Journal of
Imaging

Article

Spatially Coherent Clustering Based on Orthogonal Nonnegative Matrix Factorization

Pascal Fernsel

Center for Industrial Mathematics, University of Bremen, 28359 Bremen, Germany; p.fernsel@uni-bremen.de;
Tel.: +49-(0)421-218-63814

Abstract: Classical approaches in cluster analysis are typically based on a feature space analysis. However, many applications lead to datasets with additional spatial information and a ground truth with spatially coherent classes, which will not necessarily be reconstructed well by standard clustering methods. Motivated by applications in hyperspectral imaging, we introduce in this work clustering models based on Orthogonal Nonnegative Matrix Factorization (ONMF), which include an additional Total Variation (TV) regularization procedure on the cluster membership matrix to enforce the needed spatial coherence in the clusters. We propose several approaches with different optimization techniques, where the TV regularization is either performed as a subsequent post-processing step or included into the clustering algorithm. Finally, we provide a numerical evaluation of 12 different TV regularized ONMF methods on a hyperspectral dataset obtained from a matrix-assisted laser desorption/ionization imaging measurement, which leads to significantly better clustering results compared to classical clustering models.

Keywords: orthogonal nonnegative matrix factorization; clustering; spatial coherence; hyperspectral data; MALDI imaging

Citation: Fernsel, P. Spatially Coherent Clustering Based on Orthogonal Nonnegative Matrix Factorization. *J. Imaging* **2021**, *7*, 194. https://doi.org/10.3390/jimaging7100194

Academic Editors: Fabiana Zama and Elena Loli Piccolomini

Received: 19 August 2021
Accepted: 23 September 2021
Published: 28 September 2021

1. Introduction

Cluster analysis has been studied over the past fifty years in the machine learning community and is one of the central topics in unsupervised learning with a wide range of possible research directions and application fields, including image segmentation, document clustering, and bioinformatics [1]. The general clustering problem is to partition a given set of objects $\mathcal{O}$ into different groups, such that the objects within a group are more similar to each other compared to the objects in other groups. One typical approach is based on a feature space analysis. The basic concept is to assign to each object $\sigma \in \mathcal{O}$ a feature vector $x_\sigma \in \mathcal{X}$ containing the characteristics of σ, where X is a suitably defined feature space. Furthermore, a similarity measure and a suitable minimization problem is defined to introduce the notion of similarity between the feature vectors and to formulate the clustering problem.

However, many types of datasets contain additional spatial information, which is typically not used in a classical cluster analysis. Characteristic examples are images or, more generally, hyperspectral datasets, where each measured spectrum is associated to a point in a two- or three-dimensional space. Furthermore, many application fields, such as mass spectrometry imaging or Earth remote sensing, naturally lead to datasets with spatially coherent regions. Hence, a classical cluster analysis, which is entirely based on a feature space analysis, does not lead necessarily to spatially coherent clusters and is, therefore, not sufficient to reconstruct the coherent regions in these kind of data.

Hence, this work focuses on a combined clustering analysis, which takes into account both the feature space and the spatial coherence of the clusters. We introduce numerous clustering methods based on ONMF for general nonnegative datasets with spatial information and include a TV regularization procedure to regularize the cluster membership matrix to induce the needed spatial coherence. Furthermore, we discuss different optimization techniques for the ONMF models and derive the corresponding minimization algorithms. Finally, we perform a numerical evaluation on a mass spectrometry imaging dataset acquired from a matrix-assisted laser desorption/ionization imaging measurement of a human colon tissue sample and compare the proposed clustering methods to classical ONMF approaches.

This paper is organized as follows. After a short description of the related work and the used notation in Sections 1.1 and 1.2, we give a brief outline of the basics of ONMF approaches, its relations to K-means clustering, and details on possible solution algorithms in Section 2. In Section 3, we introduce the proposed methods in this work, which are divided into so-called separated methods and combined methods. Section 4 is entirely devoted to the numerical experiments and the evaluation of the discussed methods. Finally, Section 5 concludes the findings and gives an outlook for future possible research directions.

1.1. Related Work

The natural relation between ONMF and clustering models is well studied. One of the first theoretical analysis was provided by Ding et al. in Reference [2]. By comparing the cost functions of different Nonnegative Matrix Factorization (NMF) and K-means models, the authors could show in their work, for example, the strong relationship between K-means clustering and ONMF with an orthogonality constraint on one of the factorization matrices, as well as kernel K-means and symmetric ONMF. Furthermore, the connections to spectral clustering and tri-factorizations were studied. Several works followed with a similar theoretical emphasis on tri-factorizations [3] and multiple other NMF models [4].

The previous mentioned works focus on the theoretical side but also give some first update rules to solve the corresponding ONMF problems. However, more work has been done for the algorithm development. Many classical approaches are based on multiplicative update rules [3,5–7]. More recent works are, for instance, based on nuclear norm optimization [8], further multiplicative update schemes [9,10], hierarchical alternating least squares [11–13], and proximal alternating linearized minimization [14], together with the very recent work of Reference [15], EM, such as algorithms and augmented Lagrangian approaches [16], deep neural networks [17], and other techniques [18,19]. Very recently, Reference [20] developed a block coordinate descent-based projected gradient algorithm specifically for solving ONMF problems. Regarding nonnegative matrix tri-factorizations, the very recent work of Reference [21] introduces a block inertial Bregman proximal algorithm for general nonsmooth and non-convex problems and applies it to the application case of symmetric nonnegative matrix tri-factorization. Furthermore, the authors in Reference [22] develop four algorithms for the specific problem case of symmetric multi-type nonnegative matrix tri-factorization comprising a fixed-point method, block-coordinate descent with projected gradient, and a gradient method with exact line search, as well as an adaptive moment estimation approach. Finally, we would like to refer the interested reader at this point to the two review articles on NMF methods for clustering by References [23,24] and a book on NMF by Reference [25].

While analyzing and developing optimization algorithms for NMF clustering methods is a major topic throughout the literature, studying clustering techniques with spatial coherence by incorporating the local information of the considered data points is far less common. This topic is primarily analyzed in the context of image segmentation problems [26–29]. The subject of spatial coherence can also be found in the literature of hyperspectral image analysis. Several NMF models with total variation regularization, which include the local neighborhood information of each data point, have been analyzed for the critical processing step of hyperspectral unmixing [30–32]. These articles can be considered as closest to our approach, since we also focus on the application of generalized NMF models to hyperspectral images. Further works also consider NMF models with different TV penalty terms for hyperspectral image denoising [33–35], some of which will be used in the later course of this work for the derivation of optimization algorithms of the respective clustering models. Reference [36] uses a variant of the NMF, namely the nonnegative matrix underapproximation, which solves the NMF by identifying localized and sparse features sequentially. Similar to the considered approach in this work, the authors introduce a sparsity constraint, make use of the fact that it is more likely that neighboring pixels are contained in the same features, and apply the approach to hyperspectral datasets.

However, all the aforementioned works on ONMF or TV regularized NMF only include either orthogonality constraints or TV regularization into their NMF models, whereas we focus on combining both of these properties to obtain a spatially coherent clustering method for hyperspectral datasets.

The only work, which includes TV regularization, as well as penalty terms to enforce an orthogonality constraint on one of the matrices, is, to the best of our knowledge, the survey article of Reference [37], with a rather general focus on the development of algorithms based on surrogate functions leading to multiplicative update schemes.

1.2. Notation

Matrices will play a major role throughout this work and are denoted, unless otherwise stated, by capital Latin or Greek letters (e.g., $X, U, \Psi, \dots$). The entry of a matrix U in the i-th row and j-th column is indicated as U_{ij}. The same holds true for a matrix product, where its ij-th entry is given by $(UV)_{ij}$. Furthermore, we use a dot to indicate rows and columns of matrices. The i-th row and j-th column of U are written as $U_{i,\bullet}$ and $U_{\bullet,j}$, respectively. Moreover, we denote the i-th iteration of a matrix U in an algorithm by $U^{[i]}$.

Moreover, we write $\|U\|_F$ and $\|U_{\bullet,j}\|_2$ for the Frobenius norm of a matrix U and the usual Euclidean norm of a vector $U_{\bullet,j}$. We also use the notion of nonnegative matrices and write $U \geq 0$ or $U \in \mathbb{R}_{\geq 0}^{m \times n}$ with $\mathbb{R}_{\geq 0} := \{x \in \mathbb{R} \mid x \geq 0\}$ for an $m \times n$ matrix U, which has only nonnegative entries. The notation for the dimension of the matrices in the NMF problems are reused throughout the article and will be introduced in the following, Section 2.

2. Background

2.1. Orthogonal NMF and K-Means

Nonnegative Matrix Factorization (NMF), originally introduced by Paatero and Tapper in 1994 as positive matrix factorization [38], is a specific matrix factorization method designed to obtain a low-rank approximation of a given and typically large nonnegative data matrix. Different from the widely used Principal Component Analysis (PCA), which is based on the singular value decomposition and allows for computation of a best rank K approximation of a given arbitrary matrix, the NMF constrains the matrix factors to be nonnegative. This property makes the NMF the method of choice where the considered data naturally fulfills a nonnegativity constraint so that the interpretability of the factor matrices is ensured. NMF has been widely used for data compression, source separation, feature extraction, clustering, or even for solving inverse problems. Possible application fields are hyperspectral unmixing [30–32], document clustering [8,39], and music analysis [40] but also medical imaging problems, such as dynamic computed tomography, to perform a joint reconstruction and low-rank decomposition of the corresponding dynamic inverse problem [41], or Matrix-Assisted Laser Desorption/Ionization (MALDI) imaging, where it can be used for tumor typing in the field of bioinformatics as a supervised classification method [42].

Mathematically, the standard NMF problem can be formulated as follows: For a given nonnegative matrix $X \in \mathbb{R}_{\geq 0}^{M \times N}$, the task is to find two nonnegative matrices $U \in \mathbb{R}_{\geq 0}^{M \times K}$ and $V \in \mathbb{R}_{\geq 0}^{K \times N}$ with $K \ll \min\{M, N\}$, such that

$$X \approx UV = \sum_{k=1}^{K} U_{\bullet,k} V_{k,\bullet}.$$

This allows the approximation of the columns $X_{\bullet,n}$ and rows $X_{m,\bullet}$ via a superposition of just a few basis vectors $\{U_{\bullet,k}\}_k$ and $\{V_{k,\bullet}\}_k$, such that $X_{\bullet,n} \approx \sum_k V_{kn} U_{\bullet,k}$ and $X_{m,\bullet} \approx \sum_k U_{mk} V_{k,\bullet}$. In this way, the NMF can be seen as a basis learning tool with additional nonnegativity constraints.

The typical variational approach to tackle the NMF problem is to reformulate it as a minimization problem by defining a suitable discrepancy term $\mathcal{D}$ according to the noise assumption of the underlying problem. The default case of Gaussian noise corresponds to the Frobenius norm on which we will focus on in this work. Further possible choices include the Kullback–Leibler divergence or more generalized divergence functions [25].

Moreover, NMF problems are non-linear and ill-conditioned [43,44]. Thus, they require stabilization techniques, which is typically done by adding regularization terms $\mathcal{R}_j$ into the NMF cost function $\mathcal{F}$. However, besides the use case of regularization, the penalty terms $\mathcal{R}_j$ can be also used to enforce additional properties to the factorization matrices U and V. The general NMF minimization problem can, therefore, be written as

$$\min_{U,V \geq 0} \mathcal{D}(X, UV) + \sum_{j=1}^{J} \alpha_j \mathcal{R}_j(U, V) =: \min_{U,V \geq 0} \mathcal{F}(U, V), \tag{1}$$

where α_j are regularization parameters. Common choices for $\mathcal{R}_j$ are ℓ_1 and ℓ_2 penalty terms. Further possible options are total variation regularization or other penalty terms which enforce orthogonality or even allow a supervised classification scheme in case the NMF is used as a prior feature extraction step [37,42]. In this work, we will focus on the combination of orthogonality constraints and total variation penalty terms to construct an NMF model for spatially coherent clustering methods.

Another essential step for computing the NMF of a given matrix X is the determination of an optimal number of features K. Typical techniques used throughout the literature are based on heuristic or approximative methods, including core consistency diagnostics via a PCA or residual analysis (also see Reference [25]). A straightforward technique used in Reference [45] is based on the analysis of the rank-one matrices $U_{\bullet,k} V_{k,\bullet}$. For a $K \in \mathbb{N}$ chosen sufficiently large, the considered NMF algorithm is executed to obtain the factorization UV. Afterwards, the norm of the rank-one matrices $U_{\bullet,k} V_{k,\bullet}$ for every $k \in \{1, \ldots, K\}$ is analyzed. By the choice of a large K, the NMF algorithm is forced to compute additional irrelevant features, which can be identified by a small norm of the corresponding rank-one matrices. By choosing a suitably defined threshold for the values of the norm, a suitable K can be obtained.

This work, however, will not focus on methods to determine an optimal number of features. Hence, we assume in the numerical part in Section 4 that the true number of features in the considered dataset is known in advance so that K can be set a priori.

On the other hand, K-means clustering is one of the most commonly used prototype-based, partitional clustering technique. As for any other clustering method, the main task is to partition a given set of objects into groups such that objects in the same group are more similar compared to the ones in other groups. These groups are usually referred to as clusters. In mathematical terms, the aim is to partition the index set $\{1, 2, \ldots, M\}$ of a corresponding given dataset $\{x_m \in \mathbb{R}^N \mid m = 1, \ldots, M\}$ into disjoint sets $\mathcal{I}_k \subset \{1, \ldots, M\}$, such that $\cup_{k=1,\ldots,K} \mathcal{I}_k = \{1, \ldots, M\}$.

Many different variations and generalizations of K-means have been proposed and analyzed (see, for instance, References [1,46] and the references therein), but we will focus in this section on the most common case. The method is based on two main ingredients. On the one hand, a similarity measure $\operatorname{dist}(\cdot, \cdot)$ is needed to specify the similarity between data points. The default choice is the squared Euclidean distance $\operatorname{dist}(x_i, x_j) := \|x_i - x_j\|_2^2$. On the other hand, so-called representative centroids $c_k \in \mathbb{R}^N$ are computed for each cluster $\mathcal{I}_k$. The computation of the clusters and centroids is based on the minimization of the within-cluster variances given by $\mathcal{J} = \sum_{k=1}^{K} \sum_{m \in \mathcal{I}_k} \operatorname{dist}(x_m, c_k)$. Due to the NP-hardness of the minimization problem [47], heuristic approaches are commonly used to find an approximate solution. The K-means algorithm is the most common optimization technique which is based on an alternating minimization. After a suitable initialization, the first step is to assign each data point x_m to the cluster with the closest centroid with respect to the distance measure $\operatorname{dist}(\cdot,\cdot)$. In the case of the squared Euclidean distance, the centroids are recalculated in a second step based on the mean of its newly assigned data points to minimize the sum of the within-cluster variances. Both steps are repeated until the assignments do not change anymore.

The relationship between the NMF and K-means clustering can be easily seen by adding further constraints to both problems. First of all, from the point of view of K-means, we assume nonnegativity of the given data and write the vectors x_m row-wise to a data matrix X such that $X = [x_1, \dots, x_M]^\intercal \in \mathbb{R}_{\geq 0}^{M \times N}$. Furthermore, we define the so-called cluster membership matrix $\tilde{U} \in \{0,1\}^{M \times K}$, such that

$$\tilde{U}_{mk} := \begin{cases} 0 & \text{if } \ m \notin \mathcal{I}_k \\ 1 & \text{if } \ m \in \mathcal{I}_k \end{cases}$$

and the centroid matrix $\tilde{V} := [c_1, \dots, c_K]^\intercal \in \mathbb{R}_{\geq 0}^{K \times N}$. With this, and by choosing the squared Euclidean distance function, it can be easily shown that the objective function $\mathcal{J}$ of K-means can be rewritten as $\mathcal{J} = \|X - \tilde{U}\tilde{V}\|_F^2$, which has the same structure of the usual cost function of an NMF problem. However, the usual NMF does not constrain one of the matrices to have binary entries or, more importantly, to be row-orthogonal as it is the case for $\tilde{U}$. This ensures that each row of $\tilde{U}$ contains only one nonzero element which gives the needed clustering interpretability.

This gives rise to the problem of ONMF, which is given by

$$\min_{U,V \geq 0} \mathcal{F}(U,V), \ \text{ s.t. } \ U^\intercal U = I,$$

where I is the identity matrix. The matrices U and V of an ONMF problem will be henceforth also referred to as cluster membership matrix and centroid matrix, respectively. In the case of the Frobenius norm as discrepancy term and without any regularization terms $\mathcal{R}_j$, it can be shown that this problem is equivalent to weighted variant of the spherical K-means problem [16]. For further variants of the relations between ONMF and K-means, we refer to the works of References [2–4] and the review articles of References [23,24].

2.2. Algorithms for Orthogonal NMF

Due to the ill-posedness of NMF problems and possible constraints on the matrices, tailored minimization approaches are needed. In this section, we review shortly common optimization techniques of NMF and ONMF problems, which will also be used in this work for the derivation of algorithms for ONMF models, including spatial coherence.

For usual choices of $\mathcal{D}$ and $\mathcal{R}_j$ in the NMF problem (1), the corresponding cost function $\mathcal{F}$ is convex in each of the variables U and V but non-convex in (U,V). Therefore, the majority of optimization algorithms for NMF and ONMF problems are based on alternating minimization schemes

$$U^{[i+1]} = \arg\min_{U \geq 0} \mathcal{F}(U, V^{[i]}), \tag{2}$$

$$V^{[i+1]} = \arg\min_{V \geq 0} \mathcal{F}(U^{[i+1]}, V). \tag{3}$$

One classical technique to tackle these minimization problems are alternating multiplicative algorithms, which only consist of summations and multiplications of matrices and, therefore, ensure the nonnegativity of U and V without any additional projection step provided that they are initialized appropriately. This approach was mainly popularized by the works of Lee and Seung [48,49], which also brought much attention to the NMF, in general. The update rules are usually derived by analyzing the Karush–Kuhn–Tucker

(KKT) first-order optimality conditions for each of the minimization problems in (2) and (3) or via the so-called Majorize-Minimization (MM) principle. The basic idea of the latter technique is to replace the NMF cost function $\mathcal{F}$ by a majorizing surrogate function $\mathcal{Q}_{\mathcal{F}} : \mathrm{dom}(\mathcal{F}) \times \mathrm{dom}(\mathcal{F}) \to \mathbb{R}$, which is easier to minimize and whose tailored construction leads to the desired multiplicative updates rules defined by

$$A^{[i+1]} := \arg \min_{A \in \mathrm{dom}(\mathcal{F})} \mathcal{Q}_{\mathcal{F}}(A, A^{[i]}).$$

with the defining properties of a surrogate function that $\mathcal{Q}_{\mathcal{F}}$ majorizes $\mathcal{F}$ and $\mathcal{Q}_{\mathcal{F}}(A, A) = \mathcal{F}(A)$ for all $A \in \mathrm{dom}(\mathcal{F})$, it can be easily shown that the above update rule leads to a monotone decrease of the cost function $\mathcal{F}$ (also see Appendix A). However, the whole method is based on an appropriate construction of the surrogate functions, which is generally non-trivial. Possible techniques for common choices of $\mathcal{D}$ and $\mathcal{R}_j$ in the NMF cost function are based on the quadratic upper bound principle and Jensen's inequality [37]. Overall, multiplicative algorithms offer a flexible approach to various choices of NMF cost functions and will also be used in this work for some of the proposed and comparative methods.

Another classical method is based on Alternating (nonnegative) Least Squares (ALS) algorithms. They are based on the estimation of the stationary points of the cost function with a corresponding fixed point approach and a subsequent projection step to ensure the nonnegativity of the matrices. An extension to this procedure is given by Hierarchical Alternating (nonnegative) Least Squares (HALS) algorithms, which solve nonnegative ALS problems column-wise for both matrices U and V [11,12] and will also be used as a comparative methodology.

An optimization approach, which was recently used for NMF problems, is the widely known Proximal Alternating Linearized Minimization (PALM) [14,15], together with its extensions, including stochastic gradients [50]. As a first step, the cost function is split up into a differentiable part $\mathcal{F}_1$ and a non-differentiable part $\mathcal{F}_2$. In its basic form, the PALM update rules consist of alternating gradient descent steps for U and V with learning rates based on the Lipschitz constants of the gradient of $\mathcal{F}_1$ in combination with a subsequent computation of a proximal operator of the function $\mathcal{F}_2$. Some of these techniques will be used for the proposed methods in this work and will be discussed in more detail in Section 3.

Further well-known techniques are, for example, projected gradient algorithms consisting of additive update rules, quasi-Newton approaches based on second-order derivatives of the cost function, or algorithms based on an augmented Lagrangian concept [16,25].

All these methods can be also used to derive suitable algorithms for ONMF problems. Common approaches to include the orthogonality constraints are the use of Lagrangian multipliers [3,5,6,11,12] or the replacement of the hard constraint $U^\intercal U = I$ by adding a suitable penalty term into the NMF cost function to enforce approximate row-orthogonality for U controlled by a regularization parameter [7,9,10,15,20,37]. Other methods include optimization algorithms on the Stiefel manifold [19], the use of sparsity and nuclear norm minimization [8,39], or other techniques [14,16,18].

In the next section, we will introduce the considered NMF models in this work and derive the corresponding optimization algorithms.

3. Orthogonal NMF with Spatial Coherence

In this section, we present the spatially coherent clustering methods based on ONMF models, together with the derivation of their optimization algorithms. Different from classical clustering approaches via ONMF, our proposed technique includes the local information of a measured data point into the clustering process. This is done by including a TV regularization procedure on the cluster membership matrix U, which naturally leads to spatially coherent regions, while preserving their edges.

This is, for instance, especially helpful for hyperspectral datasets. If the neighborhood of a measured spectrum $X_{m,\bullet}$ is associated to one cluster $\mathcal{I}_k$, the inclusion of spatial coherence in the clustering model makes it more likely that $X_{m,\bullet}$ will be also classified to $\mathcal{I}_k$. This spatial coherence can be observed in many spectral imaging applications, such as Earth remote sensing or MALDI imaging, where locally close spectra have a higher probability to belong to the same class.

In the following, we divide our proposed techniques into so-called separated and combined methods.

3.1. Separated Methods

One straightforward approach to design a spatially coherent clustering method is to compute a clustering based on a classical ONMF model and subsequently perform a post-processing on the obtained cluster membership matrix based on total variation denoising. The general workflow is provided in Algorithm 1 and will be henceforth referred to as `ONMF-TV`.

Algorithm 1 `ONMF-TV`

```
1: Input: $X \in \mathbb{R}_{\geq 0}^{M \times N}$, $K \in \mathbb{N}$, $\tau > 0$, $\theta \in \mathbb{R}^J$
2: Initialize: $U^{[0]} \in \mathbb{R}_{\geq 0}^{M \times K}$, $V^{[0]} \in \mathbb{R}_{\geq 0}^{K \times N}$
3: $(U, V) \leftarrow \mathrm{ONMF}_\theta(X, U^{[0]}, V^{[0]})$
4: $U \leftarrow \textsc{TVDenoiser}_\tau(U)$
```

After a suitable initialization of the cluster membership matrix and the centroid matrix in Line 2, a classical ONMF model is used in Line 3 to perform a clustering of the given data X, where θ are possible hyperparameters which typically have to be chosen a priori. These ONMF models will be chosen based on some of the works described in Section 2.2 and will be specified in more detail in Section 4.2. Afterwards, a TV denoising algorithm is applied on the cluster membership matrix U obtained by ONMF_θ to induce the spatial coherence in the clustering. In this work and for all considered separated methods, the denoising step is based on evaluating the proximal mapping column-wise on U, which is defined by the minimization problem

$$\mathrm{prox}_{\tau \|\cdot\|_{\mathrm{TV}}}(x) := \underset{y \in \mathbb{R}^M}{\arg\min} \left\{ \frac{1}{2} \|y - x\|^2 + \tau \|y\|_{\mathrm{TV}} \right\}. \tag{4}$$

Here, $\tau > 0$ is a regularization parameter, and $\| \cdot \|_{\mathrm{TV}}$ denotes the classical isotropic TV [51,52] and corresponds to the definition in (6) with $\varepsilon_{\mathrm{TV}} = 0$ if $\| \cdot \|_{\mathrm{TV}}$ is applied on a matrix U. The algorithm used to solve the above minimization problem is based on a Fast Iterative Shrinkage/Thresholding Algorithm (FISTA) described in Reference [52], with a maximal iteration number of 100. Afterwards, every negative entry of U is projected to zero to ensure the corresponding nonnegativity constraint. For further details on the implementation of the TV denoising algorithm and the separated methods, in general, we refer the reader to the provided codes of all considered algorithms in our GitLab [53].

We consider this workflow as baseline for our comparison with the combined methods presented in the following Section 3.2. For the later numerical evaluation of the separated methods in Section 4.4, we will compare these approaches with and without the TV post-processing step to get an impression of the advantage of adding a TV regularization procedure to the clustering method.

The initialization methods, the stopping criteria and the choice of the hyperparameters will be specified in more detail in Section 4 of the numerical experiments of this work.

3.2. Combined Methods

In this section, we present the so-called combined methods, together with different optimization algorithms for their numerical solution. Different from the separated methods in Section 3.1, this coupled approach includes a total variation penalty term into the ONMF model to induce the spatial coherence in the clustering. The combination of orthogonality constraints and a total variation penalty term in the NMF problem leads, in general, to a higher effort to derive suitable solution algorithms. However, the main motivation is that this joint workflow allows the clustering process to take advantage of the TV regularization, leading to better local minima, which could, therefore, enhance the quality of the clustering compared to classical approaches or the previous described separated methods, where the spatial coherence is just enforced in an independent subsequent TV denoising step.

In the following, we will present different multiplicative update rules and algorithms based on proximal gradient descent approaches.

As for the separated methods, the initialization and the stopping criteria, as well as the choice of the hyperparameters, for all considered approaches will be described in Section 4 in more detail.

3.2.1. Multiplicative Update Rules

`ONMFTV-MUL1`

Our first multiplicative algorithm is taken from the work of Reference [37] without any modification and is based on the ONMF model

$$\min_{U,V\geq 0} \frac{1}{2}\|X-UV\|_F^2 + \frac{\sigma}{2}\|I-U^\intercal U\|_F^2 + \frac{\tau}{2}\|U\|_{\mathrm{TV}_{\varepsilon_{\mathrm{TV}}}}, \tag{5}$$

where $\sigma, \tau \geq 0$ are regularization parameters, and $\|\cdot\|_{\mathrm{TV}_{\varepsilon_{\mathrm{TV}}}}$ is the smoothed, isotropic total variation [37,54] defined by

$$\|U\|_{\mathrm{TV}_{\varepsilon_{\mathrm{TV}}}} := \sum_{k=1}^{K}\sum_{m=1}^{M}|\nabla_{mk}U| := \sum_{k=1}^{K}\sum_{m=1}^{M}\sqrt{\varepsilon_{\mathrm{TV}}^2 + \sum_{\tilde{m}\in\mathcal{N}_m}(U_{mk}-U_{\tilde{m}k})^2}. \tag{6}$$

Furthermore, $\varepsilon_{\mathrm{TV}} > 0$ is a positive, small, predefined constant to ensure the differentiability of the TV penalty term, which is needed due to the MM principle for the optimization approach. Finally, $\mathcal{N}_m$ are index sets referring to spatially neighboring pixels. The default case in two dimensions for the neighborhood of a non-boundary pixel in (i,j) is $\mathcal{N}_{(i,j)} = \{(i+1,j),(i,j+1)\}$ to obtain an estimate of the gradient components in both directions.

The derivation of the solution algorithm is described in Reference [37] and is based on the MM principle mentioned in Section 2.2. The surrogate function $\mathcal{Q}_{\mathcal{F}}(x,a)$ of such approaches majorizes $\mathcal{F}$ and is typically quadratic in x. In order to avoid complications in constructing a suitable surrogate function for the cost function in (5), the fourth order terms from $\|I-U^\intercal U\|_F^2$ have to be avoided. In Reference [37], this problem is solved by introducing an auxiliary variable $W \in \mathbb{R}_{\geq 0}^{M\times K}$ and by reformulating the minimization problem in (5) as

$$\min_{U,V,W\geq 0} \underbrace{\frac{1}{2}\|X-UV\|_F^2 + \frac{\sigma_1}{2}\|I-W^\intercal U\|_F^2 + \frac{\sigma_2}{2}\|W-U\|_F^2 + \frac{\tau}{2}\|U\|_{\mathrm{TV}_{\varepsilon_{\mathrm{TV}}}}}_{=:\mathcal{F}(U,V,W)}. \tag{7}$$

For this problem, a suitable surrogate function and a multiplicative algorithm can be derived. The details of the derivation will not be discussed here and can be found in Reference [37] in all details. We also provide a short outline of the derivation in Appendix A.

In the following, we describe the final update rules obtained by the MM principle in Algorithm 2 and define for this purpose the matrices $P(U), Z(U) \in \mathbb{R}_{\geq 0}^{M \times K}$ as

$$P(U)_{mk} := \frac{\sum_{\tilde{m} \in \mathcal{N}_m} 1}{|\nabla_{mk} U|} + \sum_{\tilde{m} \in \bar{\mathcal{N}}_m} \frac{1}{|\nabla_{\tilde{m}k} U|}, \tag{8}$$

$$Z(U)_{mk} := \frac{1}{P(U)_{mk}} \left(\frac{1}{|\nabla_{mk} U|} \sum_{\tilde{m} \in \mathcal{N}_m} \frac{U_{mk} + U_{\tilde{m}k}}{2} + \sum_{\tilde{m} \in \bar{\mathcal{N}}_m} \frac{U_{mk} + U_{\tilde{m}k}}{2|\nabla_{\tilde{m}k} U|} \right), \tag{9}$$

where $\bar{\mathcal{N}}_m$ is the so-called adjoint neighborhood given by $\tilde{m} \in \bar{\mathcal{N}}_m \Leftrightarrow m \in \mathcal{N}_{\tilde{m}}$.

Algorithm 2 `ONMFTV-MUL1`

```
Input $X \in \mathbb{R}_{\geq 0}^{M \times N},\ K \in \mathbb{N},\ \sigma_1, \sigma_2, \tau > 0,\ i = 0$
Initialize $U^{[0]}, W^{[0]} \in \mathbb{R}_{>0}^{M \times K},\ V^{[0]} \in \mathbb{R}_{>0}^{K \times N}$
repeat
    $U^{[i+1]} = \left[ U^{[i]} \circ \left( \frac{X V^{[i]\intercal} + \tau P(U^{[i]}) \circ Z(U^{[i]}) + (\sigma_1 + \sigma_2) W^{[i]}}{\tau P(U^{[i]}) \circ U^{[i]} + \sigma_2 U^{[i]} + U^{[i]} V^{[i]} V^{[i]\intercal} + \sigma_1 W^{[i]} W^{[i]\intercal} U^{[i]}} \right) \right]_{>0}$
    $V^{[i+1]} = \left[ V^{[i]} \circ \left( \frac{U^{[i+1]\intercal} X}{U^{[i+1]\intercal} U^{[i+1]} V^{[i]}} \right) \right]_{>0}$
    $W^{[i+1]} = \left[ W^{[i]} \circ \left( \frac{(\sigma_1 + \sigma_2) U^{[i+1]}}{\sigma_1 U^{[i+1]} U^{[i+1]\intercal} W^{[i]} + \sigma_2 W^{[i]}} \right) \right]_{>0}$
    $i \leftarrow i + 1$
until Stopping criterion satisfied
```

We denote by $\circ$, as well as the fraction line, the element-wise (Hadamard) multiplication and division, respectively. Due to the multiplicative structure of the update rules, the nonnegativity of the iterates is preserved. However, a strict positive initialization of U, V, and W is needed to avoid numerical instabilities and the zero locking phenomenon caused by zero entries in the matrices, which is characteristic for all algorithms based on multiplicative update rules (see, e.g., Reference [9]). For the same reason, we perform a subsequent element-wise projection step for every matrix defined by $[\lambda]_{>0} := \max\{\lambda, \varepsilon_{P_1}\}$ with $\varepsilon_{P_1} = 1 \times 10^{-16}$. Analogously, a projection step is applied for too large entries, with $\varepsilon_{P_2} = 1 \times 10^{35}$ being the corresponding parameter.

The asymptotic computational complexity of `ONMFTV-MUL1` can be easily obtained by analyzing the performed matrix multiplications and the involved for-loops in the algorithm (also see Reference [53]), which leads to a complexity of $\mathcal{O}(KMN + K^2N + K^2M)$. However, K is usually chosen such that $K \ll \min\{M, N\}$. Hence, regarding the asymptotic computational complexity, we can consider K as a positive constant, so that we obtain $\mathcal{O}(MN)$.

Finally, the above algorithm ensures a monotone decrease of the cost function in (7) due to its construction based on the MM principle [37]. This leads to the convergence of the cost function values, since $\mathcal{F}(U, V, W)$ is bounded from below.

Theorem 1 (`ONMFTV-MUL1`)**.** *Algorithm 2 ensures a monotone decrease of the cost function defined by the NMF model in (7).*

`ONMFTV-MUL2`

In this section, we derive another multiplicative algorithm, following the ideas in the work of Reference [35], based on a continuous formulation of an isotropic and differentiable version of the TV penalty term given by

$$\mathrm{TV}_{\varepsilon_{\mathrm{TV}}}(u) := \int_{\Omega} \|\nabla u\|_{\varepsilon_{\mathrm{TV}}} \, \mathrm{d}(x_1, x_2) \tag{10}$$

for $\Omega \subset \mathbb{R}^2$, a sufficiently smooth $u : \Omega \to \mathbb{R}$ with bounded variation (see, e.g., Reference [51]), and for

$$\|\nabla u\|_{\varepsilon_{\mathrm{TV}}} := \sqrt{\left(\frac{\partial u}{\partial x_1}\right)^2 + \left(\frac{\partial u}{\partial x_2}\right)^2 + \varepsilon_{\mathrm{TV}}^2}, \tag{11}$$

with a small $\varepsilon_{\mathrm{TV}} > 0$. The application of the TV regularization on the discrete matrix U is done via a subsequent discretization step, which is specified in more detail in Appendix B. Thus, we consider in this section the orthogonal NMF model

$$\min_{U,V\geq 0} \underbrace{\frac{1}{2}\|X - UV\|_F^2 + \frac{\sigma_1}{4}\|I - U^\intercal U\|_F^2 + \tau\,\mathrm{TV}_{\varepsilon_{\mathrm{TV}}}(U)}_{=:\mathcal{F}(U,V)}, \tag{12}$$

with the sloppy notation of $\mathrm{TV}_{\varepsilon_{\mathrm{TV}}}(U)$ for the matrix U, where the discretization step is implicitly included. Different from the model `ONMFTV-MUL1` in the previous section, we do not include any auxiliary variable W.

The update rule for U is based on a classical gradient descent approach

$$U^{[i+1]} := U^{[i]} - \Gamma^{[i]} \circ \nabla_U \mathcal{F}(U^{[i]}, V^{[i]}), \tag{13}$$

with a step size $\Gamma^{[i]} \in \mathbb{R}_{\geq 0}^{M\times K}$. By computing the gradient $\nabla_U \mathcal{F}(U^{[i]}, V^{[i]})$ and choosing an appropriate step size $\Gamma^{[i]}$, this leads to the multiplicative update rule shown in Algorithm 3. For the minimization with respect to V, we simply choose the multiplicative update rule given in Algorithm 2. The term $\mathrm{div}\left(\nabla U^{[i]} / \|\nabla U^{[i]}\|_{\varepsilon_{\mathrm{TV}}}\right)$ is again an abuse of notation and implicitly includes a discretization step (see Appendix B.2). It can be seen as the gradient of the TV penalty term and is obtained by analyzing the corresponding Euler-Lagrange equation (see Appendix B.1). Note that, in discretized form, it is a matrix of size $M \times K$ and can also contain negative entries. Hence, this update step is, strictly speaking, not a multiplicative update, since it cannot enforce the nonnegativity of the matrix U by itself. However, as in Algorithm 2, the projection step given by $[\cdot]_{>0}$ is applied subsequently for both matrices to avoid numerical instabilities and to ensure the nonnegativity of U. Different from the approach in `ONMFTV-MUL1`, a monotone decrease of the cost function cannot be guaranteed based on the MM principle, since $\mathcal{F}$ also contains fourth-order terms due to the penalty term $\|I - U^\intercal U\|_F^2$.

Algorithm 3 `ONMFTV-MUL2`

```
Input  $X \in \mathbb{R}_{\geq 0}^{M\times N},\ K \in \mathbb{N},\ \sigma_1, \tau > 0,\ i = 0$
Initialize  $U^{[0]} \in \mathbb{R}_{>0}^{M\times K},\ V^{[0]} \in \mathbb{R}_{>0}^{K\times N}$
repeat
    $U^{[i+1]} = \left[ U^{[i]} \circ \dfrac{XV^{[i]\intercal} + \tau\,\mathrm{div}\left(\frac{\nabla U^{[i]}}{\|\nabla U^{[i]}\|_{\varepsilon_{\mathrm{TV}}}}\right) + \sigma_1 U^{[i]}}{U^{[i]}V^{[i]}V^{[i]\intercal} + \sigma_1 U^{[i]}U^{[i]\intercal}U^{[i]}} \right]_{>0}$
    $V^{[i+1]} = \left[ V^{[i]} \circ \left( \dfrac{U^{[i+1]\intercal}X}{U^{[i+1]\intercal}U^{[i+1]}V^{[i]}} \right) \right]_{>0}$
    $i \leftarrow i + 1$
until Stopping criterion satisfied
```

By analyzing the involved matrix multiplications in Algorithm 3, we can see that the asymptotic computational complexity is the same as in `ONMFTV-MUL1`. Hence, we obtain for `ONMFTV-MUL2` $\mathcal{O}(KMN + K^2N + K^2M)$.

Furthermore, note the similarity of the update rules given in Reference [9], where no total variation penalty term is considered. For more details on the derivation of Algorithm 3 and the discretization of the divergence term, we refer the reader to Appendix B.

3.2.2. Proximal Alternating Linearized Minimization

Adapting the optimization procedure of the very recent work of Reference [50], we consider in this section several Proximal Alternating Linearized Minimization (PALM) schemes. For all presented methods in this section, we consider the NMF model

$$\min_{U,V,W\geq 0} \underbrace{\frac{1}{2}\|X-UV\|_F^2+\frac{\sigma_1}{2}\|I-W^\intercal U\|_F^2+\frac{\sigma_2}{2}\|W-U\|_F^2}_{=:\mathcal{F}(U,V,W)}+\tau\underbrace{\|U\|_{\mathrm{TV}}}_{=:\mathcal{J}(U)}, \tag{14}$$

where $\|\cdot\|_{\mathrm{TV}}$ is defined according to (4). One crucial step for the application of this optimization approach is to divide the whole cost function into a differentiable part $\mathcal{F}$ and a non-differentiable part $\mathcal{J}$. The additional auxiliary variable W is needed to ensure the Lipschitz continuity of the partial gradients of $\mathcal{F}$ and, hence, the convergence rates of the respective algorithms shown in Reference [50].

The general scheme of these algorithms are based on an alternating minimization procedure with a gradient descent step with respect to the differentiable function $\mathcal{F}(U,V,W)$ via the computation of full gradients or gradient estimates and a subsequent column-wise application of the proximal operator for the non-differentiable part of the ONMF model. In the case of the algorithm for U, the proximal operator with respect to the function $\mathcal{J}$ is evaluated, which leads to the update rule

$$U_{\bullet,k}^{[i+1]} := \left[\operatorname{prox}_{\tau\eta\mathcal{J}}\left(U_{\bullet,k}^{[i]}-\eta(\tilde{\nabla}_U\mathcal{F}(U^{[i]},V,W))_{\bullet,k}\right)\right]_{\geq 0} \tag{15}$$

for a suitable step size $\eta > 0$. Furthermore, $\tilde{\nabla}_U\mathcal{F}$ is either the partial derivative $\nabla_U\mathcal{F}$ of $\mathcal{F}$ with respect to U or some random gradient estimate (also see `ONMFTV-SPRING`). The nonnegativity constraint of the matrices is ensured by a final projection of all negative values to zero denoted in (15) by $[\cdot]_{\geq 0}$. In the following, we will write in short

$$U^{[i+1]} := \left[\operatorname{prox}_{\tau\eta\mathcal{J}}\left(U^{[i]}-\eta\tilde{\nabla}_U\mathcal{F}(U^{[i]},V,W)\right)\right]_{\geq 0}$$

for the whole matrix U. The evaluation of the proximal mapping for all considered methods based on the PALM scheme in this section is based on the same FISTA algorithm [52] as it is the case for the separated approaches of Section 3.1, whereby the maximal iteration number is reduced to 5. As usual for PALM algorithms, adaptive step sizes based on the local Lipschitz constants of the partial gradients of $\mathcal{F}$ are used, which will be approximated via the power iteration for all considered approaches. More information on the computation of these estimates and the choice of the step sizes are given in the subsequent descriptions of the specific algorithms and in Section 4 of the numerical experiments, as well as in Appendix C. Details on the derivation of the gradients and the Lipschitz constants for the specific NMF model in (14) are given in Appendix C.

Regarding the update rules for the matrices V and W for the considered ONMF model in (14), the application of the proximal operator can be neglected, since there is no non-differentiable part in (14) depending on either V or W. Hence, only the corresponding gradient descent and projection steps are performed for both matrices (see Algorithms 4 and 5).

`ONMFTV-PALM`

The proximal alternating linearized minimization is based on the general algorithm scheme dscribed above. The main steps are illustrated in Algorithm 4. For the gradient descent step, the full classical partial derivatives $\nabla_U\mathcal{F}$, $\nabla_V\mathcal{F}$, and $\nabla_W\mathcal{F}$ are computed without the consideration of any batch sizes. Furthermore, the function PowerIt denotes the power method to compute the needed step sizes. The derivation of the algorithm, together with the computation of the gradients and the Lipschitz constants for the step sizes, are described in Appendix C.1. The algorithm terminates until a suitable stopping criterion is satisfied, which will be further specified in Section 4 and Appendix D. This approach will be henceforth referred to as `ONMFTV-PALM`.

As for the multiplicative algorithms `ONMFTV-MUL1` and `ONMFTV-MUL2`, one can easily obtain the asymptotic computational complexity for Algorithm 4 by analyzing the involved matrix multiplications (also see Reference [53]), leading to the same complexity $\mathcal{O}(KMN + K^2N + K^2M)$ as for proposed multiplicative update rules.

Algorithm 4 `ONMFTV-PALM`

```
Input $X \in \mathbb{R}_{\geq 0}^{M \times N}$,  $K \in \mathbb{N}$,  $\sigma_1, \sigma_2, \tau > 0$,  $i = 0$
Initialize $U^{[0]}, W^{[0]} \in \mathbb{R}_{\geq 0}^{M \times K}$,  $V^{[0]} \in \mathbb{R}_{\geq 0}^{K \times N}$
repeat
    $\eta_{U^{[i]}} = \textsc{PowerIt}_U(V^{[i]}, W^{[i]})$
    $U^{[i+1]} = \left[\mathrm{prox}_{\tau \eta_{U^{[i]}} \mathcal{J}}\left(U^{[i]} - \eta_{U^{[i]}} \nabla_U \mathcal{F}(U^{[i]}, V^{[i]}, W^{[i]})\right)\right]_{\geq 0}$
    $\eta_{V^{[i]}} = \textsc{PowerIt}_V(U^{[i+1]})$
    $V^{[i+1]} = \left[V^{[i]} - \eta_{V^{[i]}} \nabla_V \mathcal{F}(U^{[i+1]}, V^{[i]}, W^{[i]})\right]_{\geq 0}$
    $\eta_{W^{[i]}} = \textsc{PowerIt}_W(U^{[i+1]})$
    $W^{[i+1]} = \left[W^{[i]} - \eta_{W^{[i]}} \nabla_W \mathcal{F}(U^{[i+1]}, V^{[i+1]}, W^{[i]})\right]_{\geq 0}$
    $i \leftarrow i + 1$
until Stopping criterion satisfied
```

`ONMFTV-iPALM`

A slightly extended version of `ONMFTV-PALM` is the so-called Inertial Proximal Alternating Linearized Minimization (iPALM) algorithm, which introduces an additional momentum term into Algorithm 4 and, hence, improves the convergence rate. Since the iPALM algorithm still follows the rough outline of `ONMFTV-PALM` and uses the classical partial gradients of $\mathcal{F}$, we will not present the whole algorithm at this point and refer the reader to the corresponding work in Reference [55], Appendix C.2, and our GitLab [53], where the corresponding codes are available online. This method will be referred to as `ONMFTV-iPALM`.

Since `ONMFTV-iPALM` follows the same major principles as `ONMFTV-PALM`, the asymptotic computational complexity of `ONMFTV-iPALM` is the same as in the case of `ONMFTV-PALM`.

`ONMFTV-SPRING`

The so-called Stochastic Proximal Alternating Linearized Minimization (SPRING) is an extended version of the PALM method, where the full gradients are replaced by random estimates. One basic assumption of this approach is that $\mathcal{F}$ is separable depending on the considered variable for which the cost function is minimized. In the case of the minimization with respect to U, the function $\mathcal{F}$ can be expressed as

$$\mathcal{F}(U, V, W) = \sum_{n=1}^{N} \frac{1}{2} \|X_{\bullet,n} - UV_{\bullet,n}\|_2^2 + \underbrace{\sum_{k=1}^{K} \frac{\sigma_1}{2} \|I_{\bullet,k} - U^\intercal W_{\bullet,k}\|_2^2 + \frac{\sigma_2}{2} \|U_{\bullet,k} - W_{\bullet,k}\|_2^2}_{=:\tilde{\mathcal{F}}(U,W)}.$$

However, instead, we use the formulation

$$\mathcal{F}(U, V, W) = \sum_{n=1}^{N} \left[\frac{1}{2} \|X_{\bullet,n} - UV_{\bullet,n}\|_2^2 + \frac{1}{N} \tilde{\mathcal{F}}(U, W)\right] =: \sum_{n=1}^{N} \mathcal{F}_n(U, V, W)$$

to be able to compute the estimates of the gradients based on the functions $\mathcal{F}_n$. These random estimates are formed by using just a few indices of $\mathcal{F}_n$, which are the elements of the so-called mini-batch $n \in \mathcal{B}_{i,j}^{U} \subset \{1, \ldots, N\}$, where i denotes the iteration number of the

SPRING algorithm, and $j \in \{1, \dots, s_r\}$ specifying the currently used subsample of indices, with $1/s_r$ being the subsample ratio. One classical example of a gradient estimator, which is also used in this work, is the Stochastic Gradient Descent (SGD) estimator given by

$$\tilde{\nabla}_U^{i,j} \mathcal{F}(U, V, W) := \sum_{n \in \mathcal{B}_{i,j}^U} \nabla_U \mathcal{F}_n(U, V, W). \tag{16}$$

For other possible gradient estimators, such as the SAGA or SARAH estimator, we refer the reader to the work of Reference [50] and the references therein.

The case of the minimization with respect to V and W is more straightforward. For V, the penalty terms for the orthogonality can be dropped, and we use the separability of the function with respect to the rows of X and U. The computation of the SGD estimator is done analogously. For minimizing with respect to W, the expression based on $\tilde{\mathcal{F}}(U, W)$ can be used by omitting the data fidelity term. However, since $K \ll \min\{M, N\}$, we use in this work the full gradient of $\mathcal{F}$ for the minimization with respect to W.

The main steps of the algorithm are presented in Algorithm 5, which will be referred to as `ONMFTV-SPRING`. For details on the choice of the hyperparameters and the computation of the gradients, as well as the step sizes, we refer the reader to Section 4 and Appendix C.

Algorithm 5 `ONMFTV-SPRING`

```
Input $X \in \mathbb{R}_{\geq 0}^{M \times N}, \ K \in \mathbb{N}, \ \sigma_1, \sigma_2, \tau > 0, \ s_r \in \mathbb{N}, \ i = 0$
Initialize $U^{[0,1]}, W^{[0,1]} \in \mathbb{R}_{\geq 0}^{M \times K}, \ V^{[0,1]} \in \mathbb{R}_{\geq 0}^{K \times N}$
repeat
    for $j = 1, \dots, s_r$ do
        $\eta_{U^{[i,j]}} = \textsc{PowerIt}_U(V^{[i,j]}, W^{[i,j]})$
        $U^{[i,j+1]} = \left[\operatorname{prox}_{\tau \eta_{U^{[i,j]}} \mathcal{J}}\left(U^{[i,j]} - \eta_{U^{[i,j]}} \tilde{\nabla}_U^{i,j} \mathcal{F}(U^{[i,j]}, V^{[i,j]}, W^{[i,j]})\right)\right]_{\geq 0}$
        $\eta_{V^{[i,j]}} = \textsc{PowerIt}_V(U^{[i,j+1]})$
        $V^{[i,j+1]} = \left[V^{[i,j]} - \eta_{V^{[i,j]}} \tilde{\nabla}_V^{i,j} \mathcal{F}(U^{[i,j+1]}, V^{[i.j]}, W^{[i,j]})\right]_{\geq 0}$
        $\eta_{W^{[i,j]}} = \textsc{PowerIt}_W(U^{[i,j+1]})$
        $W^{[i,j+1]} = \left[W^{[i,j]} - \eta_{W^{[i,j]}} \nabla_W \mathcal{F}(U^{[i,j+1]}, V^{[i,j+1]}, W^{[i,j]})\right]_{\geq 0}$
    end for
    $U^{[i+1,1]} = U^{[i,s_r+1]}, \ V^{[i+1,1]} = V^{[i,s_r+1]}, \ W^{[i+1,1]} = W^{[i,s_r+1]}$
    $i \leftarrow i + 1$
until Stopping criterion satisfied
```

The asymptotic computational complexity of `ONMFTV-SPRING` can be obtained by analyzing the involved for-loops and matrix multiplications (see Reference [53]), leading to a complexity of $\mathcal{O}(KMN + s_r K^2 N + s_r K^2 M)$, which now also involves s_r. However, note that s_r can be also seen as a positive constant, which in this case would lead to the same complexity as in `ONMFTV-PALM`.

Convergence Analysis

While this work does not focus on a theoretical convergence analysis of the considered optimization methods of the ONMF models, we provide in the following some brief information on the convergence properties of the PALM algorithms.

The convergence theory of PALM algorithms has been analyzed in various works [50,55,56]. The convergence properties depend heavily on the properties of the considered cost function. A non-standard extension of the usual PALM scheme in the case of our considered ONMF model in (14) is the introduction of a third auxiliary variable. However, according to References [50,56], the full convergence theory of the PALM and SPRING scheme easily extends to an arbitrary number of variables.

Furthermore, $\mathcal{F}$ and $\mathcal{J}$ in (14) have to satisfy specific properties to ensure basic convergence properties of the PALM algorithms. $\mathcal{F}$ needs to be a finite-valued, differentiable function with its gradient $\nabla\mathcal{F}$ being Lipschitz continuous on bounded sets of $\mathbb{R}^{M\times K} \times \mathbb{R}^{K\times N} \times \mathbb{R}^{M\times K}$. This is obviously true in the case of (14) as a consequence of the mean value theorem, since $\mathcal{F}$ is a C^2 function. Furthermore, the partial gradients $\nabla_U\mathcal{F}, \nabla_V\mathcal{F}$, and $\nabla_W\mathcal{F}$ need to be Lipschitz continuous with modulus L_U, L_V, and L_W, respectively, which is also true in our considered ONMF model (also see Appendix C).

Regarding the non-differentiable part in (14), $\mathcal{J}$ needs to be a proper lower semi-continuous function, which is bounded from below. Since the classic TV semi-norm used in (14) is lower semi-continuous (see Reference [51]), these properties also hold in our case.

To ensure the specific convergence property that a sequence of iterates $Y^{[k]} := (U^{[k]}, V^{[k]}, W^{[k]})$ converge to a critical point $Y^* := (U^*, V^*, W^*)$ of the whole cost function $\mathcal{G}(U,V,W) := \mathcal{F}(U,V,W) + \mathcal{J}(U)$ in (14), additional requirements on $\mathcal{G}$ are needed. In the case of `ONMFTV-PALM`, $\mathcal{G}$ needs to be a so-called Kurdyka-Łojasiewicz (KL) function (see References [50,56]), whereas, for `ONMFTV-SPRING`, $\mathcal{G}$ has to be a semialgebraic function [50]. For a full-length treatise of the definition and properties of semialgebraic functions in the field of real algebraic geometry, we refer the reader to Reference [57]. The following Lemma is a nonsmooth version of the Łojasiewicz gradient inequality and can be found in Reference [56].

Lemma 1. *Let $f : \mathbb{R}^n \to (-\infty, +\infty]$ be a proper and lower semicontinuous function. If σ is semialgebraic, then σ fulfills the KL property at any point in the domain of σ and is, hence, a KL function.*

Since $\mathcal{G}$ is obviously a proper and lower semicontinuous function, it is sufficient to show that $\mathcal{G}$ is semialgebraic. To do so, we note, first of all, that $\mathcal{F}(U,V,W)$ is a real polynomial function and, hence, semialgebraic. By using further basic properties of semialgebraic functions, it is also easy to see that $\mathcal{J}(U)$ is semialgebraic. Since $\mathcal{J}$ is given by the equation in (6) with $\varepsilon_{\mathrm{TV}} = 0$, it consists of a real polynomial function with a subsequent application of the square root function, which is still semialgebraic. Since finite sums of semialgebraic functions are semialgebraic, $\mathcal{G}$ is semialgebraic. Hence, the sequence of iterates $Y^{[k]}$ produced by `ONMFTV-PALM` converge to a critical point of $\mathcal{G}$. In the case of `ONMFTV-SPRING`, the iterates converge in expectation to a critical point.

Similar arguments with additional assumptions on the momentum terms can be applied for `ONMFTV-iPALM`, so that we also achieve a convergence to a critical point for this algorithm. While the authors in Reference [55] do not prove improved convergence rates, they show that the additional momentum terms lead to an improved performance in practice. For further results on the convergence rates of the considered algorithms, we refer the reader to the aforementioned references.

4. Numerical Experiments

4.1. Dataset

Concerning the numerical experiments of this work, we consider a hyperspectral dataset obtained from a Matrix-Assisted Laser Desorption/Ionization (MALDI) imaging measurement of a human colon tissue sample [58]. This technique is a Mass Spectrometry Imaging (MALDI-MSI) method, which is able to provide a spatial molecular profile of a given analyte. Together with the technological advancements in acquisition speed and robustness over the last decade, MALDI-MSI has become a standard tool in proteomics and applications in medical research, such as characterization of tumor subtypes and extraction of characteristic spectra [42], have become feasible.

In general, a measurement with a mass spectrometer can be subdivided into three main steps: the ionization of the sample, followed by the separation and detection of the ions. Considering MALDI imaging, the ionization part is characterized by the term MALDI. Different from other ionization techniques, such as the so-called Secondary-Ion

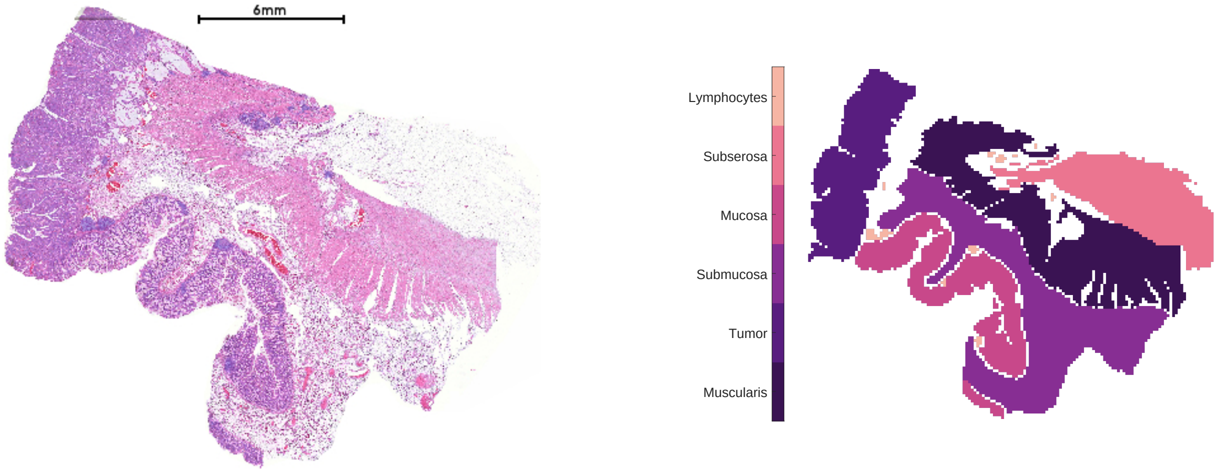
Mass Spectrometry (SIMS) or Desorption Electrospray Ionization (DESI), MALDI allows measurement of a wider mass range of the extracted ions. This is possible due to an application of a matrix onto the tissue sample to optimize the transfer of the ionization energy and to extract the molecules out of the analyte. The needed ionization energy is provided by a laser, which is shot on the tissue sample by following a grid pattern. The separation of the molecules is done by a mass analyzer. One typical method, which was also used for the dataset in this work, is to accelerate the ions by an electric field and to measure the Time-Of-Flight (TOF) of the particles. The final step of the whole measurement is the detection of the ions with the help of an ion detector.

Another major aspect of the whole MALDI-MSI workflow is the preparation of the analytes before the actual measurement with a MALDI-TOF device, which follows standardized protocols. For more information, we refer the reader to References [58,59].

For every point on the tissue slide, which is shot by the laser, a whole mass spectrum is acquired, leading to a hyperspectral dataset. The whole data is then written into the matrix $X \in \mathbb{R}_{\geq 0}^{M \times N}$, where each entry denotes the intensity of the detected particles of a specific mass-to-charge ratio (m/z-value). Typically, the measured spectra are ordered row-wise to the matrix, such that every column corresponds to an intensity plot of the whole tissue slide for a specific m/z-value (m/z-image).

Typical data sizes range from 1×10^4 to one million spectra and m/z-images, respectively. Furthermore, MALDI datasets are naturally nonnegative. These properties make the NMF an ideal analyzing tool for MALDI imaging datasets due to the nonnegativity constraints leading to a meaningful physical interpretation of the acquired matrices U and V. For more details on the application of NMF models to MALDI imaging problems and the interpretation of the factorization matrices, we refer the reader to the works of References [37,42].

Figure 1 shows the human colon tissue dataset used for the subsequent numerical evaluation of the methods. Figure 1a is the histological image after the application of a so-called Hematoxylin and Eosin (H&E) staining, which allows a distinction between different tissue types. Figure 1b shows the histological annotation, which divides the dataset into six different classes and constitutes the ground truth for the subsequent numerical evaluation. The human colon dataset consists of 12,049 acquired spectra, each containing 20,000 measured m/z values covering a mass range of 600 Da to 4000 Da. However, we only consider the actual annotated spectra to ensure that each considered spectra can be reasonably classified in one of the 6 classes shown in Figure 1b. Hence, we restrict ourselves to 8725 spectra leaving out the zero and the non-annotated ones, which leads to the nonnegative data matrix $X \in \mathbb{R}_{\geq 0}^{8725 \times 20{,}000}$ for the numerical experiments.



(**a**) Histological image after H&E staining (**b**) Ground truth

Figure 1. Histological image of the considered MALDI dataset after H&E staining (**a**) and the histological annotation (**b**).

4.2. Choice of Separated Methods

As discussed in Section 3.1, we consider the separated approaches based on the workflow given in Algorithm 1 as baseline for the comparison with the combined methods presented in Section 3.2. In this section, we specify shortly seven ONMF algorithms to compute the clustering in Line 3 of Algorithm 1, which are based on different works throughout the literature and will be used for the numerical comparison in Section 4.4 (see Table 1). Numerical tests for the optimization approach in the more recent work of Reference [8], which is based on a sparsity and nuclear norm minimization, did not lead to satisfactory results and, hence, will not be presented in this work.

Furthermore, we will evaluate the separated methods with and without the TV denoising step in Section 4.4. In such a way, we obtain a comparison between the clustering results of the classical ONMF approaches in Table 1 with and without the TV post-processing and by that an intuitive view on the advantage of adding a TV regularization procedure to the clustering method.

The choice of the hyperparameters and details on the initialization of the matrices, as well as the used stopping criteria of all separated methods, are discussed in Section 4.3 and Appendix D.1.

Table 1. Designations of the considered separated methods (left column) and short explanation of the corresponding ONMF algorithm (right column).

Separated Method	Description
`K-means-TV`	Classical K-means clustering algorithm.
`ONMF-TV-Choi`	Alternating multiplicative update rules based on Reference [5].
`ONMF-TV-Ding`	Alternating multiplicative update rules based on Reference [3].
`ONMF-TV-Pompili1`	Alternating expectation-maximization algorithm similar to the default spherical K-means algorithm [16,60].
`ONMF-TV-Pompili2`	Alternating algorithm based on an augmented Lagrangian approach with strict orthogonality constraints [16]. Nonnegativity is obtained asymptotically by using a quadratic penalty.
`ONMF-TV-Kimura`	Hierarchical alternating least squares algorithm, which is applied column-wise on U and row-wise on V [11].
`ONMF-TV-Li`	Hierarchical alternating least squares algorithm with approximate orthogonality constraints and subsequent projection steps to ensure nonnegativity [12].

4.3. Setup

In this section, we describe the initialization methods and stopping criteria, the calculation of the final hard clustering, the choice of the various hyperparameters, and the used cluster validation measures. Furthermore, we give some further details on the general numerical setup.

For every considered separated and combined method in Sections 3.1 and 3.2, we perform 30 replicates of the experiment to get an impression of the performance stability, since the used initialization methods are partially based on randomly selected data points or matrices. For each method, we use the same random seed in the beginning of the experiment. Furthermore, we measure for each replicate the computational time, including the time for the calculation of the initialization of the factorization matrices.

For the evaluation of the clusterings, we use several different clustering validation measures discussed in Reference [61]. Due to the known ground truth of the data, we restrict ourselves to external clustering validation measures. Based on the results in this work, we primarily consider the so-called normalized Van Dongen criterion ($\mathrm{VD_n}$) and the normalized Variation of Information ($\mathrm{VI_n}$), since they give the most representative, quantitative measures in most of the general cases. Furthermore, they are normalized into the $[0,1]$ range and give reasonable results in cases there are clusters without any corresponding data points, which is different from classical measures, such as the purity. In addition, we consider as a secondary measure the Entropy (E). For all considered measures, it holds that a lower value indicates a better clustering performance.

To provide the definition of the clustering validation measures, we denote $n_{k\tilde{k}}$ as the number of data points in cluster $\mathcal{I}_k$ from class $\mathcal{C}_{\tilde{k}}$ for $k, \tilde{k} \in \{1, \ldots, K\}$ and define

$$n := \sum_{k,\tilde{k}=1}^{K} n_{k\tilde{k}}, \qquad n_{k,\bullet} := \sum_{\tilde{k}=1}^{K} n_{k\tilde{k}}, \qquad n_{\bullet,\tilde{k}} := \sum_{k=1}^{K} n_{k\tilde{k}},$$
$$p_{k\tilde{k}} := \frac{n_{k\tilde{k}}}{n}, \qquad p_k := \frac{n_{k,\bullet}}{n}, \qquad \tilde{p}_{\tilde{k}} := \frac{n_{\bullet,\tilde{k}}}{n}.$$

Using this notation, the definition of all considered clustering validation measures are given in Table 2.

Table 2. Definitions of all considered clustering validation measures.

Measure	Definition	Range
Entropy (E)	$-\sum_{k=1}^{K} p_k \left(\sum_{\tilde{k}=1}^{K} \frac{p_{k\tilde{k}}}{p_k} \log\left(\frac{p_{k\tilde{k}}}{p_k} \right) \right)$	$[0, \log(K)]$
Normalized Variation of Information (VI$_\mathrm{n}$)	$1 + 2 \cdot \frac{\sum_{k,\tilde{k}=1}^{K} p_{k\tilde{k}} \log(p_{k\tilde{k}}/(p_k \tilde{p}_{\tilde{k}}))}{\sum_{k=1}^{K} p_k \log(p_k) + \sum_{\tilde{k}=1}^{K} \tilde{p}_{\tilde{k}} \log(\tilde{p}_{\tilde{k}})}$	$[0, 1]$
Normalized Van Dongen criterion (VD$_\mathrm{n}$)	$\frac{2n - \sum_{k=1}^{K} \max_{\tilde{k}}\{n_{k\tilde{k}}\} - \sum_{\tilde{k}=1}^{K} \max_k\{n_{k\tilde{k}}\}}{2n - \max_k\{n_{k,\bullet}\} - \max_{\tilde{k}}\{n_{\bullet,\tilde{k}}\}}$	$[0, 1]$

Concerning the initialization approaches, we consider either the classical K-means++ method or an initialization based on the singular value decomposition of the data matrix X by following the works of References [62,63]. In short, the latter method is based on the computation of a truncated Singular Value Decomposition (SVD) of the data matrix following a Krylov approach described in Reference [62] and performing specific projection, as well as normalization steps, based on the algorithm described in Reference [63]. For further details on the initialization, we refer the reader to the provided code in the GitLab [53]. For `K-means-TV`, we use K-means++ as the typical initialization method for the K-means algorithm. To achieve the optimal results for each of the remaining proposed and comparative methods, both initialization methods are tested. The initialization method which leads to a better clustering stability and performance in terms of the VD$_\mathrm{n}$ is chosen. However, it turns out that, for all methods, except `K-means-TV`, `ONMF-TV-Ding`, and `ONMF-TV-Pompili1`, the initialization based on the SVD of X leads to better clustering results. The concrete choices for every ONMF model are described in Appendix D.

Regarding the stopping criteria, we simply set for the considered ONMF models a maximal iteration number until a sufficient convergence is reached, except for `K-means-TV`, `ONMF-TV-Pompili1`, and `ONMF-TV-Pompili2`, where we use the internal stopping criteria of the respective algorithms. For more information, we refer the reader to Appendix D and the work of Reference [16].

Another aspect is the computation of the final clustering based on the cluster membership matrix U. Most of the considered ONMF models yield a cluster membership matrix U having multiple positive entries in each row, which is related to a soft clustering. To obtain the final hard clustering, we assign every data point to the cluster, which corresponds to the column in U, where the maximal value in the row is attained. In the case that there are two or more equal entries in one row, we choose the cluster by a random choice.

A main part of the whole workflow of the numerical evaluation is the choice of the various hyperparameters of the considered ONMF models. In particular, these include the regularization parameters σ_1, σ_2, and τ of the combined methods and the parameter τ of the TV post-processing of the separated methods. For all considered methods, we perform a grid search of the corresponding parameters. An appropriate subset of the parameter space is chosen and experiments for a wide range of possible combinations of parameters are performed. For each considered method, the parameter configuration leading to the best performance stability and $\mathrm{VD_n}$ is chosen. More details and the specific selection of the hyperparameters are given in Appendix D.

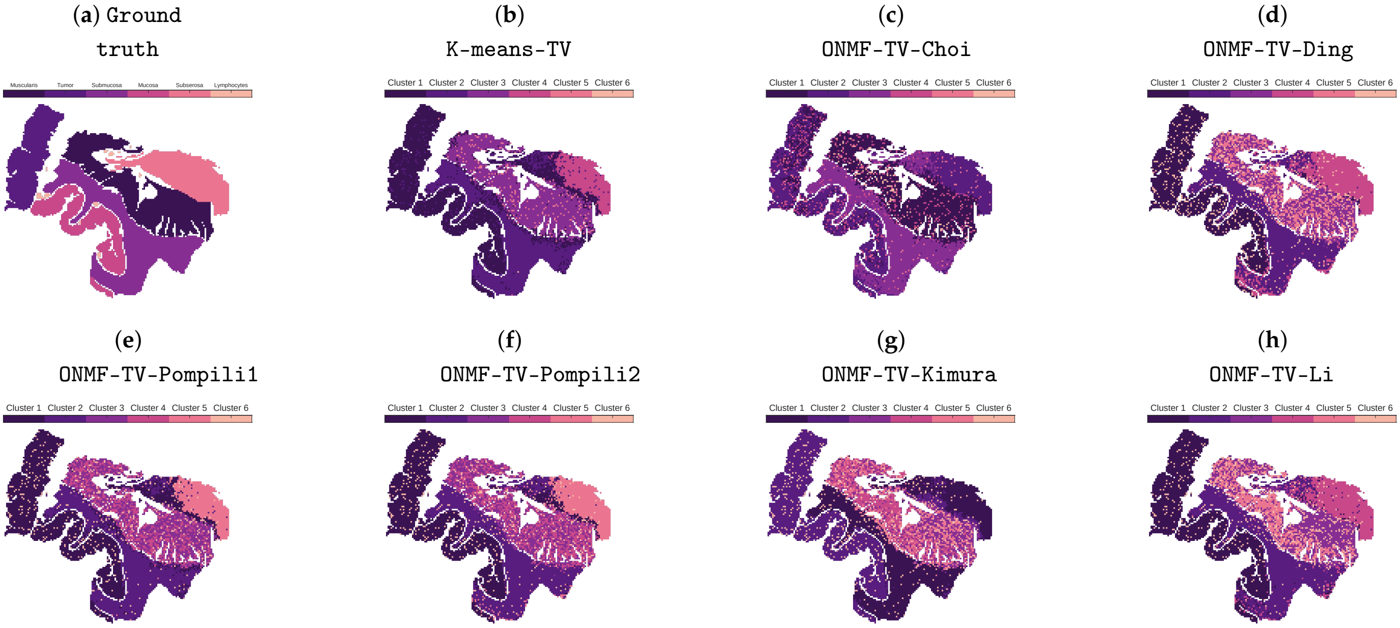
Furthermore, we perform several projection steps to enhance the numerical stability. For all considered methods, we project the data matrix X by applying $[\cdot]_{>0}$ defined as in Section 3.2.1. Furthermore, we perform specifically for the multiplicative update rules and for `ONMF-TV-Kimura` the same projection step after the initialization of the matrices. For the combined method `ONMFTV-SPRING`, we perform an additional projection step of the parameter $\tau\eta_{U^{[i,j]}}$ used for the application of the proximal operator described in Algorithm 5 to avoid too large parameters for the TV regularization. A similar projection is done for the step size in each gradient descent step. More details are given in Appendix C.3.

The algorithms were implemented with MATLAB® R2021a and executed on an Intel® Core™ i7-7700K quad core CPU @4.20 GHz with 32 GB of RAM. The corresponding MATLAB® codes can be found in our GitLab [53].

4.4. Results and Discussion

In this section, we present and discuss the numerical results obtained in the evaluation discussed in the previous sections.

Figure 2 shows the clusterings of the separated methods without the subsequent TV regularization step of the best performed replicate with respect to the $\mathrm{VD_n}$, which was obtained after the TV post-processing. In general, every separated method is able to identify all classes shown in Figure 1b, except the distinction between the tumor and mucosa. Since the combined methods are also not able to distinguish between both classes (see Figure 3), this is probably due to the fundamental, underlying NMF model with the orthogonality constraints, which are not able to identify the regions in an unsupervised workflow. In the case that annotated datasets are available for training, supervised machine learning methods could lead to different results. However, this work does not focus on these kind of approaches.


(**a**) `Ground truth`

(**b**) `K-means-TV`

(**c**) `ONMF-TV-Choi`

(**d**) `ONMF-TV-Ding`

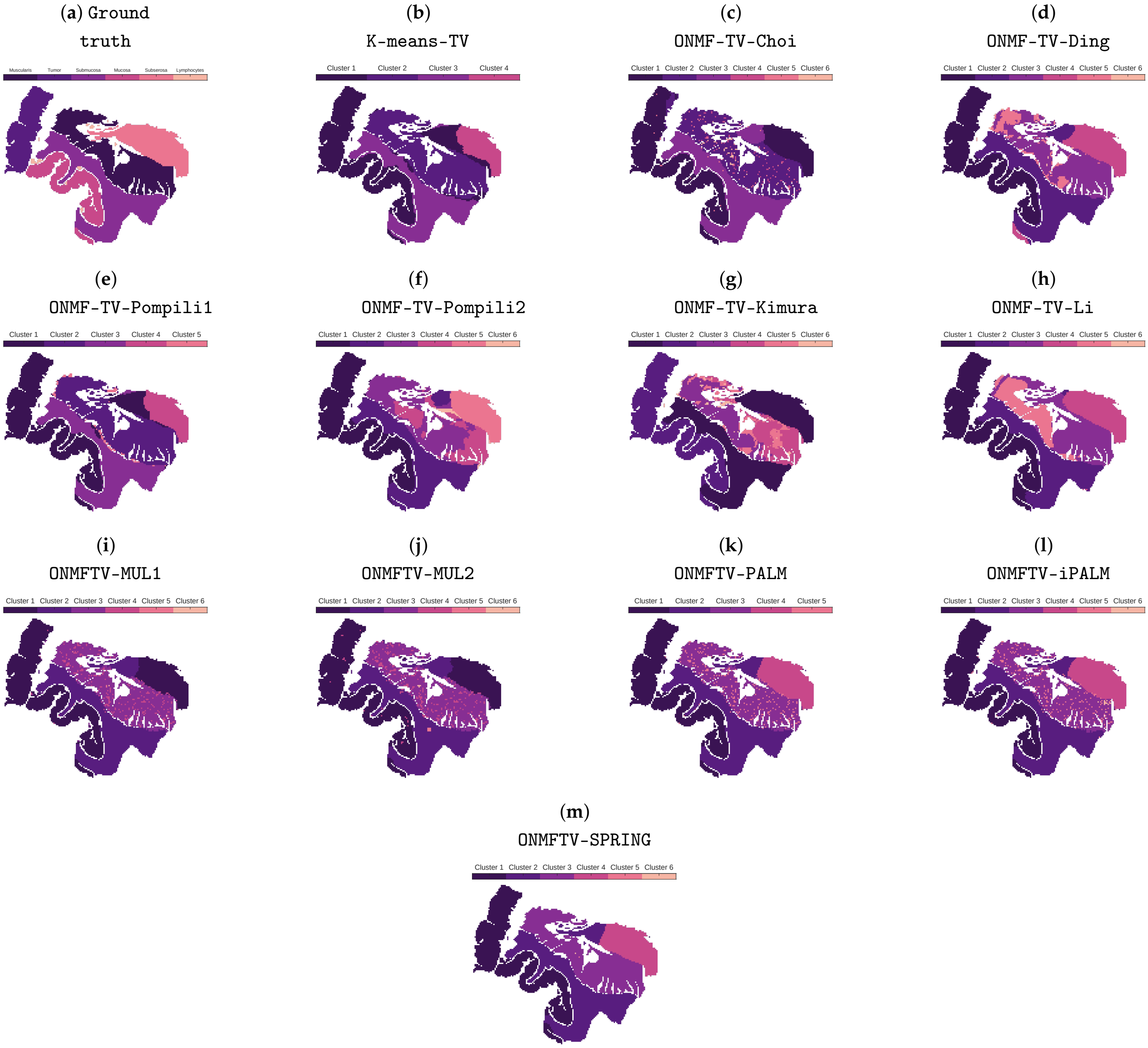
(**e**) `ONMF-TV-Pompili1`

(**f**) `ONMF-TV-Pompili2`

(**g**) `ONMF-TV-Kimura`

(**h**) `ONMF-TV-Li`

Figure 2. Ground truth and clusterings of all considered separated methods without TV regularization. The best-performing replicate, including the TV regularization, is chosen based on the normalized van Dongen criterion.



(**a**) Ground truth (**b**) K-means-TV (**c**) ONMF-TV-Choi (**d**) ONMF-TV-Ding (**e**) ONMF-TV-Pompili1 (**f**) ONMF-TV-Pompili2 (**g**) ONMF-TV-Kimura (**h**) ONMF-TV-Li (**i**) ONMFTV-MUL1 (**j**) ONMFTV-MUL2 (**k**) ONMFTV-PALM (**l**) ONMFTV-iPALM (**m**) ONMFTV-SPRING

Figure 3. Ground truth and clusterings of all considered methods, including the TV regularization. The best-performing replicate, including the TV regularization, is chosen based on the normalized van Dongen criterion.

Furthermore, every separated method is not able to guarantee a spatially coherent clustering in the region of the muscularis. Moreover, every method, except `K-means-TV`, does not provide any spatial coherence in any of the classes of the tissue slide. This is in contrast to the results in Figure 3, which shows the clusterings of all considered methods of the best performed replicate with respect to the $\mathrm{VD_n}$, including the TV regularization. Every method is able to provide a spatially coherent clustering with some few exceptions in the region of the muscularis and, hence, leads to significantly improved clusterings, in general. Comparing the results within Figure 3, the clustering of `ONMFTV-SPRING` seems to be the one, which best reproduces the given annotations. Furthermore, we note that some methods lead to clusterings with clusters, which do not contain any data points (see, i.e., `K-means-TV`, `ONMF-TV-Pompili1`, `ONMFTV-PALM`). However, this is also owing to the fact that the class of the lymphocytes is significantly underrepresented compared to the other ones (see Figure 1b). Furthermore, this behavior is also dependent on the choice of the TV regularization parameter τ.

For a quantitative evaluation, we provide in Figure 4 box plots of all replicates of every performed experiment and for all considered methods with respect to the VD_n and VI_n. For each combined method, one box plot is plotted, which visualizes the corresponding clustering validation measure of all 30 replicates of the experiment through its quartiles. As usual, the line in between the box indicates the median of the validation measure of all replicates. Furthermore, the vertical lines are the so-called whiskers, which indicate the variability outside the lower and upper quartile. Finally, each plotted point in the graph represents one replicate of the corresponding experiment indicating the obtained validation measure.

For every separated method, two box plots are plotted in different colors, each representing the obtained clustering validation measure with or without the TV post-processing (see the legend). In such a way, a comparison of the classical ONMF algorithms of Table 1 with and without the total variation regularization is possible.

Based on such a visualization, a quantitative evaluation and comparison of all considered methods by considering different clustering validation measures is possible. Furthermore, the box plots visualize the performance variability of every method and, hence, give some indications on the stability of the approaches. As described in Section 4.3, lower values of the VD_n and VI_n indicate better clustering results.

For both measures, the observations of the qualitative evaluation above can be confirmed. First, we note that, for all separated methods, the TV post-processing does indeed lead to clusterings with better cluster validation measures. Moreover, the combined methods based on the PALM scheme achieve the best results with respect to both measures, from which `ONMFTV-PALM` and `ONMFTV-iPALM` achieve the highest performance stability. While some experiments of `ONMFTV-SPRING` attain the best values compared to all other methods, this approach is less stable than the non stochastic approaches `ONMFTV-PALM` and `ONMFTV-iPALM`. Furthermore, we note that both combined methods `ONMFTV-MUL1` and `ONMFTV-MUL2` based on the multiplicative update rules do not perform better than some of the other separated methods. Comparing the separated methods with each other, we see that `ONMF-TV-Li` performs remarkably well, with a high stability compared to the other approaches. Note that the stability also seems to depend on the initialization procedure. Regarding this, the SVD seems to favor more stable results than the K-means++ initialization.

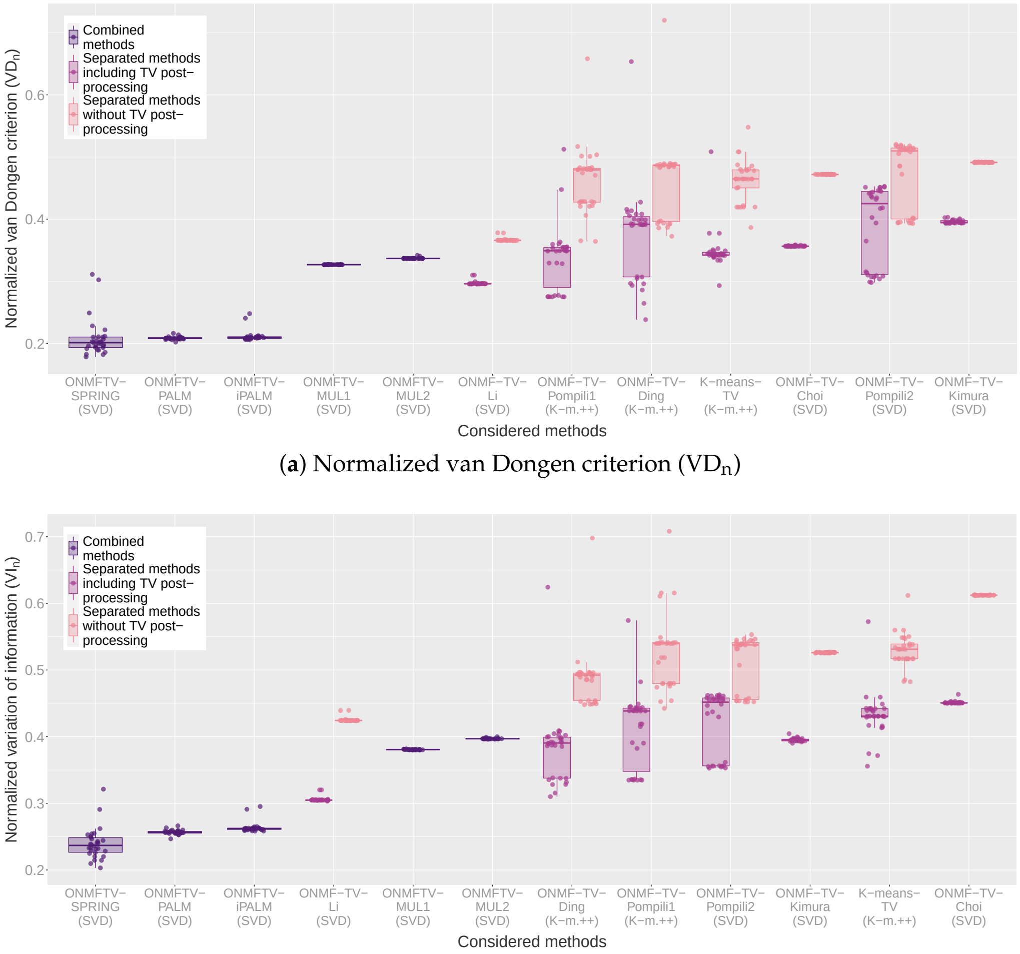
Similar to the cluster validation measures VD_n and VI_n shown in Figure 4, Figure 5 displays the entropy of all performed experiments for all methods. As described in Section 4.3, lower values of the entropy measure indicate better clustering results. This measure also confirms the observation, that the combined methods `ONMFTV-SPRING`, `ONMFTV-PALM`, and `ONMFTV-iPALM` achieve the best results. Concerning the other methods, the outcomes are similar to the ones of the VD_n and VI_n and shall not be discussed in detail.

Figure 6 shows the box plots of the computational times of all replicates for every method. For each considered approach, one box plot is plotted visualizing the computational cost of all performed replicates of the experiment. As in the previous figures, each plotted point corresponds to a replicate showing the specific needed computational time.

The combined methods `ONMFTV-PALM` and `ONMFTV-iPALM` are one of the fastest methods, together with `K-means-TV`, which requires the least time to compute the experiments. Furthermore, we note that `ONMFTV-SPRING` needs significantly more time compared to the other PALM algorithms, which can be also seen by the slightly different asymptotic computational complexities (see Section 3.2.2). The other separated methods are faster than `ONMFTV-SPRING`, except `ONMF-TV-Pompili2`, which needs significantly more time than every other considered approach.

All in all, based on the experiments performed on the MALDI dataset, we recommend the methods based on the PALM scheme, particularly `ONMFTV-PALM`, as well as

ONMFTV-iPALM, since they give the most stable results, while achieving comparatively good results with low computational effort.



(**a**) Normalized van Dongen criterion (VD_n)

(**b**) Normalized variation of information (VI_n)

Figure 4. Box plots of the normalized van Dongen criterion (VD_n) and the normalized variation of information (VI_n) of all performed experiments.

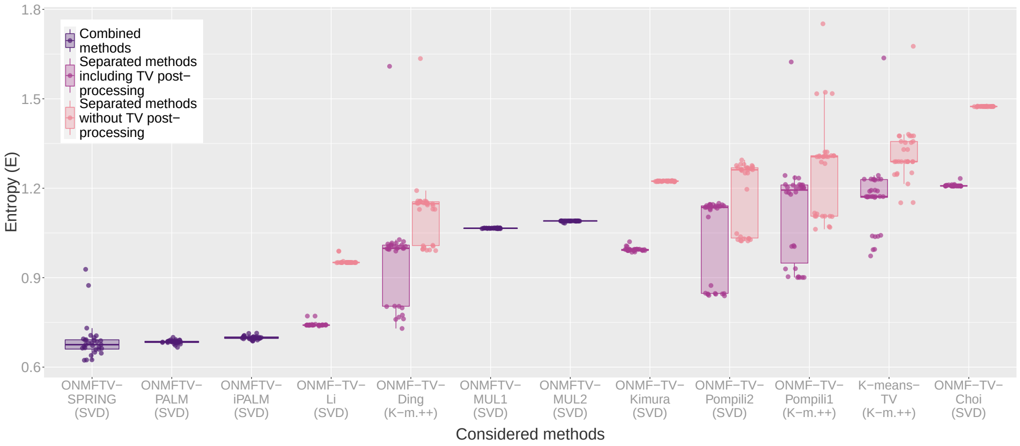


Figure 5. Box plot of the Entropy (E) of all performed experiments.

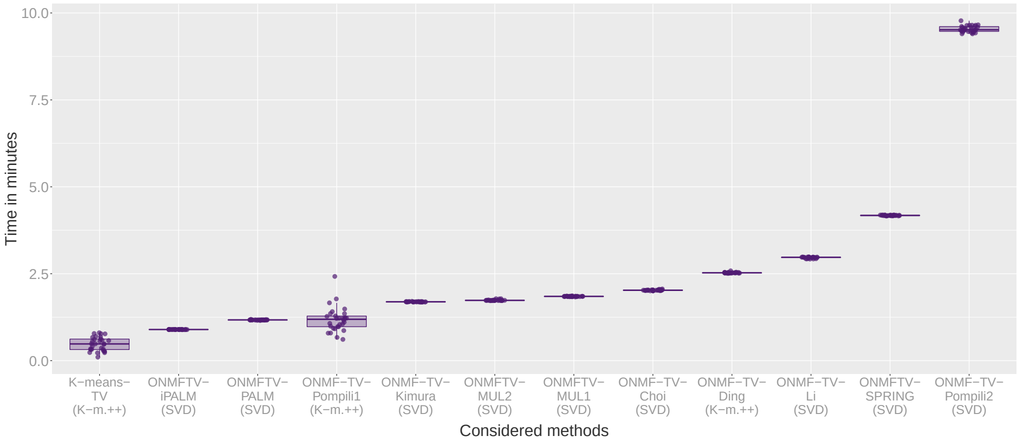


Figure 6. Box plot of the computational times in minutes of all performed experiments.

5. Conclusions

In this work, we have considered various orthogonal nonnegative matrix factorization (ONMF) models, together with different optimization approaches for clustering hyperspectral data, as the main application field. Furthermore, we have introduced total variation regularization in the proposed ONMF models to ensure spatial coherence in the obtained clusterings constituting the main innovation in this paper motivated by numerous spectral imaging applications, which naturally satisfy the spatial coherence in the data.

After a brief description of the main principles of ONMF models, their relation to classical clustering methods, and different optimization techniques, we have proposed so-called separated methods, which apply the TV denoising step after the computation of a classical ONMF algorithm. Furthermore, we have introduced more sophisticated combined methods with different optimization procedures, which include the TV regularization into the considered ONMF model.

For the numerical evaluation, we have compared 12 different TV regularized ONMF methods on a MALDI-MSI human colon hyperspectral dataset with six different spatially coherent tissue regions, which constitute the ground truth for the clustering problem. The qualitative and quantitative results confirmed our expectation that the TV regularization significantly improves the clustering performance. Furthermore, the combined methods based on the proximal alternating linearized minimization have led to the best clustering outcomes and performance stability. Hence, based on the numerical evaluation of the MALDI dataset, we recommend the methods `ONMFTV-PALM` and `ONMFTV-iPALM`, as well as `ONMFTV-SPRING`.

Several further research directions could be of interest. One limitation of the presented approaches is the need of a manual a-priori choice of the needed hyperparameters. Hence, a useful extension of the proposed methods could be to introduce an automated way to choose suitable parameters. Another aspect is the analysis and the derivation of optimization algorithms for the case of discrepancy terms different from the Frobenius norm. Moreover, further gradient estimators different from the SGD could be examined for the method `ONMFTV-SPRING`. Furthermore, another major point is the consideration of more hyperspectral datasets from different application fields and a more thorough numerical evaluation of the different ONMF methods.

A more theoretical research direction could be an extended convergence analysis in particular for the multiplicative algorithms `ONMFTV-MUL1` and `ONMFTV-MUL2`. Finally, the investigation of spatially coherent clustering models in infinite dimension space leading to “continuous” factorization problems with gradient based penalty terms could be interesting. In this setting, the analysis of first order conditions could lead to connections to corresponding K-means clustering models and partial differential equations, whose solutions give insight to the according distance measures and clusters. A first step for such an investigation could be to start with a finite dimensional space based on ONMF models.

Funding: This project was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) within the framework of RTG "π^3: Parameter Identification—Analysis, Algorithms, Applications"—Project number 281474342/GRK2224/1.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The MATLAB® codes of the considered algorithms in this work are available in our GitLab [53] for general nonnegative hyperspectral datasets. Restrictions apply to the availability of the hyperspectral dataset used in this work as it was obtained with the permission of the third party SCiLS (see Reference [58] and www.scils.de).

Acknowledgments: The author would like to thank Sibylle Hess (Eindhoven University of Technology) for fruitful discussions. The used tissue sample is a SCiLS example dataset provided by Axel Walch (German Research Center for Environmental Health, Helmholtz Zentrum München).

Conflicts of Interest: The author declares that there is no conflict of interest regarding the publication of this paper.

Abbreviations

The following abbreviations are used in this manuscript:

H&E	Hematoxylin and Eosin
E	Entropy
iPALM	Inertial PALM
MALDI	Matrix-Assisted Laser Desorption/Ionization
MM	Majorize-Minimization
NMF	Nonnegative Matrix Factorization
ONMF	Orthogonal NMF
PALM	Proximal Alternating Linearized Minimization
SVD	Singular Value Decomposition
SGD	Stochastic Gradient Descent
SPRING	Stochastic Proximal Alternating Linearized Minimization
TV	Total Variation
$\mathrm{VD_n}$	Normalized Van Dongen Criterion
$\mathrm{VI_n}$	Normalized Variation of Information

Appendix A. Derivation of the Algorithm `ONMFTV-MUL1`

In this section, we give an outline of the proof of Theorem 1. For more details on the derivation, we refer the reader to the work of Reference [37].

The update rules in Algorithm 2 are based on the Majorize-Minimization (MM) principle [64]. The basic idea behind this concept is to replace the cost function $\mathcal{F}$ by a so-called surrogate function $\mathcal{Q}_{\mathcal{F}}$, whose minimization is simplified and leads to the desired multiplicative algorithms.

Definition A1. *Let $\Omega \subset \mathbb{R}^{M \times K}$ be an open set, and $\mathcal{F} : \Omega \to \mathbb{R}$ a given cost function. A function $\mathcal{Q}_{\mathcal{F}} : \Omega \times \Omega \to \mathbb{R}$ is called a surrogate function, if it satisfies the following properties:*

(i) $\mathcal{Q}_{\mathcal{F}}(U, A) \geq \mathcal{F}(U)$ *for all* $U, A \in \Omega$,

(ii) $\mathcal{Q}_{\mathcal{F}}(U, U) = \mathcal{F}(U)$ *for all* $U \in \Omega$.

The minimization procedure based on the MM approach is defined by

$$U^{[i+1]} := \arg\min_{U \in \Omega} \mathcal{Q}_{\mathcal{F}}(U, U^{[i]}).$$

Together with the properties of $\mathcal{Q}_{\mathcal{F}}$, this leads directly to the monotone decrease of the cost function $\mathcal{F}$, since

$$\mathcal{F}(U^{[i+1]}) \leq \mathcal{Q}_{\mathcal{F}}(U^{[i+1]}, U^{[i]}) \leq \mathcal{Q}_{\mathcal{F}}(U^{[i]}, U^{[i]}) = \mathcal{F}(U^{[i]}).$$

Tailored construction techniques for surrogate functions leads additionally to the desired multiplicative structure of the update rules and are typically based on Jensen's inequality or the so-called quadratic upper bound principle [37,64].

We first focus on the minimization with respect to U. It can be shown that, for each term of the cost function $\mathcal{F}$ in (7), a surrogate function can be constructed, which finally results to a suitable surrogate $\mathcal{Q}_{\mathcal{F}} := \mathcal{Q}_{\mathcal{F}_1} + \mathcal{Q}_{\mathcal{F}_2} + \mathcal{Q}_{\mathcal{F}_3}$ defined by

$$\begin{aligned}
\mathcal{Q}_{\mathcal{F}_1}(U, A) &:= \frac{1}{2} \sum_{m=1}^{M} \sum_{n=1}^{N} \frac{1}{(AV)_{mn}} \sum_{k=1}^{K} A_{ik} V_{kn} \left(X_{mn} - \frac{U_{mk}}{A_{mk}} (AV)_{mn} \right)^2, \\
\mathcal{Q}_{\mathcal{F}_2}(U, A) &:= \frac{\sigma_1}{2} \sum_{k=1}^{K} \sum_{\ell=1}^{K} \frac{1}{(W^\intercal A)_{k\ell}} \sum_{m=1}^{M} W_{mk} A_{m\ell} \left(\delta_{k\ell} - \frac{U_{m\ell}}{A_{m\ell}} (W^\intercal A)_{k\ell} \right)^2 + \frac{\sigma_2}{2} \|W - U\|_F^2, \\
\mathcal{Q}_{\mathcal{F}_3}(U, A) &:= \frac{\tau}{2} \left(\sum_{k=1}^{K} \sum_{m=1}^{M} \left[P(A)_{mk} (U_{mk} - Z(A)_{mk})^2 \right] + C(A) \right),
\end{aligned}$$

where $C(A)$ is some function depending only on A, and with $P(A)$ and $Z(A)$ given as in Equations (8) and (9). Computing the first order condition $\nabla_U \mathcal{Q}_{\mathcal{F}}(U, A) = 0$ and applying classical calculation rules for derivatives leads then to the desired update rule given in Line 4 of Algorithm 2.

The update rules for V and W are treated similarly.

Appendix B. Details on the Algorithm `ONMFTV-MUL2`

Appendix B.1. Derivation of the Update Rules

Algorithm 3 is based on a classical gradient descent approach with a suitably chosen step size to ensure a multiplicative structure of the update rules. We will discuss here only the derivation for the minimization with respect to U. The update rules for V are classical results and can be found in various works [37,49].

Regarding the update rule for U, we consider the classical gradient descent step

$$U^{[i+1]} := U^{[i]} - \Gamma^{[i]} \circ \nabla_U \mathcal{F}(U^{[i]}, V),$$

given also in (13). Thus, we need an explicit expression for $\nabla_U \mathcal{F}(U, V)$. Classical calculation rules for derivatives yields

$$\nabla_U \mathcal{F}(U, V) = UVV^\intercal - XV^\intercal + \sigma_1 (UU^\intercal U - U) + \nabla_U \operatorname{TV}_{\varepsilon_{\mathrm{TV}}}(U).$$

The gradient of $\operatorname{TV}_{\varepsilon_{\mathrm{TV}}}(U)$ can be acquired via the Euler-Lagrange equation and is a classical result. By considering the continuous case with the function $u : \Omega \to \mathbb{R}$ of Section 3.2.1 and defining $\mathcal{L}(x_1, x_2, u, u_1, u_2) := \|\nabla u\|_{\varepsilon_{\mathrm{TV}}}$ with $u_i := \partial u / \partial x_i$ for $i \in \{1, 2\}$, the Euler Lagrange Equation

$$\frac{\partial \mathcal{L}}{\partial u} - \sum_{i=1}^{2} \frac{\partial}{\partial x_i} \left(\frac{\partial \mathcal{L}}{\partial u_i} \right) = 0$$

gives the formal relationship $\nabla_u \operatorname{TV}_{\varepsilon_{\mathrm{TV}}}(u) := -\operatorname{div}(\nabla u / \|\nabla u\|_{\varepsilon_{\mathrm{TV}}})$. Thus, the gradient descent step is given by

$$U^{[i+1]} := U^{[i]} - \Gamma^{[i]} \circ \left(U^{[i]} VV^\intercal - XV^\intercal - \tau \operatorname{div} \left(\frac{\nabla U^{[i]}}{\|\nabla U^{[i]}\|_{\varepsilon_{\mathrm{TV}}}} \right) + \sigma_1 U^{[i]} U^{[i]\intercal} U^{[i]} - \sigma_1 U^{[i]} \right).$$

To ensure a multiplicative structure of the update rules, we set the step size to be

$$\Gamma^{[i]} := \frac{U^{[i]}}{U^{[i]} V V^{\intercal} + \sigma_1 U^{[i]} U^{[i]\intercal} U^{[i]}},$$

which leads directly to the update rule in Line 4 of Algorithm 3.

Appendix B.2. Discretization of the TV Gradient

In this section, we describe the discretization procedure of the divergence term $\operatorname{div}\left(\nabla U^{[i]} / \|\nabla U^{[i]}\|_{\varepsilon_{\mathrm{TV}}}\right)$, which occurs in Line 4 of Algorithm 3.

To perform such a discretization, it is needed for express the divergence term in terms of sums and products of first and second order derivatives. To simplify the notation, we stick in this first step to the continuous case and consider the function u mentioned in (10). By considering the definition in (11) and applying classical calculation rules for derivatives, we get

$$\operatorname{div}\left(\frac{\nabla u}{\|\nabla u\|_{\varepsilon_{\mathrm{TV}}}}\right) = \frac{\varepsilon_{\mathrm{TV}}^2 \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) + \left(\frac{\partial u}{\partial x}\right)^2 \frac{\partial^2 u}{\partial y^2} + \left(\frac{\partial u}{\partial y}\right)^2 \frac{\partial^2 u}{\partial x^2} - 2 \frac{\partial u}{\partial x} \frac{\partial u}{\partial y} \frac{\partial^2 u}{\partial x \partial y}}{\|\nabla u\|_{\varepsilon_{\mathrm{TV}}}^3}.$$

For the discretization of this expression, we assume that u is a given discretized image of size $d_1 \times d_2$. Note that, in the NMF models, which are considered in this work, this would correspond to one (reshaped) column of the matrix U. To approximate the partial derivatives in the above expression, we use in the following a central differencing scheme and interpret the x and y directions as the vertical and horizontal axis in the image u, respectively. Thus, we define

$$\begin{aligned}
(\Delta_x u)_{ij} &:= \frac{u_{i+1,j} - u_{i-1,j}}{2}, & (\Delta_y u)_{ij} &:= \frac{u_{i,j+1} - u_{i,j-1}}{2}, \\
(\Delta_{xx} u)_{ij} &:= u_{i+1,j} - 2u_{ij} + u_{i-1,j}, & (\Delta_{yy} u)_{ij} &:= u_{i,j+1} - 2u_{ij} + u_{i,j-1}, \\
(\Delta_{xy} u)_{ij} &:= \frac{u_{i+1,j+1} - u_{i+1,j-1} - u_{i-1,j+1} + u_{i-1,j-1}}{4}. & &
\end{aligned}$$

The discretized gradients can also be interpreted as matrices of size $d_1 \times d_2$, which finally leads to the discretization of the divergence term

$$\operatorname{div}\left(\frac{\nabla u}{\|\nabla u\|_{\varepsilon_{\mathrm{TV}}}}\right) = \frac{\varepsilon_{\mathrm{TV}}^2 (\Delta_{xx} u + \Delta_{yy} u) + (\Delta_x u)^2 \circ \Delta_{yy} u + (\Delta_y u)^2 \circ \Delta_{xx} u - 2 \Delta_x u \circ \Delta_y u \circ \Delta_{xy} u}{\left((\Delta_x u)^2 + (\Delta_y u)^2 + \varepsilon_{\mathrm{TV}}^2 1_{d_1 \times d_2}\right)^{3/2}},$$

where $1_{d_1 \times d_2}$ is a matrix of size $d_1 \times d_2$ with ones in every entry.

Appendix C. Algorithmic Details on the Proximal Gradient Descent Approach

In this section, we give some information about the involved gradients, Lipschitz constants, and the computation of the step sizes of all algorithms based on the proximal gradient descent approach (see Section 3.2.2).

Appendix C.1. `ONMFTV-PALM`

We start in this section with details on the gradients of the algorithm `ONMFTV-PALM`. Since the computations are straightforward, we will only state the final results of all three gradients:

$$\begin{aligned}
\nabla_U \mathcal{F}(U, V, W) &= U V V^{\intercal} - X V^{\intercal} + \sigma_1 (W W^{\intercal} U - W) + \sigma_2 (U - W), \\
\nabla_V \mathcal{F}(U, V, W) &= U^{\intercal} U V - U^{\intercal} X, \\
\nabla_W \mathcal{F}(U, V, W) &= \sigma_1 (U U^{\intercal} W - U) + \sigma_2 (W - U).
\end{aligned}$$

For the calculation of the Lipschitz constants, we compute for the minimization with respect to U and arbitrary matrices $U_1, U_2 \in \mathbb{R}^{M \times K}$

$$\begin{aligned}
\|\nabla_U \mathcal{F}(U_1, V, W) - \nabla_U \mathcal{F}(U_2, V, W)\| &= \|(U_1 - U_2)VV^\intercal + \sigma_1 WW^\intercal(U_1 - U_2) + \sigma_2(U_1 - U_2)\| \\
&\leq (\|VV^\intercal\| + \|\sigma_1 WW^\intercal + \sigma_2 I_{M\times M}\|) \cdot \|U_1 - U_2\| \\
&= (\lambda_{1,U} + \lambda_{2,U}) \cdot \|U_1 - U_2\|,
\end{aligned}$$

where $\|\cdot\|$ is the spectral norm, and $\lambda_{1,U}, \lambda_{2,U}$ the maximal absolute eigenvalue of the symmetric matrices $VV^\intercal$ and $\sigma_1 WW^\intercal + \sigma_2 I_{M\times M}$, respectively, with $I_{M\times M}$ being the identity matrix of size $M \times M$. The other cases are treated analogously and result in

$$\begin{aligned}
\|\nabla_V \mathcal{F}(U, V_1, W) - \nabla_V \mathcal{F}(U, V_2, W)\| &\leq \|U^\intercal U\| \cdot \|V_1 - V_2\| = \lambda_{1,V}\|V_1 - V_2\|, \\
\|\nabla_W \mathcal{F}(U, V, W_1) - \nabla_W \mathcal{F}(U, V, W_2)\| &\leq \|\sigma_1 UU^\intercal + \sigma_2 I_{M\times M}\| \cdot \|W_1 - W_2\| \\
&= \lambda_{1,W}\|W_1 - W_2\|.
\end{aligned}$$

Numerically, the eigenvalues of the matrices are approximated via the power iteration with 5 iterations. The step sizes η_U, η_V, and η_W described in Algorithm 4 are computed based on the Lipschitz constants, which are given by $L_U := \lambda_{1,U} + \lambda_{2,U}$, $L_V := \lambda_{1,V}$, and $L_W := \lambda_{1,W}$ with the computation above. The standard choice for the step size of the `ONMFTV-PALM` algorithm are simply the inverse Lipschitz constants, i.e. $\eta_U = 1/L_U, \eta_V = 1/L_V$, and $\eta_W = 1/L_W$ [50,56].

Appendix C.2. ONMFTV-iPALM

The calculation of the gradients and Lipschitz constants for `ONMFTV-iPALM` are the same as in the case of `ONMFTV-PALM` with the slight difference that the gradients are evaluated at other points. Furthermore, the step sizes are set by $\eta_U = 0.9/L_U, \eta_V = 0.9/L_V$, and $\eta_W = 0.9/L_W$ according to Reference [50]. For details, we refer the reader to References [50,55] and to the codes available in our GitLab [53].

Appendix C.3. ONMFTV-SPRING

The computation of the gradients for `ONMFTV-SPRING` are based on the SGD estimator given in (16). As in the case of `ONMFTV-PALM`, the computations are straightforward, and we get, by defining the mini-batch $\mathcal{B}^V_{i,j} \subset \{1, \dots, M\}$ for the minimization with respect to V and with $\mathcal{F}_m(U, V, W) := {}^1\!/_2\|X_{m,\bullet} - U_{m,\bullet}V\|_2^2$, the gradients

$$\begin{aligned}
\tilde{\nabla}^{i,j}_U \mathcal{F}(U, V, W) &= \sum_{n\in\mathcal{B}^U_{i,j}} \left[(UV)_{\bullet,n} V^\intercal_{n,\bullet} - X_{\bullet,n} V^\intercal_{n,\bullet} + {}^1\!/_N(\sigma_1(WW^\intercal U - W) + \sigma_2(U - W))\right] \\
&= UV_{\bullet,\mathcal{B}^U_{i,j}} V^\intercal_{\mathcal{B}^U_{i,j},\bullet} - X_{\bullet,\mathcal{B}^U_{i,j}} V^\intercal_{\mathcal{B}^U_{i,j},\bullet} + \frac{|\mathcal{B}^U_{i,j}|}{N}(\sigma_1(WW^\intercal U - W) + \sigma_2(U - W)), \\
\tilde{\nabla}^{i,j}_V \mathcal{F}(U, V, W) &= \sum_{m\in\mathcal{B}^V_{i,j}} \nabla_V \mathcal{F}_m(U, V, W) = \sum_{m\in\mathcal{B}^V_{i,j}} U^\intercal_{\bullet,m}(UV)_{m,\bullet} - U^\intercal_{\bullet,m} X_{m,\bullet} \\
&= U^\intercal_{\bullet,\mathcal{B}^V_{i,j}} U_{\mathcal{B}^V_{i,j},\bullet} V - U^\intercal_{\bullet,\mathcal{B}^V_{i,j}} X_{\mathcal{B}^V_{i,j},\bullet},
\end{aligned}$$

where $V_{\bullet,\mathcal{B}^U_{i,j}}, X_{\bullet,\mathcal{B}^U_{i,j}}$, and $U_{\mathcal{B}^V_{i,j},\bullet}, X_{\mathcal{B}^V_{i,j},\bullet}$ are the submatrices of U, V, and X, which are constrained column-wise and row-wise based on the index sets $\mathcal{B}^U_{i,j}$ and $\mathcal{B}^V_{i,j}$, respectively. For the minimization with respect to W, the full partial gradient of $\mathcal{F}$ is used, which was already computed in Appendix C.1.

The computation of the Lipschitz constants of the partial gradients of $\mathcal{F}$ with respect to U and V is based on the SGD estimator and goes analogously to the steps in Appendix C.1. Hence, we have

$$\begin{aligned}\|\tilde{\nabla}_U^{i,j}\mathcal{F}(U_1,V,W) - \tilde{\nabla}_U^{i,j}\mathcal{F}(U_2,V,W)\| &\leq \left(\|V_{\bullet,\mathcal{B}_{i,j}^U}V_{\mathcal{B}_{i,j}^U,\bullet}^\intercal\| + \frac{|\mathcal{B}_{i,j}^U|}{N}\|\sigma_1 WW^\intercal + \sigma_2 I_{M\times M}\|\right)\|U_1 - U_2\| \\ &= (\lambda_{1,U} + \lambda_{2,U})\cdot\|U_1 - U_2\|, \\ \|\tilde{\nabla}_V^{i,j}\mathcal{F}(U,V_1,W) - \tilde{\nabla}_V^{i,j}\mathcal{F}(U,V_2,W)\| &\leq \|U_{\bullet,\mathcal{B}_{i,j}^V}^\intercal U_{\mathcal{B}_{i,j}^V,\bullet}\|\cdot\|V_1 - V_2\| = \lambda_{1,V}\|V_1 - V_2\|,\end{aligned}$$

together with the Lipschitz constants L_U, L_V analogously to Appendix C.1. Since we consider the full partial gradient of $\mathcal{F}$ with respect to W, we also get the same Lipschitz constant L_W as in Appendix C.1.

The choice of of the step sizes $\eta_{U^{[i,j]}}$ and $\eta_{V^{[i,j]}}$ are chosen according to the work of Reference [50] by defining

$$\eta_{U^{[i,j]}} := \min\left\{\frac{1}{\sqrt{\lceil i\cdot|\mathcal{B}_{i,j}^U|/N\rceil}\cdot L_U}, \frac{1}{L_U}\right\}, \quad \eta_{V^{[i,j]}} := \min\left\{\frac{1}{\sqrt{\lceil i\cdot|\mathcal{B}_{i,j}^V|/M\rceil}\cdot L_V}, \frac{1}{L_V}\right\}.$$

Furthermore, as described in Section 4.3, we perform an additional projection step of the parameter $\tau\eta_{U^{[i]}}$ (see Line 6 in Algorithm 5 and the application of the $\mathrm{prox}_{\tau\eta_{U^{[i,j]}}\mathcal{J}}$ operator) to avoid too large regularization parameters of the TV penalty term by applying $\mathrm{prox}_{\tau_{i,j}\mathcal{J}}$ with $\tau_{i,j} := \min\{\tau\eta_{U^{[i,j]}},\ 1\times 10^{-3}\}$. For more details on the algorithm, we refer the reader to the codes available in our GitLab [53].

Appendix D. Parameter Choice

As described in Section 4.3, we choose the regularization parameters for the numerical experiments of every considered method empirically by performing multiple experiments for different parameters and choosing the ones which lead to stable experiments and the best VD_n. For the separated methods, we also follow, partially, the recommendations of the corresponding works. In the following sections, we describe the choice of these and other hyperparameters in more detail.

Appendix D.1. Separated Methods

In this section, we describe the choice of the main hyperparameters of all separated methods. Table A1 shows the selected parameters for the numerical experiments in Section 4, along with the used stopping criteria (stopCrit) and initialization methods (initMethod).

Table A1. Parameter choice of the separated methods for the numerical experiments with the MALDI dataset.

Method	τ	**stopCrit**	$i_{\max}$	**initMethod**
`K-means-TV`	1	Cluster assignment	–	K-means++
`ONMF-TV-Choi`	2×10^{-2}	maxIt	6×10^{2}	SVD
`ONMF-TV-Ding`	2×10^{-2}	maxIt	8×10^{2}	K-means++
`ONMF-TV-Pompili1`	1	Cluster assignment	–	K-means++
`ONMF-TV-Pompili2`	4×10^{-2}	Pompili Internal	–	SVD
`ONMF-TV-Kimura`	2×10^{-2}	maxIt	7×10^{2}	SVD
`ONMF-TV-Li`	3×10^{-2}	maxIt	2×10^{2}	SVD

All separated methods are initialized either based on the SVD approach described in Section 4.3 or via K-means++. Both methods were tried out for every separated method in the numerical experiments, and one of them was chosen based on the stability and quality measures of the results. For the specific case of `ONMF-TV-Pompili1`, we use a mixture of the K-means++ method and the internal initialization procedure of Reference [16].

Furthermore, different stopping criteria were used. Besides the classical stopping criterion based on a maximal iteration number $i_{\max}$ (maxIt), the clustering algorithms `K-means-TV` and `ONMF-TV-Pompili1` (i.e., Line 3 in Algorithm 1) are stopped until the cluster assignments in the cluster membership matrix U do not change anymore. For the special case of `ONMF-TV-Pompili2`, the algorithm stops until the current iterates are "sufficiently" nonnegative (see Reference [16]).

Finally, the TV denoising algorithm in Line 4 of all separated methods is based on a Fast Iterative Shrinkage-Thresholding Algorithm (FISTA; see Reference [65]) with a maximal iteration number of 100.

Appendix D.2. Combined Methods

In this section, we describe the choice of the main hyperparameters of all combined methods. As in the previous section, Table A2 shows the selected parameters for the numerical experiments in Section 4, along with the used stopping criteria (stopCrit) and initialization methods (initMethod).

Table A2. Parameter choice of the combined methods for the numerical experiments with the MALDI dataset.

Method	σ_1	σ_2	τ	ε_{TV}	s_r	stopCrit	$i_{\max}$	initMethod
`ONMFTV-MUL1`	0.5	0.5	5×10^{-3}	$\sqrt{1 \times 10^{-5}}$	–	maxIt	8×10^2	SVD
`ONMFTV-MUL2`	1	–	1×10^{-3}	$\sqrt{1 \times 10^{-5}}$	–	maxIt	7×10^2	SVD
`ONMFTV-PALM`	0.1	0.1	1×10^{-1}	–	–	maxIt	4×10^2	SVD
`ONMFTV-iPALM`	0.1	0.1	1×10^{-1}	–	–	maxIt	3×10^2	SVD
`ONMFTV-SPRING`	0.1	0.1	1×10^{-4}	–	40	maxIt	1×10^2	SVD

All combined methods are initialized based on the SVD approach described in Section 4.3 and are stopped until $i_{\max}$ iterations are reached (maxIt). Regarding the initialization methods, kmeans++ and the SVD method were tried out, and one of them was finally chosen based on the quality of the results as it is the case for the separated methods. For `ONMFTV-SPRING`, we choose $s_r = 40$, according to Reference [50].

References

1. Aggarwal, C.C.; Reddy, C.K. *Data Clustering: Algorithms and Applications*, 1st ed.; Chapman & Hall/CRC: Boca Raton, FL, USA, 2013. [CrossRef]
2. Ding, C.; He, X.; Simon, H.D. On the Equivalence of Nonnegative Matrix Factorization and Spectral Clustering. In Proceedings of the 2005 SIAM International Conference on Data Mining, Newport Beach, CA, USA, 21–23 April 2005; Volume 5, pp. 606–610. [CrossRef]
3. Ding, C.; Li, T.; Peng, W.; Park, H. Orthogonal Nonnegative Matrix T-Factorizations for Clustering. In Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, 20–23 August 2006; pp. 126–135. [CrossRef]
4. Li, T.; Ding, C. The Relationships Among Various Nonnegative Matrix Factorization Methods for Clustering. In Proceedings of the Sixth International Conference on Data Mining (ICDM'06), Hong Kong, China, 18–22 December 2006; pp. 362–371. [CrossRef]
5. Choi, S. Algorithms for orthogonal nonnegative matrix factorization. In Proceedings of the International Joint Conference on Neural Networks, Hong Kong, China, 1–8 June 2008; pp. 1828–1832. [CrossRef]
6. Yang, Z.; Oja, E. Linear and Nonlinear Projective Nonnegative Matrix Factorization. *IEEE Trans. Neural Netw.* **2010**, *21*, 734–749. [CrossRef] [PubMed]
7. Li, Z.; Wu, X.; Peng, H. Nonnegative Matrix Factorization on Orthogonal Subspace. *Pattern Recognit. Lett.* **2010**, *31*, 905–911. [CrossRef]

8. Pan, J.; Ng, M.K. Orthogonal Nonnegative Matrix Factorization by Sparsity and Nuclear Norm Optimization. *SIAM J. Matrix Anal. Appl.* **2018**, *39*, 856–875. [CrossRef]
9. Mirzal, A. A convergent algorithm for orthogonal nonnegative matrix factorization. *J. Comput. Appl. Math.* **2014**, *260*, 149–166. [CrossRef]
10. Zhang, M.; Jia, P.; Shen, Y.; Shen, F. Hyperspectral image classification method based on orthogonal NMF and LPP. In Proceedings of the 2016 IEEE International Instrumentation and Measurement Technology Conference Proceedings, Taipei, Taiwan, 23–26 May 2016; pp. 1–5. [CrossRef]
11. Kimura, K.; Tanaka, Y.; Kudo, M. A Fast Hierarchical Alternating Least Squares Algorithm for Orthogonal Nonnegative Matrix Factorization. In Proceedings of the Sixth Asian Conference on Machine Learning, Nha Trang City, Vietnam, 26–28 November 2015; Volume 39, pp. 129–141.
12. Li, B.; Zhou, G.; Cichocki, A. Two Efficient Algorithms for Approximately Orthogonal Nonnegative Matrix Factorization. *IEEE Signal Process. Lett.* **2015**, *22*, 843–846. [CrossRef]
13. Li, W.; Li, J.; Liu, X.; Dong, L. Two fast vector-wise update algorithms for orthogonal nonnegative matrix factorization with sparsity constraint. *J. Comput. Appl. Math.* **2020**, *375*, 112785. [CrossRef]
14. Wang, S.; Chang, T.H.; Cui, Y.; Pang, J.S. Clustering by Orthogonal Non-negative Matrix Factorization: A Sequential Non-convex Penalty Approach. In Proceedings of the 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2019; pp. 5576–5580. [CrossRef]
15. Ahookhosh, M.; Hien, L.T.K.; Gillis, N.; Patrinos, P. Multi-block Bregman proximal alternating linearized minimization and its application to orthogonal nonnegative matrix factorization. *Comput. Optim. Appl.* **2021**, *79*, 681–715. [CrossRef]
16. Pompili, F.; Gillis, N.; Absil, P.A.; Glineur, F. Two algorithms for orthogonal nonnegative matrix factorization with application to clustering. *Neurocomputing* **2014**, *141*, 15–25. [CrossRef]
17. Qiu, Y.; Zhou, G.; Xie, K. Deep Approximately Orthogonal Nonnegative Matrix Factorization for Clustering. *arXiv* **2017**, arXiv:1711.07437.
18. Asteris, M.; Papailiopoulos, D.; Dimakis, A.G. Orthogonal NMF through Subspace Exploration. In *Advances in Neural Information Processing Systems 28*; Curran Associates, Inc.: New York, NY, USA, 2015; pp. 343–351.
19. Zhang, W.E.; Tan, M.; Sheng, Q.Z.; Yao, L.; Shi, Q. Efficient Orthogonal Non-Negative Matrix Factorization over Stiefel Manifold. In Proceedings of the 25th ACM International on Conference on Information and Knowledge Management. Association for Computing Machinery, Indianapolis, IN, USA, 24–28 October 2016; pp. 1743–1752. [CrossRef]
20. Asadi, S.; Povh, J. A Block Coordinate Descent-Based Projected Gradient Algorithm for Orthogonal Non-Negative Matrix Factorization. *Mathematics* **2021**, *9*, 540. [CrossRef]
21. Ahookhosh, M.; Hien, L.T.K.; Gillis, N.; Patrinos, P. A Block Inertial Bregman Proximal Algorithm for Nonsmooth Nonconvex Problems with Application to Symmetric Nonnegative Matrix Tri-Factorization. *J. Optim. Theory Appl.* **2021**, *190*, 234–258. [CrossRef]
22. Hribar, R.; Hrga, T.; Papa, G.; Petelin, G.; Povh, J.; Pržulj, N.; Vukašinović, V. Four algorithms to solve symmetric multi-type non-negative matrix tri-factorization problem. *J. Glob. Optim.* **2021**, 1–30. [CrossRef]
23. Li, T.; Ding, C. Nonnegative matrix factorizations for clustering: A survey. In *Data Clustering*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2014; pp. 149–175.
24. Türkmen, A.C. A Review of Nonnegative Matrix Factorization Methods for Clustering. *arXiv* **2015**, arXiv:1507.03194.
25. Gillis, N. *Nonnegative Matrix Factorization*; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2020; doi:10.1137/1.9781611976410. [CrossRef]
26. Despotović, I.; Vansteenkiste, E.; Philips, W. Spatially Coherent Fuzzy Clustering for Accurate and Noise-Robust Image Segmentation. *IEEE Signal Process. Lett.* **2013**, *20*, 295–298. [CrossRef]
27. Zabih, R.; Kolmogorov, V. Spatially coherent clustering using graph cuts. In Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, 27 June–2 July 2004; Volume 2. [CrossRef]
28. Huang, R.; Sang, N.; Luo, D.; Tang, Q. Image segmentation via coherent clustering in L*a*b* color space. *Pattern Recognit. Lett.* **2011**, *32*, 891–902. [CrossRef]
29. Mignotte, M. A de-texturing and spatially constrained K-means approach for image segmentation. *Pattern Recognit. Lett.* **2011**, *32*, 359–367. [CrossRef]
30. He, W.; Zhang, H.; Zhang, L. Total Variation Regularized Reweighted Sparse Nonnegative Matrix Factorization for Hyperspectral Unmixing. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 3909–3921. [CrossRef]
31. Feng, X.R.; Li, H.C.; Li, J.; Du, Q.; Plaza, A.; Emery, W.J. Hyperspectral Unmixing Using Sparsity-Constrained Deep Nonnegative Matrix Factorization With Total Variation. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 6245–6257. [CrossRef]
32. Feng, X.R.; Li, H.C.; Wang, R. Hyperspectral Unmixing Based on Sparsity-Constrained Nonnegative Matrix Factorization with Adaptive Total Variation. In Proceedings of the 2019 IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 28 July–2 August 2019; pp. 2139–2142. [CrossRef]
33. Zhang, T.; Fang, B.; Liu, W.; Tang, Y.Y.; He, G.; Wen, J. Total variation norm-based nonnegative matrix factorization for identifying discriminant representation of image patterns. *Neurocomputing* **2008**, *71*, 1824–1831. [CrossRef]

34. Yin, H.; Liu, H. Nonnegative matrix factorization with bounded total variational regularization for face recognition. *Pattern Recognit. Lett.* **2010**, *31*, 2468–2473. [CrossRef]
35. Leng, C.; Cai, G.; Yu, D.; Wang, Z. Adaptive total-variation for non-negative matrix factorization on manifold. *Pattern Recognit. Lett.* **2017**, *98*, 68–74. [CrossRef]
36. Casalino, G.; Gillis, N. Sequential dimensionality reduction for extracting localized features. *Pattern Recognit.* **2017**, *63*, 15–29. [CrossRef]
37. Fernsel, P.; Maass, P. A Survey on Surrogate Approaches to Non-negative Matrix Factorization. *Vietnam J. Math.* **2018**, *46*, 987–1021. [CrossRef]
38. Paatero, P.; Tapper, U. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics* **1994**, *5*, 111–126. [CrossRef]
39. Kim, J.; Park, H. *Sparse Nonnegative Matrix Factorization for Clustering*; Technical Report; Georgia Institute of Technology: Atlanta, GA, USA, 2008.
40. Févotte, C.; Bertin, N.; Durrieu, J.L. Nonnegative Matrix Factorization with the Itakura-Saito-Divergence: With Application to Music Analysis. *Neural Comput.* **2009**, *21*, 793–830. [CrossRef] [PubMed]
41. Arridge, S.; Fernsel, P.; Hauptmann, A. Joint Reconstruction and Low-Rank Decomposition for Dynamic Inverse Problems. *arXiv* **2020**, arXiv:2005.14042.
42. Leuschner, J.; Schmidt, M.; Fernsel, P.; Lachmund, D.; Boskamp, T.; Maass, P. Supervised non-negative matrix factorization methods for MALDI imaging applications. *Bioinformatics* **2019**, *35*, 1940–1947. [CrossRef] [PubMed]
43. Klingenberg, B.; Curry, J.; Dougherty, A. Non-negative matrix factorization: Ill-posedness and a geometric algorithm. *Pattern Recognit.* **2009**, *42*, 918–928. [CrossRef]
44. Pham, Q.M.; Lachmund, D.; Hào, D.N. Convergence of proximal algorithms with stepsize controls for non-linear inverse problems and application to sparse non-negative matrix factorization. *Numer. Algorithms* **2020**, *85*, 1255–1279. [CrossRef]
45. Cai, J.F.; Jia, X.; Gao, H.; Jiang, S.B.; Shen, Z.; Zhao, H. Cine Cone Beam CT Reconstruction Using Low-Rank Matrix Factorization: Algorithm and a Proof-of-Principle Study. *IEEE Trans. Med. Imaging* **2014**, *33*, 1581–1591. [CrossRef]
46. Chen, K. On Coresets for k-Median and k-Means Clustering in Metric and Euclidean Spaces and Their Applications. *SIAM J. Comput.* **2009**, *39*, 923–947. [CrossRef]
47. Mahajan, M.; Nimbhorkar, P.; Varadarajan, K. The planar k-means problem is NP-hard. *Theor. Comput. Sci.* **2012**, *442*, 13–21. [CrossRef]
48. Lee, D.D.; Seung, H.S. Learning the parts of objects by non-negative matrix factorization. *Nature* **1999**, *401*, 788–791. [CrossRef] [PubMed]
49. Lee, D.D.; Seung, H.S. Algorithms for non-negative matrix factorization. In Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems (NIPS) 2000, Denver, CO, USA, 27 November–2 December 2000; pp. 556–562.
50. Driggs, D.; Tang, J.; Liang, J.; Davies, M.; Schönlieb, C.B. SPRING: A fast stochastic proximal alternating method for non-smooth non-convex optimization. *arXiv* **2020**, arXiv:2002.12266.
51. Chambolle, A.; Caselles, V.; Novaga, M.; Cremers, D.; Pock, T. An introduction to Total Variation for Image Analysis. *Theor. Found. Numer. Methods Sparse Recovery* **2010**, *9*, 263–340. [CrossRef]
52. Beck, A.; Teboulle, M. Fast Gradient-Based Algorithms for Constrained Total Variation Image Denoising and Deblurring Problems. *IEEE Trans. Image Process.* **2009**, *18*, 2419–2434. [CrossRef]
53. Fernsel, P. Spatially Coherent Clustering Based on Orthogonal Nonnegative Matrix Factorization—Codes and Algorithms. Available online: https://gitlab.informatik.uni-bremen.de/s_p32gf3/spatially_coherent_clustering_with_onmf (accessed on 18 August 2021).
54. Defrise, M.; Vanhove, C.; Liu, X. An algorithm for total variation regularization in high-dimensional linear problems. *Inverse Probl.* **2011**, *27*, 065002. [CrossRef]
55. Pock, T.; Sabach, S. Inertial Proximal Alternating Linearized Minimization (iPALM) for Nonconvex and Nonsmooth Problems. *SIAM J. Imaging Sci.* **2016**, *9*, 1756–1787. [CrossRef]
56. Bolte, J.; Sabach, S.; Teboulle, M. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Math. Program.* **2014**, *146*, 459–494. [CrossRef]
57. Bochnak, J.; Coste, M.; Roy, M.F. *Real Algebraic Geometry*; Springer: Berlin/Heidelberg, Germany, 1998.
58. Alexandrov, T.; Meding, S.; Trede, D.; Kobarg, J.; Balluff, B.; Walch, A.; Thiele, H.; Maass, P. Super-resolution segmentation of imaging mass spectrometry data: Solving the issue of low lateral resolution. *J. Proteom.* **2011**, *75*, 237–245. [CrossRef]
59. Aichler, M.; Walch, A. MALDI Imaging mass spectrometry: Current frontiers and perspectives in pathology research and practice. *Lab. Investig.* **2015**, *95*, 422–431. [CrossRef] [PubMed]
60. Banerjee, A.; Dhillon, I.S.; Ghosh, J.; Sra, S. Generative Model-based Clustering of Directional Data. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), Washington, DC, USA, 24–27 August 2003. [CrossRef]

61. Xiong, H.; Li, Z. Clustering Validation Measures. In *Data Clustering*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2014; pp. 571–605.
62. Halko, N.; Martinsson, P.G.; Shkolnisky, Y.; Tygert, M. An Algorithm for the Principal Component Analysis of Large Data Sets. *SIAM J. Sci. Comput.* **2011**, *33*, 2580–2594. [CrossRef]
63. Boutsidis, C.; Gallopoulos, E. SVD based initialization: A head start for nonnegative matrix factorization. *Pattern Recognit.* **2008**, *41*, 1350–1362. [CrossRef]
64. Lange, K. *Optimization*, 2nd ed.; Springer Texts in Statistics; Springer: New York, NY, USA, 2013; Volume 95.
65. Beck, A.; Teboulle, M. A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems. *SIAM J. Imaging Sci.* **2009**, *2*, 183–202. [CrossRef]