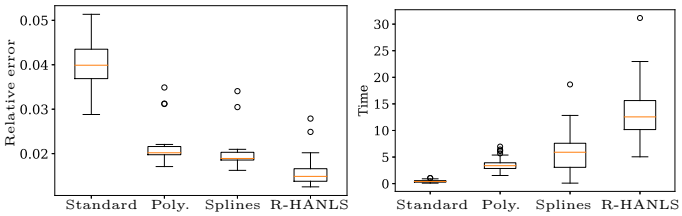


Fig. 3. Boxplots of the relative error (left) and the computation time (right), for different models solving NMF problem on mixture of signals presented in Figure 1.

IV. Application to spectrum images unmixing

We also tested our method to unmix difficult spectrum images of multicomponent nanostructures from [1] and [8], where the authors use a method called MCR-LLM (Multivariate Curve Resolution by Log-Likelihood Maximization) to solve the unmixing problem. We apply the same preprocessing (see the cited papers for more information about the data sets), and scale the input data matrix so that each row has unit Manhattan norm and normalize the rows of matrix X in the decomposition (this is done without loss of generality in NMF by scaling A so that AX stays unchanged). We use k-means clustering to initialize weights in X, and matrix A is initialized using the unconstrained minimizer corresponding to this fixed X (an initial iterate that does not satisfy the constraints does not cause any trouble in our implementation).

A first one-dimensional data set [1] contains the energy loss of the components compared to their position. The shape of such an energy loss does not look like a rational function. However, the relative abundance of each element should vary smoothly with its position, hence we will approximate it using rational functions of degrees $d_1 = d_2 = 20$. Because we use a random initialization, we report the best result out of ten runs for each method (as noise seems to be Poisson [1], we use the Kullback-Leibler divergence on the error to pick the best out of the ten tests). Then we compare the results between different methods visually, as depicted Figure 4.

We observe that R-NMF gives results similar to those of MCR-LLM, except that the relative abundances are smoothed thanks to the rational functions. When using standard NMF the result displays more noise, and features several unexpected peaks for all abundances, especially for SiO_2 (blue curve). Using splines or polynomials also leads to some noise but with a lower level than standard NMF. However our lack of knowledge in chemistry does not allow us to clearly determine the best method between MCR-LLM and R-NMF.

For the next two-dimensional dataset [8], we reshape all images into vectors to form the input matrix. This time, the dataset contains the binding energy (instead of the energy loss), which resembles rational functions more. Furthermore relative abundances are computed in a 2D space, so that we cannot properly represent them with univariate rational functions. Therefore, we use rational functions to represent the binding energy, with degrees $d_1 = d_2 = 44$, and report again on the best out of ten tests in Figure 5.

All methods are able to discriminate the three categories C_0, C_1 and C_2 in the abundance maps, and there is no noticeable difference between the three approaches using NMF. This time, NMF methods obtain abundances that appear less discriminate than MCR-LLM. Note however that other state-of-the-art methods presented in [8] are unable to distinguish between the three categories, so that our method still makes sense in this case.

Recall that in the 1D-case we could use the smoothness of changes in relative abundances to improve the results. To do the same for the 2D-case would require to represent relative abundances with smooth bivariate functions. This would result in a functional NMF model involving (possibly multivariate) functions in both factors A and X.

V. Conclusion

We have introduced R-NMF, a factorization model based on rational functions to unmix discretized signals. When comparing with standard NMF or with NMF over polynomials or splines, we found that the use of rational functions outperforms existing methods in terms of denoising and reconstruction ability, at the cost of an increase in computational time. This better reconstruction may be due to the wider range of representation of rational functions as well as their ability to present an essentially unique factorization. Our algorithm is also able to obtain results similar to specialized state of the art algorithms for spectral unmixing of multicomponent nanostructures. We also saw from that application that, for some datasets, both factors possess exploitable structural properties. Investigation of this bi-functional NMF model, as well as the inclusion of multivariate signals in the factors, is left for future work.

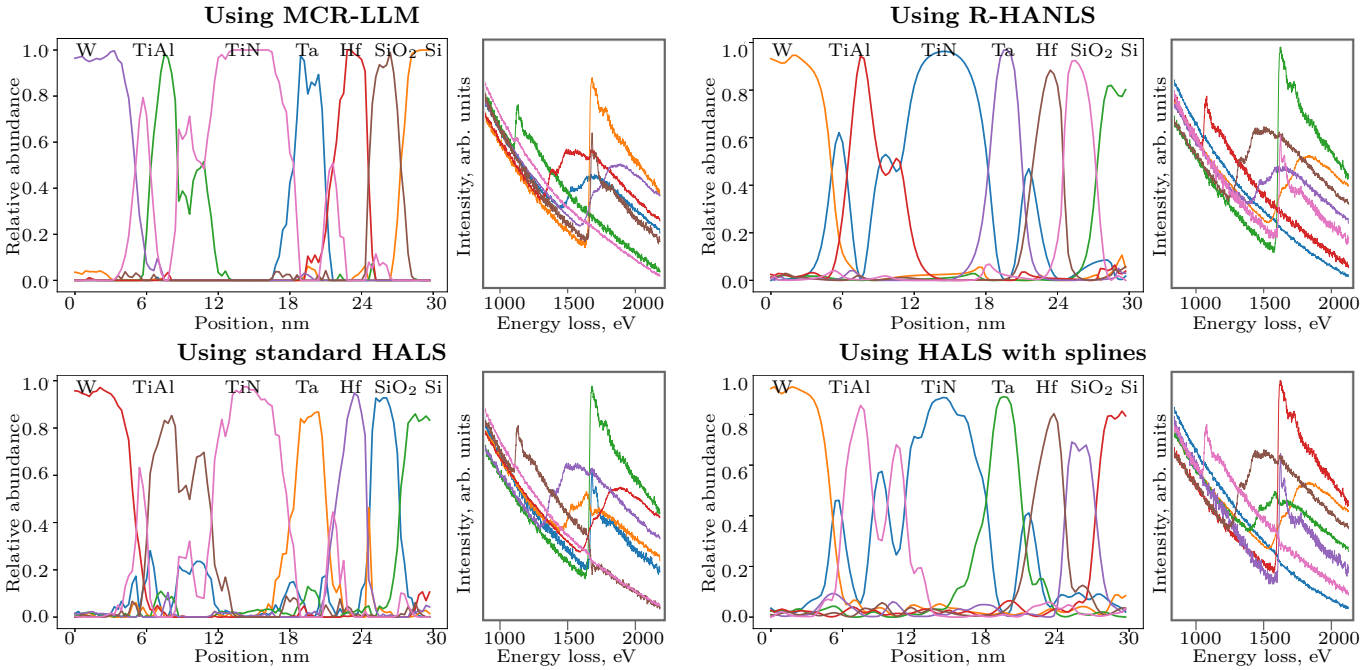


Fig. 4. Results of different methods on 1D dataset. Result obtained using polynomials were close to the ones obtained with splines.

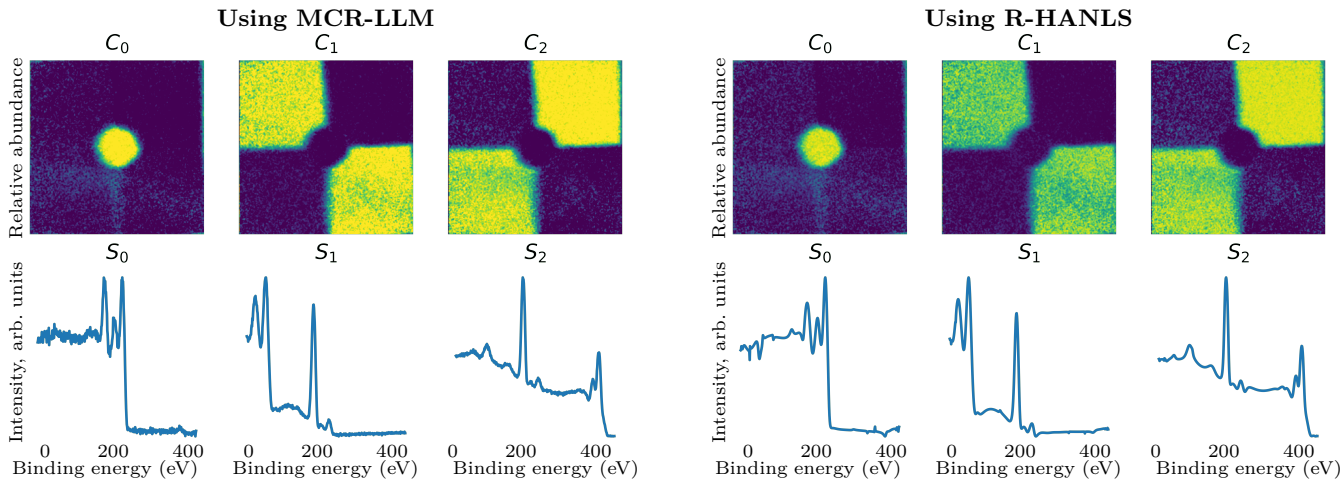


Fig. 5. Results on 2D dataset. Results obtained using splines, polynomials and standard HALS were very close to results of R-HANLS.

References

[1] Nadi Braidy and Ryan Gosselin. Unmixing noisy co-registered spectrum images of multicomponent nanostructures. *Scientific reports*, 9(1):1–8, 2019.

[2] Andrzej Cichocki, Rafal Zdunek, and Shun-ichi Amari. Hierarchical ALS algorithms for nonnegative matrix and 3D tensor factorization. In *International Conference on Independent Component Analysis and Signal Separation*, pages 169–176. Springer, 2007.

[3] Andrzej Cichocki, Rafal Zdunek, Anh Huy Phan, and Shun-ichi Amari. *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation*. John Wiley & Sons, 2009.

[4] Otto Debals, Marc Van Barel, and Lieven De Lathauwer. Löwner-based blind signal separation of rational functions with applications. *IEEE Transactions on Signal Processing*, 64(8):1909–1918, 2015.

[5] Nicolas Gillis. The why and how of nonnegative matrix factorization. *Regularization, Optimization, Kernels, and Support Vector Machines*, 12(257), 2014.

[6] Cécile Hautecoeur and François Glineur. Nonnegative matrix factorization over continuous signals using parametrizable functions. *Neurocomputing*, 2020.

[7] R Kokaly and al. USGS spectral library version 7, 2017.

[8] Francis B Lavoie, Nadi Braidy, and Ryan Gosselin. Including noise characteristics in mcr to improve mapping and component extraction from spectral images. *Chemometrics and Intelligent Laboratory Systems*, 153:40–50, 2016.

[9] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788, 1999.

[10] Yuji Nakatsukasa, Olivier Sète, and Lloyd N Trefethen. The AAA algorithm for rational approximation. *SIAM Journal on Scientific Computing*, 40(3):A1494–A1522, 2018.

[11] Victoria Powers and Bruce Reznick. Polynomials that are positive on an interval. *Transactions of the American Mathematical Society*, 352(10):4677–4692, 2000.

[12] Lloyd N Trefethen. *Approximation Theory and Approximation Practice, Extended Edition*. SIAM, 2019.