Rank Suggestion in Non-negative Matrix Factorization: Residual Sensitivity to Initial Conditions (RSIC)

Anonymous authors
Paper under double-blind review

Abstract

Determining the appropriate rank in Non-negative Matrix Factorization (NMF) is a critical challenge that often requires extensive parameter tuning and domain-specific knowledge. Traditional methods for rank determination focus on identifying a single optimal rank, which may not capture the complex structure inherent in real-world datasets. In this study, we introduce a novel approach called Residual Sensitivity to Intial Conditions (RSIC) that suggests potentially multiple ranks of interest by analyzing the sensitivity of the relative residuals (e.g. relative reconstruction error) to different initializations. By computing the Mean Coordinatewise Interquartile Range (MCI) of the residuals across multiple random initializations, our method identifies regions where the NMF solutions are less sensitive to initial conditions and potentially more meaningful. We evaluate RSIC on a diverse set of datasets, including single-cell gene expression data, image data, and text data, and compare it against current state-of-the-art existing rank determination methods. Our experiments demonstrate that RSIC effectively identifies relevant ranks consistent with the underlying structure of the data, outperforming traditional methods in scenarios where they are computationally infeasible or less accurate. This approach provides a more scalable and generalizable solution for rank determination in NMF that does not rely on domain-specific knowledge or assumptions.

1 Introduction

Low-dimensional models of high-dimensional data are foundational for exploratory data analyses. Non-negative Matrix Factorization (NMF) has emerged as one such tool for data decomposition and analysis in various domains, including image processing (Guillamet et al., 2002; Lee & Seung, 1999; Liu et al., 2012), text mining (Hassani et al., 2019; Pauca et al., 2004), and bioinformatics (Devarajan, 2008; Gaujoux & Seoighe, 2010). By decomposing a non-negative matrix into non-negative factors, NMF is often able to extract meaningful patterns and components from complex datasets (Gillis, 2020). However, a critical challenge in applying NMF is determining the appropriate rank of decomposition (Wang & Zhang, 2013), which essentially dictates the number of components to extract from the data.

Traditionally, rank determination methods in NMF have largely focused on identifying a single "optimal" rank using heuristic methods or by leveraging additional knowledge of the distribution of the input data. They often require arbitrary parameter choices, are sensitive to the initialization, or depend on domain-specific knowledge, which may not always be available or easily interpretable. While often useful, these methods have considerable limitations. In this study, we introduce a novel approach to rank determination that seeks to suggest a number of ranks of interest instead of a single optimal rank.

Our method, Residual Sensitivity to Initial Conditions (RSIC), is based on the observation that the reconstruction error of NMF is highly sensitive to the initial conditions of the factorization. This approach is grounded in a multi-resolution perspective by considering the stability of a rank's residual to its initial conditions. By doing so, we look to open up new avenues for interpreting NMF results, especially on complex datasets where a single rank may not be able to capture all relevant information.

Our method is designed to be general and applicable to a wide range of datasets, without requiring domain-specific knowledge or assumptions. Other methods, such as consensus-matrix methods, self-comparison methods, and cross-validation based approaches, have been proposed in the literature to determine the rank of NMF. The vast majority of rank selection techniques have a strong preference for lower ranks, which may not always be appropriate for the data at hand. Our method on the other hand, does not have an algorithmic bias for lower ranks and is designed to suggest multiple ranks. Our methodology is applicable across a wide range of domains and does not rely on domain-specific parameters or *a priori* assumptions about the distribution of the data.

This paper is organized as follows. Relevant background information on NMF and the methods against which we compare are given in section 2. A detailed description of our method is given in section 3. A high-level description of the datasets we compare on is given in section 4. Our experimental setup is given in section 5. The results are given in section 6. Finally, a discussion and conclusion is given in section 7.

2 Background

2.1 Non-negative Matrix Factorization

NMF is a low-rank matrix decomposition of an $m \times n$ non-negative matrix, $\boldsymbol{A}$, in which non-negativity constraints are imposed in computation of the lower rank factor matrices. Given a rank k decomposition, the factor matrices $\boldsymbol{W}$ and $\boldsymbol{H}$ are $m \times k$ and $k \times n$ in dimension, respectively. Although a variety methods for solving for $\boldsymbol{W}$ and $\boldsymbol{H}$ exist, such as Hierarchical Alternating Least Squares (Kimura et al., 2015; Gillis & Glineur, 2011) or Gradient Descent (Lee & Seung, 2000), we use Sequential Coordinate Descent (SCD) (Franc et al., 2005; Lin, 2007; Hsieh & Dhillon, 2011) and Multiplicative Update (MU) (Lee & Seung, 2000; Lin, 2007) exclusively in this study . Additionally, although there exists a variety of objective functions such as Kullback-Leibler divergence (Lee & Seung, 2000) and Itakura-Saito divergence (Févotte et al., 2009), to allow for easier comparison with other work in this field, we minimize the Euclidean distance between $\boldsymbol{A}$ and the reconstruction, formulated as

$$\frac{1}{2} \|\boldsymbol{A} - \boldsymbol{W}\boldsymbol{H}\|_F^2 . \tag{1}$$

Given this minimization problem, which is subject to non-negativity constraints, and letting $\boldsymbol{B}_W = \boldsymbol{A}^T\boldsymbol{W}$, $\boldsymbol{B}_H = \boldsymbol{A}\boldsymbol{H}^T$, $\boldsymbol{G}_W$ be the gram matrix of $\boldsymbol{W}$, and $\boldsymbol{G}_H$ be the gram matrix of $\boldsymbol{H}^T$, the SCD update rules can be written in vector form as

$$\boldsymbol{H}_{i,:} \longleftarrow \max\left(0, \boldsymbol{H}_{i,:} + \frac{(\boldsymbol{B}_W)_{i,:} - (\boldsymbol{H}^T\boldsymbol{G}_W)_{i,:}}{(\boldsymbol{G}_W)_{i,i}}\right),$$

for all $i \in \{1, 2, \ldots, k\}$, and

$$\boldsymbol{W}_{:,j} \longleftarrow \max\left(0, \boldsymbol{W}_{:,j} + \frac{(\boldsymbol{B}_H)_{:,j} - (\boldsymbol{W}\boldsymbol{G}_H)_{:,j}}{(\boldsymbol{G}_H)_{j,j}}\right),$$

for all $j \in \{1, 2, \ldots, k\}$ (Lin, 2007; Hsieh & Dhillon, 2011; Franc et al., 2005). Note that the max function is applied element-wise. Similarly, the MU rules can be written as

$$\begin{aligned} \boldsymbol{H} &\longleftarrow \boldsymbol{H} \odot \left(\boldsymbol{B}_W^T \oslash (\boldsymbol{G}_W\boldsymbol{H})\right) \\ \boldsymbol{W} &\longleftarrow \boldsymbol{W} \odot \left(\boldsymbol{B}_H \oslash (\boldsymbol{W}\boldsymbol{G}_H)\right), \end{aligned}$$

where $\odot$ and $\oslash$ denote the Hadamard product and division, respectively (Lee & Seung, 2000; Lin, 2007).

Under the view of NMF as an algorithm which produces a clustering, the matrix $\boldsymbol{W}$ assigns weights to basis vectors in $\boldsymbol{H}$, indicating how strongly each data point is associated with different clusters. These clusters, which are represented by the basis vectors of $\boldsymbol{H}$, often group similar data points together in a meaningful way.

2.2 Limitations of Non-negative Matrix Factorizations

While NMF has been shown to be useful due to its ability to extract meaningful information, it has shortcomings. First, the NMF factorization is not unique, making the problem of computing the NMF ill-posed (Gillis, 2012). There are typically numerous equally as good solutions (e.g. factorizations for which the Frobenius norm of the residual are equally small). Second, it is well understood that NMF is highly sensitive to the initial conditions of the problem, and considerable work has gone into getting around this problem, though it largely domain-dependent (Rosales et al., 2016; Yang et al., 2021; Devarajan, 2008) Third, the underlying optimization problem is generally non-convex, meaning there is no guarantee that the obtained solution is a global minimum (e.g. there is no guarantee that the obtained factorization is the best factorization).

Fundamentally, NMF is a low-rank decomposition, where the factorization can often be meaningfully interpreted as clustering. However, without additional constraints or sparsity enforcing conditions, the NMF is not a hard clustering algorithm (Kim & Park, 2008). Whether used for a soft clustering or not, in order for the factorization to be useful, it is crucial to know which rank decompositions are meaningful. Significant work has been done to this effect (as discussed in subsection 2.3, with most studies focusing on classification datasets (for which class labels are known). Typically, this is to ensure that there is a "true rank". While we adopt this convention for consistency and for lack of any better evaluation to date, it is crucial to recognize that NMF is not a classification algorithm (e.g. it is unsupervised and has no knowledge of the underlying classes). While it is often used for clustering (and it is not uncommon to evaluate clustering with classification data), fundamentally the decomposition of basis vectors, even at the same rank as the number of classes, may have no correlation with the classes considered as the true classes and may be meaningless.

2.3 Previous Work on Rank Determination

A variety of methods to determine the rank have been proposed in the literature. These can largely be broken into three main categories: consensus-matrix methods, self-comparison methods, and cross-validation based approaches. A brief overview for each of these methods is provided in this section and further details for the implementations we use for all methods we compare against are given in section 5 .

2.3.1 Consensus-Matrix Methods

Cophenetic correlation and dispersion coefficient both compute a consensus matrix based on the clustering obtained with NMF, then compute their metric based on this matrix (Kim & Park, 2007; Brunet et al., 2004). In both cases, the consensus matrix is computed and then averaged over a number of random starts.

After computing the averaged consensus matrix, both methods compute their metric. Cophenetic correlation then chooses a rank based on when the cophenetic correlation first begins to drop (Brunet et al., 2004). However, it is important to note that what constitutes a drop is dependent on how many ranks are plotted as this affects the range of the y-axis and consequently what appears to be a drop. Alternatively, the dispersion coefficient method chooses a rank based on when the dispersion coefficient is maximized (Kim & Park, 2007).

In the literature, both methods are often run on a relatively small range of ranks around the point at which the authors expect the optimal rank to be. With cophenetic correlation, the optimal rank is chosen based on an extremely small drop in the coefficient over the range. In fact, one run in Brunet et al. (2004) was shown to be inconclusive based on the lack of a drop on the short range of ranks tested. In our testing, we find the behavior of cophenetic correlation to be erratic on all of our datasets when testing higher ranks than was tested in the original implementation. Similarly, we find the dispersion coefficient to be generally increasing as a function of rank after the initial drop. In both cases, these metrics leave the user unsure of what to pick unless choosing to run on a limited range of ranks. It is not always possible to determine an appropriately small range to check, especially when the rank of the underlying data cannot be determined *a priori*. Furthermore, the number of ranks included when plotting the cophenetic coefficient changes how stretched the plot is left-to-right between consecutive ranks, which drastically affects how steep a drop appears to be.

2.3.2 Self-Comparison Methods

Self-comparison methods can be subset into two categories, split validation and permutation comparison. Split validation cuts the input matrix in half randomly, reorders the halves by the similarity of the basis vectors, then computes the similarity between the two halves. This similarity metric can take a few forms but has shown success in the past using adjusted Rand index (ARI) (Hubert & Arabie, 1985; Grossberger et al., 2018) and inner product (Sotiras et al., 2017). We choose not to compare against inner product as it had poor performance on real data in the testing performed by Muzzarelli et al. (2019).

Permutation compares the slope of the elbow of the reconstruction error of the factorization of $\boldsymbol{A}$ against the slope of the elbow of the reconstruction error of a permuted version of $\boldsymbol{A}$. Effectively, this compares the ability of NMF to reconstruct the dataset against the ability of NMF to reconstruct a random matrix of exactly the same magnitude as the original matrix. Effectively, when the slope of the reconstruction error of $\boldsymbol{A}$ is equal to that or greater than the slope of the permuted matrix, no extra information is able to be extracted from the original dataset compared to a random one.

2.3.3 Non-Categorized Methods

The elbow method is a popular technique for rank determination used in cluster analysis. This method involves plotting the residual as a function of k and picking the elbow of the curve as the correct number of clusters to use. The elbow in the graph is where the rate of decrease changes, representing the point at which increasing the number of clusters does not significantly improve the fit of the model. Although the elbow method is at least partially subjective, we compare against it for completeness.

Akaike information criterion (AIC) was successful in determining rank on time-series data in Cheung et al. (2015) using a modified implementation of NMF. We choose not to compare against AIC as it had poor performance in Gilad et al. (2020) and was otherwise criticized by Ito et al. (2016) for requiring assumptions that do not necessarily hold in NMF. In this study, we focus on a general approach that does not make statistical assumptions regarding NMF or the underlying data.

Similarly, we do not compare against Cai et al. (2022), a sequential hypothesis testing method, due to their requirement that the underlying data follows certain distributions.

For similar reasons, we do not compare against the category of Bayesian methods due to their requirements for *a priori* knowledge of the distribution for prior estimation (Schmidt et al., 2009; Cemgil, 2009).

Additionally, we do not compare against methods utilizing minimum description length (MDL) as they assume a statistical model of the NMF Yamauchi et al. (2012).

Relevance determination is worthy of mention here but is ultimately not relevant to the discussion in this paper. Essentially, these methods identify relevant clusters given a larger rank NMF decomposition Tan & Févotte (2009).

2.3.4 Cross-Validation-Based Approaches

A variety of methods for NMF rank determination using cross-validation (CV) have been proposed in the literature. These include bi-CV by Owen & Perry (2009), and imputation-based CV by Kanagal & Sindhwani (2010). We choose not to compare against bi-CV as the results have been shown in practice to be unclear or unstable by Kanagal & Sindhwani (2010); Gilad et al. (2020).

Imputation-based CV is performed by optimizing for $\boldsymbol{W}$ and $\boldsymbol{H}$ given an imputed version of $\boldsymbol{A}$ in which a percentage of values are denoted as missing (Kanagal & Sindhwani, 2010). The use of a Wold holdout pattern has been shown to be performant and is most widely used (Wold, 1978; Kanagal & Sindhwani, 2010). We denote withheld values as 1 in the binary masking matrix, $\boldsymbol{M}$, and 0 otherwise. These values are hidden from computation during optimization. In most implementations, the reconstruction error is afterward calculated as

$$\frac{\|\boldsymbol{M} \odot (\boldsymbol{A} - \boldsymbol{W}\boldsymbol{H})\|_F^2}{\|\boldsymbol{M}\|_F^2}. \tag{2}$$

In the implementation originally proposed by Kanagal & Sindhwani (2010), this is performed multiple times per rank and then averaged; the rank with lowest mean reconstruction error is chosen. For the purpose of comparison in this study, we will call this method KS-CV, based on the last initials of the authors.

A variety of methods have been proposed that directly build on the work of Kanagal & Sindhwani (2010) such as MADImput, MSEImput, and CV2K (Muzzarelli et al., 2019; Gilad et al., 2020). Each of these three methods optimizes over every rank of interest a number of times, as was performed by Kanagal & Sindhwani (2010). For MADImput, the Median Absolute Deviation (MAD) of the reconstruction errors is calculated at each rank and the rank with lowest MAD is chosen (Muzzarelli et al., 2019). We choose not to compare against MSEImput because it performed poorly on all but simulated data (Muzzarelli et al., 2019). Differing from the other imputation methods described, the authors of Gilad et al. (2020) compute the error as the L^1-norm of the error over the masked values, computed against a normalized version of the initial matrix which allows them to normalize both $\boldsymbol{W}$ and $\boldsymbol{H}$, as detailed in Algorithm 2 in their paper. The rank with minimum median reconstruction error calculated as stated is chosen, but is adjusted down based on a correction step determined by a Wilcoxen rank-sum test Gilad et al. (2020).

2.4 Complexity

The majority of rank determination methods compute NMF using a standard optimization algorithm a number of times, then compute their metric. While the cost of computing these metrics is not free, it is typically substantially less than the cost of computing the NMF decomposition. On the other hand, imputation-based methods must deal with missing values during the computation of the NMF decomposition, fundamentally altering how the NMF decomposition is computed. The complexity of imputation-based CV methods is often overlooked but is a significant burden in practice.

Before discussing the complexity of computing an NMF decomposition with missing values, we must first discuss the complexity of a standard NMF decomposition. As is common practice, we only count multiplication and division, and we assume naive algorithms for common operations such as matrix multiplication. The amount of work performed at a given rank for a single optimization iteration using SCD when no values are missing is described by

$$2mnk + 2mk^2 + 2nk^2. \tag{3}$$

This assumes two Gram matrix computations, one computation of $\boldsymbol{B}_W$ and $\boldsymbol{B}_H$, and one computation of $\boldsymbol{W}\boldsymbol{G}_H$ and $\boldsymbol{H}^T\boldsymbol{G}_W$. Under the assumption that $k \ll \min(m, n)$, the time complexity is $\mathcal{O}(mnk)$. The amount of work required for a single iteration of NMF with MU is slightly more, but results in the same overall time complexity.

The computation of CV with missing values is considerably more complex. The implementation provided by Lin & Boutros (2020), and used by Gilad et al. (2020), implements imputation-based CV by creating a new Gram matrix for each row or column affected by missing values. That is to say that each column, $\boldsymbol{H}_{:,j}$, in the update of $\boldsymbol{H}$ requires a different Gram matrix, denoted by $\boldsymbol{G}_W^{(j)}$, and created as

$$\begin{aligned}\boldsymbol{G}_W^{(j)} &= \left(\text{diag}(\sim\boldsymbol{M}_{:,j})\boldsymbol{W}\right)^T \left(\text{diag}(\sim\boldsymbol{M}_{:,j})\boldsymbol{W}\right) \\ &= \boldsymbol{W}^T\text{diag}(\sim\boldsymbol{M}_{:,j})\boldsymbol{W}.\end{aligned}$$

Similarly, each row, $\boldsymbol{W}_{i,:}$, in the update of $\boldsymbol{W}$ requires a different Gram matrix, $\boldsymbol{G}_H^{(i)}$, which is created as

$$\begin{aligned}\boldsymbol{G}_H^{(i)} &= \left(\boldsymbol{H}\text{diag}(\sim\boldsymbol{M}_{i,:})\right)\left(\boldsymbol{H}\text{diag}(\sim\boldsymbol{M}_{i,:})\right)^T \\ &= \boldsymbol{H}\text{diag}(\sim\boldsymbol{M}_{i,:})\boldsymbol{H}^T,\end{aligned}$$

where $\sim$ denotes element-wise logical negation. Taking into account the need to compute all of these Gram matrices, the missing value computation cost per iteration is captured by

$$2mnk^2 + 2mnk + mk^2 + nk^2. \tag{4}$$

This assumes the computation of $m+n$ Gram matrices, one computation of $\boldsymbol{B}_W$ and $\boldsymbol{B}_H$, one computation of $\boldsymbol{W}_{i,:}\boldsymbol{G}_H^{(i)}$ for all $i \in \{1, 2, \ldots, m\}$ and $(\boldsymbol{H}_{:,j})^T\boldsymbol{G}_W^{(j)}$ for all $j \in \{1, 2, \ldots, n\}$, and at least one missing value per

row and column. This gives a time complexity of $\mathcal{O}(mnk^2)$, but attention should be given to the two equations which model their behavior. The dominating term of Equation 3 is contained within Equation 4, meaning that nearly all of the work required to compute the original factorization must be performed in addition to the additional gram matrix related work. For datasets of relatively small size, this is not particularly burdensome, but it is entirely unfeasible on larger datasets as discussed later in subsection 6.3. It should be noted that the CV2K implementation from Gilad et al. (2020) has lower time complexity but necessitates the storage of five additional matrices of equal size to $\boldsymbol{A}$; this is similarly burdensome for matrices of sufficient size.

3 Residual Sensitivity to Initial Conditions (RSIC)

As previously described in subsection 2.2, NMF is highly sensitive to the initial conditions of the $\boldsymbol{W}$ and $\boldsymbol{H}$ matrices.

We have found that, even amongst factorizations whose residuals have comparable Frobenius norms, the reconstruction error computed at any point may be arbitrarily worse or better in any given factorization. This observation highlights the inherent unpredictability in NMF due to random initializations and can be seen clearly in Figure 1. This figure shows the delta between the smallest and largest error at each point in the rank-10 reconstruction of the first face in the Faces dataset (described in detail later in subsubsection 4.2.3). The maximum delta figure is plotted over the models which had within 10% Frobenius norm of the residual of the median residual model. This figure shows that, even within models that have comparable norms, there exists massive deviation in the reconstruction error when considered coordinatewise. Similar behavior was found to be present in all of the datasets that were tested.

Of interest, these observations reveal a pattern within the variability: at certain ranks, the sensitivity to initial conditions as measured by the deltas in reconstruction error diminishes greatly, forming what can be described as "islands of stability". These ranks, where the variance in reconstruction error is minimal despite different initializations, stand out against nearby ranks that exhibit high sensitivity.

We hypothesize that these "islands of stability" are not merely random occurrences but could be indicative of inherent structure or patterns within the data that are particularly well-captured by NMF at these specific ranks. These stable ranks may correspond with factorizations that more effectively distill the essential features of the data and are less influenced by the initial conditions.

To investigate this, we measure this by computing a number of factorizations at each rank, where the randomly generated initial conditions are related across ranks in a scheme described next in subsection 3.1. We then attempt to quantify the Residual Sensitivity to Initial Conditions (RSIC) at a given rank using the metric developed in subsection 3.2. Plotted as a function of the rank, k, we then identify "islands of stability" as potential ranks of interest that should be further investigated.

3.1 Progressive Random Initialization

We implement a progressive random initialization scheme, which allows individual random initializations to be related across ranks, smoothing out variations in reconstruction error across ranks of the NMF factorization for a single progressive random initialization. For a given initialization with maximum rank of interest, $k_{\max}$, let $\mathbf{W}_{\text{init}}$ and $\mathbf{H}_{\text{init}}$ be random matrices of dimension $m \times k_{\max}$ and $k_{\max} \times n$, respectively, whose entries are generated from the uniform distribution on the half interval $[0, 1.0)$. Assume a is the desired number of initializations per rank. Let $\mathbf{W}_{\text{init}}^{(r)}$ be the r-th random initialization of $\mathbf{W}_{\text{init}}$, and $\mathbf{W}_{\text{init}}^{(r,k)}$ is a copy of the left submatrix $\left(\mathbf{W}_{\text{init}}^{(r)}\right)_{:,:k}$.

Likewise for $\mathbf{H}_{\text{init}}$, let $\mathbf{H}_{\text{init}}^{(r,k)}$ be a copy of the upper submatrix $\left(\mathbf{H}_{\text{init}}^{(r)}\right)_{:k,:}$. Then, for a given initialization, r, and letting $k_{\min}$ be the minimum rank of interest, we have the ordered set of tuples of initial matrices,

$$\mathbb{S}_{\text{init}}^{(r)} = \left\{ \left(\mathbf{W}_{\text{init}}^{(r,k_{\min})}, \mathbf{H}_{\text{init}}^{(r,k_{\min})}\right), \left(\mathbf{W}_{\text{init}}^{(r,k_{\min}+1)}, \mathbf{H}_{\text{init}}^{(r,k_{\min}+1)}\right), \ldots, \left(\mathbf{W}_{\text{init}}^{(r,k_{\max})}, \mathbf{H}_{\text{init}}^{(r,k_{\max})}\right) \right\}.$$

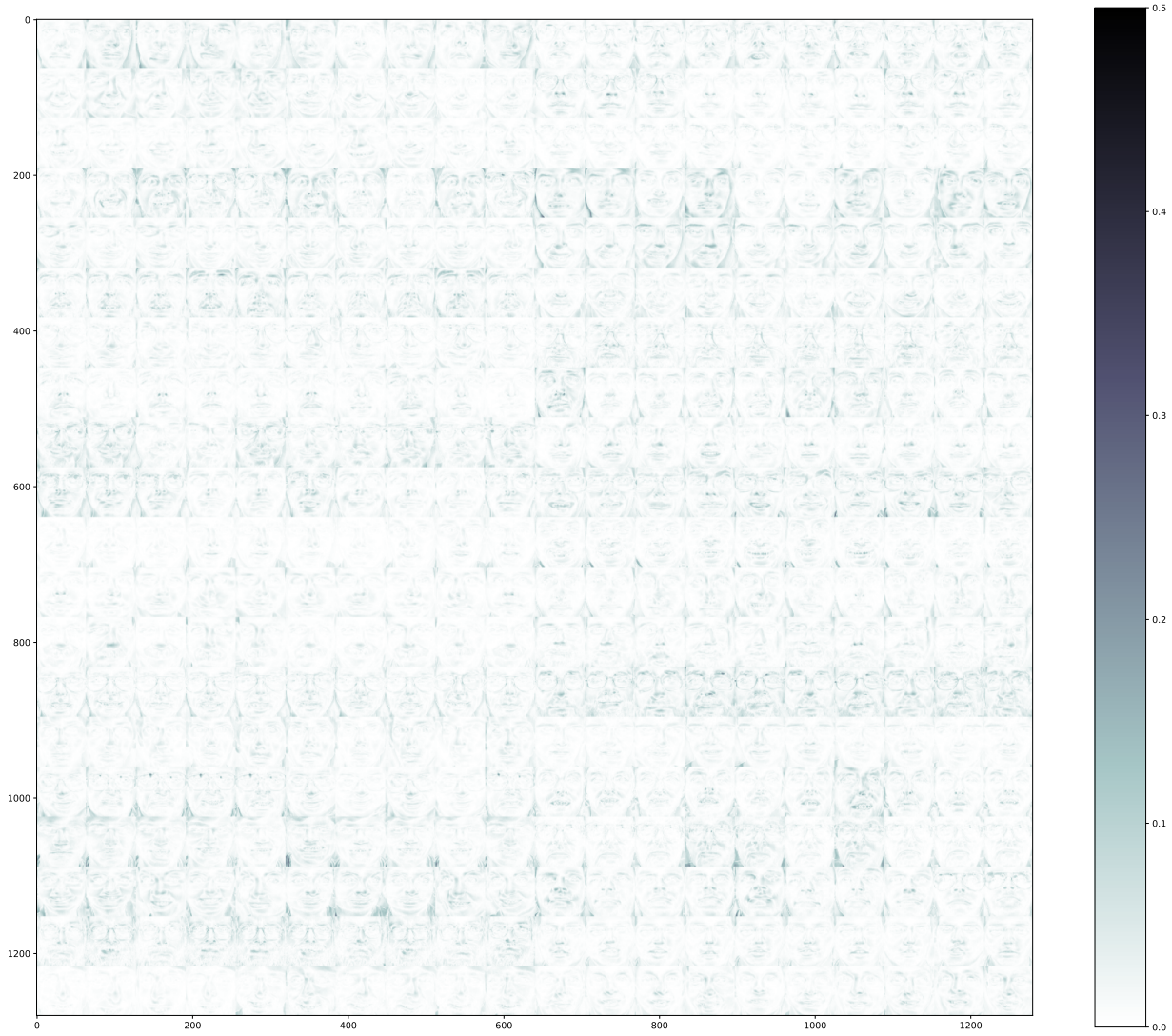


Figure 1: The delta between the smallest and largest error at each point in the rank-10 reconstruction of the first face in the Faces dataset.

Assume a set, $\mathbb{S}_{\text{init}}^{(*)}$, exists for each progressive random initialization we are performing. For simplicity, we index these sets as

$$\mathbb{S}_{\text{init}}^{(r,k)} = \left(\mathbf{W}_{\text{init}}^{(r,k)}, \mathbf{H}_{\text{init}}^{(r,k}\right).$$

Given this indexing, this specifically means that the matrices in $\mathbb{S}_{\text{init}}^{(*,k-1)}$ contain a copy of the respective matrices in $\mathbb{S}_{\text{init}}^{(*,k)}$ for all $k \in \{k_{\min}+1, k_{\min}+2, \ldots, k_{\max}\}$. This is such that $\mathbf{W}_{\text{init}}^{(*,k-1)}$ contains a copy of the left submatrices of all but the last column of $\mathbf{W}_{\text{init}}^{(*,k)}$ and $\mathbf{H}_{\text{init}}^{(*,k-1)}$ contains a copy of the upper submatrices of all but the last row of $\mathbf{H}_{\text{init}}^{(*,k)}$.

Finally, let

$$\mathbb{S}_{opt}^{(r,k)} \longleftarrow \text{NMF}\left(\boldsymbol{A}, \mathbb{S}_{init}^{(r,k)}\right)$$

be the resulting factorization of $\boldsymbol{A}$ after performing NMF with a given rank and initialization pairing. This factorization transforms the unoptimized matrices, giving their optimized counterparts. This is performed over all progressive random initializations and ranks, giving $\mathbb{S}_{\text{opt}}$.

3.2 Mean Coordinatewise Interquartile Range (MCI)

The goal of this metric is to capture the sensitivity of the relative reconstruction error at a given rank by measuring the average spread of relative reconstruction errors at each point. This is computed by taking the interquartile range (IQR) of the relative reconstruction error at each point over the number of initializations.

Let $\boldsymbol{R}^{(k)}$ be a matrix of dimension $a \times mn$, and assume $\text{vec}\,(*)$ flattens an $m \times n$ matrix to a mn dimension vector in any consistent ordering. The values of $\boldsymbol{R}^{(k)}$ are given by

$$\boldsymbol{R}^{(k)}_{r,:} = \text{vec}\left(\boldsymbol{A} - \boldsymbol{W}^{(r,k)}_{\text{opt}}\boldsymbol{H}^{(r,k)}_{\text{opt}}\right),$$

for all $r \in \{1, 2, \ldots, a\}$ with given rank, k. The Mean Coordinatewise IQR (MCI) at each rank, k, may now be computed as

$$\text{MCI}^{(k)} = \frac{1}{mn}\sum_{j=1}^{mn} \text{IQR}\left(\boldsymbol{R}^{(k)}_{:,j}\right),$$

where IQR computes the interquartile range of a vector. This is performed for all $k \in \{k_{\min}, k_{\min} + 1, \ldots, k_{\max}\}$.

Computing the MCI over a large number of factorizations requires the storage of all of these factorizations. Although this may be a significant amount of space for large matrices, the total amount of work required is significantly less than those methods based on cross-validation techniques. Additionally, the post-processing step is able to be computed in batches in a trivial manner, allowing for computation on machines with lower amounts of memory.

4 Datasets

We define the "true rank" of a dataset as the number of classes in the underlying dataset. Additionally, certain datasets cannot be designated a single true rank as sub-classes may exist which equally make sense to target in the analysis of a dataset. We discuss any potential sub-classes in our discussion of each dataset. This does not necessarily correlate directly to the optimal rank of an NMF decomposition. We compare our method on eight datasets from three different disciplines. We break this section into three subsections, one for each type of dataset used. All datasets are formatted such that each row represents a sample and the columns within a row are features. This means that the two single cell datasets, introduced next, are transposed from how they are generally presented in the literature.

4.1 Single Cell Datasets

ALL-AML was originally described in Golub et al. (1999) and retrieved using the package provided by Gaujoux & Seoighe (2023). This dataset is 38×5000 in dimension, consisting of the 5000 most highly varying human genes in the original dataset and taken from 38 bone marrow samples Golub et al. (1999); Gaujoux & Seoighe (2023). Of these 38 samples, 27 are related to acute lymphoblastic leukemia (ALL) and 11 acute myeloid leukemia (AML) Golub et al. (1999). Given the existence of two cancer types, the true rank is 2, though most authors find a rank 3 NMF decomposition to be more informative (Brunet et al., 2004).

The PBMC3K dataset was retrieved from 10x Genomics (2016) and has dimension 2700×13714. This dataset consists of 2700 cells and 13714 genes and comes pre-filtered for relevance. This dataset has no consensus true rank but has been shown using domain knowledge to contain 9 distinct cell types (Du et al., 2020), which will be treated as the true rank for our purposes.

4.2 Image-Based Datasets

4.2.1 Swimmer

The Swimmer dataset was first described in Donoho & Stodden (2003) and retrieved from the package released by Tan & Févotte (2009).[1] Swimmer has dimension 256×1024 and is a synthetic dataset consisting of 256 flattened 32×32 black and white stick figure images meant to mimic a variety of breaststroke positions. There are a total of 16 limb positions, giving a true rank of 16.

4.2.2 Full Digits & Dig0246

We retrieved the "Optical Recognition of Handwritten Digits" dataset by Alpaydin & Kaynak (1998) from the package by Pedregosa et al. (2011). We use this both as the full dataset and additionally use a selected subset for experimentation. We call the full dataset Full Digits and the subset Dig0246, containing only the digits $\{0, 2, 4, 6\}$. We subset in this manner for the sake of comparison and consistency with the authors of Muzzarelli et al. (2019), who did not run on the full dataset. We should note that the clustering behavior of NMF appears to more clearly separate the classes in Dig0246 than with Full Digits. An example of this clustering behavior will be shown later in subsection 6.2. The true ranks are 10 and 4 for Full Digits and Dig0246, respectively.

4.2.3 Faces

We retrieved the "AT&T Laboratories Cambridge Faces" dataset by Cambridge (1994) using the package provided by Pedregosa et al. (2011). This dataset has dimension 400×4096, consisting of 400 flattened black and white images of size 64×64. The dataset consists of 10 people, with 40 images from each. Given the number of subjects, the true rank is 10. The images were originally 92×112 in dimension but have been modified in the version retrieved from Pedregosa et al. (2011).

4.3 Text-Based Datasets

All text datasets were transformed using a bag-of-words model, `CountVectorizer`, provided by Pedregosa et al. (2011). The maximum document frequency was set to 95%, minimum document frequency to an integer value of 2, and the stop-words set to `english` as provided by the implementation. The maximum features was set to 4000 and 30000 for NewsGroup4000 and Web of Science, respectively. All other options were left to their defaults.

4.3.1 NewsGroup4000

We obtained the NewsGroup4000 dataset, originally found at (Lang, 1997), using the packaged provided by Pedregosa et al. (2011). This dataset is of dimension 11314×4000 and contains 20 topics ranging from sports to medical. Given the number of topics, the true rank is 20.

4.3.2 Web of Science

We obtained the Web of Science dataset, which was originally described in Kowsari et al. (2017) from Kowsari et al. (2019). The data was preprocessed as previously described using `CountVectorizer`. This dataset is of dimension 11967×28095 and contains 11967 documents from 35 categories and 7 parent categories. Fundamentally, there are two different true ranks in this dataset, 35 and 7.

5 Experimental Setup

We opt to use only publicly available packages for all significant computations. Due to the large variety of datasets and methods we compare, for computational feasibility, we perform 100 random initializations for each method on each dataset. For each dataset, we let $k_{\min} = 2$ and $k_{\max} = \min(m, n, 64)$ and perform

[1]The package may be found at `www.irit.fr/~Cedric.Fevotte/extras/pami13/ardnmf.zip`

each method on each rank between $k_{\min}$ and $k_{\max}$ inclusively. Since it has been shown that MU converges to a solution more slowly than SCD (Lin & Boutros, 2020), SCD was used as the optimization routine for all methods in which the option was available. We force all methods using SCD to run to 100 optimization iterations per initialization and rank by setting the tolerance to 1e−16. For the remaining methods in which SCD was not an option, MU was selected and forced to run for 500 optimization iterations due to its slower convergence. For consistency with Muzzarelli et al. (2019), all methods optimized the Frobenius norm defined in Equation 1. Before the start of computation for each method and dataset, the random state was set to the seed 123456789. For the computation performed with the packages provided by Gilad et al. (2020); Lin & Boutros (2020), which offer multi-threading support, computation was performed on a dedicated workstation with a 32 core Threadripper processor and 64 GB of memory. All other computation was performed sequentially on a consumer-grade Intel processor.

5.1 Elbow

For the comparison against the elbow method, we determine the elbow based on the average reconstruction error of each rank. This average is computed based on the progressive random initialization scheme described earlier. The results are provided based on a visual determination of where the bend appears to be.

5.2 Cophenetic Correlation & Dispersion Coefficient

For the comparison with cophenetic correlation and dispersion coefficient methods, (Brunet et al., 2004; Kim & Park, 2007), the NIMFA package provided by Zitnik & Zupan (2012) was used. This package does not have support for SCD and MU was used instead. The initialization type was set to `random`, number of initializations to 100, number of iterations to 500, the update to `euclidean`, and objective function to `fro`. We passed in the range of ranks from $k_{\min}$ to $k_{\max}$, and plotted the results, which are the cophenetic correlation coefficient and dispersion coefficient as determined by the package. The optimal rank for each is then chosen based on the criteria set forth by the original authors in Brunet et al. (2004) and Kim & Park (2007) and briefly described in this paper in subsubsection 2.3.1.

5.3 Permutation

For the comparison with permutation, we performed the optimization of NMF using the implementation provided by Pedregosa et al. (2011). We shuffled the the columns individually for each row using the package provided by Harris et al. (2020), then optimized the permuted matrix separately. The Frobenius norm of both the permuted and non-permuted matrix was computed as a function of rank. Using SciPy's implementation of the Savitzky-Golay filter Virtanen et al. (2020), we approximate the slope of the residuals as a function of rank. We look for the point where the reconstruction error of the unpermuted matrix decreases less sharply than that of the permuted matrix. This corresponds to the point at which the slope of the residuals of the non-permuted matrix is greater than or equal that of the slope of the residuals of the permuted matrix because both are decreasing and the slopes are negative. In terms of the approximated derivative, the rank selected is the point immediately before overtaking the elbow of the permuted matrix, or the point at which they are exactly equal. Due to floating point arithmetic, equality is considered a relative tolerance of 1×10^{-8}, computed relative to the larger of two derivative estimates. This is performed 100 times and the median result is computed.

5.4 ARI Method

For the comparison against the ARI method, we randomly split the matrix into two equal parts along the dimension relating to factors. If splitting into even parts is not possible, the larger split is truncated to size. Then NMF is computed on each split using the implementation provided by Pedregosa et al. (2011). The cosine distance between factors is computed using the package provided by Virtanen et al. (2020), giving a distance matrix. This is then passed to the `linear_sum_assignment` function as provided by Virtanen et al. (2020), which implements the Hungarian algorithm described in Crouse (2016). The result is the ARI, which is then averaged over all initializations at a given rank. The rank with highest mean ARI is selected.

We note that we ran with both the mean and median ARI, which resulted in the same decided rank in all cases.

5.5 Cross Validation Methods

For the comparison against KS-CV and MADImput, we use the package provided by Lin & Boutros (2020). The output of this package is the reconstruction error at each rank as previously defined in Equation 2. Using this output, we are able to compute each of these metrics and choose a rank based on where it is minimized. For the comparison against CV2K, we use the package provided by the author Gilad et al. (2020), which returns both a similar output and the chosen rank based on their criteria.

We note that we have chosen to use a different initialization scheme than was used by the authors of Gilad et al. (2020); Muzzarelli et al. (2019). In Muzzarelli et al. (2019), the authors performed 100 runs, initializing 20 times in each run before choosing the best model as defined by reconstruction error. In Gilad et al. (2020), the authors initialize a large number of times, optimizing for hundreds of iterations, and chose the best model defined by reconstruction error to continue. In order to perform this, we would need to run 2000 times for the (Muzzarelli et al., 2019) method and 10000 times for the (Gilad et al., 2020) method.[2] In addition to the considerable length of time required, we do not believe it is fair to arbitrarily run one method an order of magnitude more times than another. Instead, we focus on running all methods for the same number of initializations. For methods that rely on differences across random initializations, this means that they do not get the added benefit of testing against thousands of initializations.

In order to enforce these requirements, a tolerance may be provided to the (Lin & Boutros, 2020) package. For the (Gilad et al., 2020) package, minor modifications to the code are required. The code is modified to force running all 100 iterations and the `init_factor_matrices` function is modified to return the first generated $\boldsymbol{W}$ and $\boldsymbol{H}$.

6 Results

In this section, we present results and compare against other methods. In our analysis of the PBMC3k, NewsGroup4000, and Web of Science datasets, significant computational hurdles were encountered. Due to RAM constraints on our system, we were forced to run CV2k with only 16 threads for both the PBMC3k and NewsGroup4000 datasets. This limitation resulted in an excessively prolonged runtime of over 30 days for PBMC3k and longer for NewsGroup4000. Given the size of the Web of Science dataset, a further reduction in threads would have been necessary, leading to an even longer computational burden.

NNLM-CV was not constrained by memory requirements but was unable to complete even a single run on the PBMC3k dataset in 24 hours, meaning the full 100 runs would take over 100 days. The larger scales of NewsGroup4000 and Web of Science implied that completion times would be significantly longer. This difficulty, owing to the incredible time complexity of the method as described in Equation 4, led to the conclusion that the burden of computation for these methods is infeasible.

Consequently, the result table for these methods on the mentioned datasets are denoted as "N/A". In other instances, where the output was inconclusive, we have designated the results "undetermined" (Und.). For instance, undetermined would be noted when using the elbow method where there is no clear elbow or when using the permutation method but the slope of the permuted is steeper than the slope of the unpermuted data from the beginning. The remaining results are presented either as integer values or in an increasing order set format, which is applicable only to permutation as well as MCI-RSIC as previously described.

6.1 Single Cell Datasets

We show the RSIC-MCI metric as a function of rank on the ALL-AML dataset in Figure 2. The horizontal axis, representing rank, ranges from 1 to 38. Based on our selection criteria described before, we find rank 5

[2] A number of initialization was not described in the CV2K paper but was instead found on the Github linked in the paper.

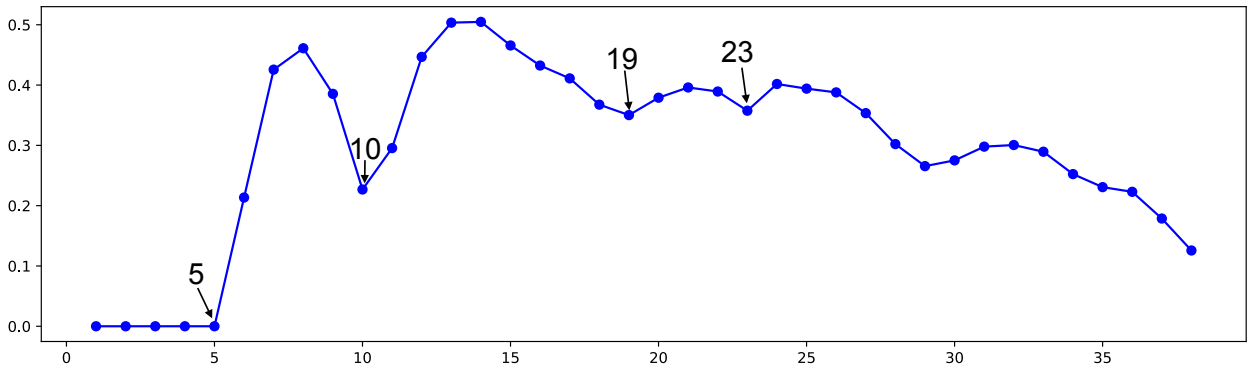


Figure 2: Mean Coordinatewise IQR (MCI) (y-axis) vs rank (x-axis) for the ALL-AML dataset for ranks 1 through 38. We identify ranks 5, 10, 19, and 23 as "islands of stability" and thereby potential ranks of interest.

to be the first island of stablility. Note that ranks 1 through 5 all have the same or similar MCI values, but rank 5 is the first rank before a significant increase in MCI.

We found that our method and permutation performed poorly on the ALL-AML dataset, whereas cophenetic, dispersion, and CV2k all converged on the generally agreed upon rank and ARI and KS-CV converged on the true rank. For PBMC3k, no method returned a solution that contended with the true rank although MCI-RSIC and elbow both returned nearby ranks.

Table 1: Results for all evaluated rank determination methods in comparison to true rank on the ALL-AML and PBMC3K single cell datasets.

Method	ALL-AML	PBMC3k
MCI-RSIC	{5, 10, 19, 23}	{3, 7, 11}
Elbow	4	8
Cophenetic	3	2
Dispersion	3	2
Permutation	5	41
ARI	2	2
KS-CV	2	N/A
CV2K	3	N/A
MADImput	4	N/A
True Rank	2	9^3

6.2 Image-Based Datasets

We show the RSIC-MCI metric as a function of rank on the Full Digits dataset in Figure 3. This image plots the MCI metric from rank 1 to 64. Like the other methods tested, this method performed equivalently poorly on the Full Digits dataset when considering the true rank. There is significant mixing between the ranks, which is indicative of poor clustering behavior.

In addition to the RSIC-MCI metric on Full Digits, we show the metric on Dig0246 in Figure 4. This image plots the RSIC-MCI metric from rank 1 to 64. This clearly shows an island of stability at rank 4 and provides evidence for on at rank 6.

Finally, we show the RSIC-MCI metric as a function of rank on the Swimmer dataset in Figure 5. This image plots the MCI metric from rank 1 to 64 and clearly shows an island of stability at rank 16, which corresponds to the true rank of the dataset.

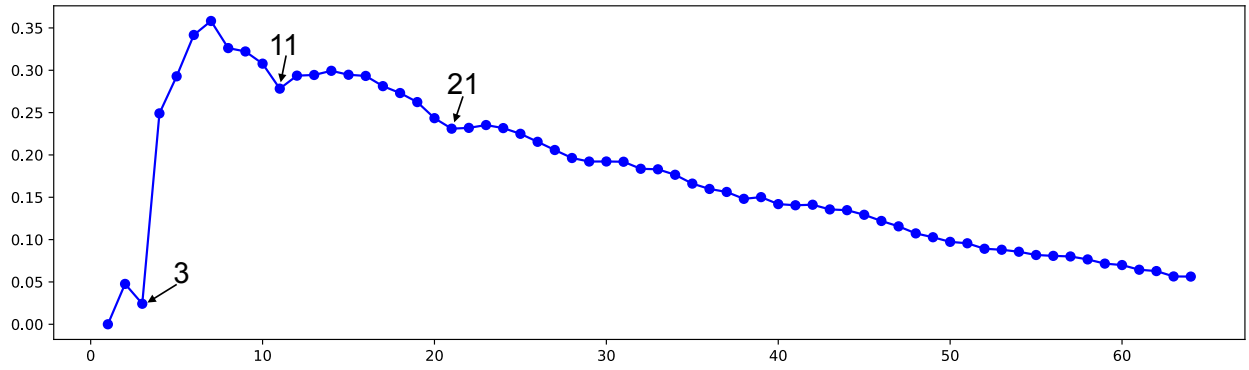


Figure 3: Mean Coordinatewise IQR (MCI) (y-axis) vs rank (x-axis) for the Full Digits dataset for ranks 1 through 64. We identify ranks 3, 11, and 21 as "islands of stability" and thereby potential ranks of interest.

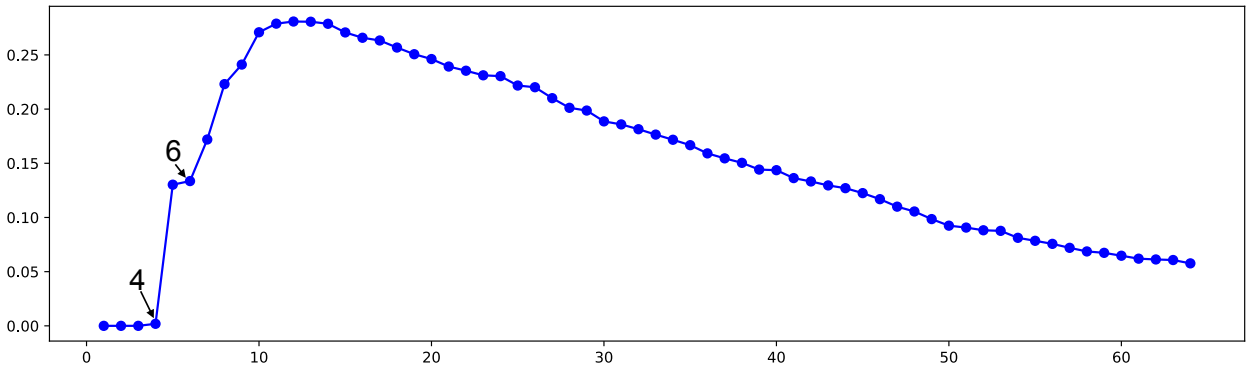


Figure 4: Mean Coordinatewise IQR (MCI) (y-axis) vs rank (x-axis) for the Dig0246 dataset for ranks 1 through 64. We identify ranks 4 and 6 as "islands of stability" and thereby potential ranks of interest.

Figure 6 shows poor clustering behavior at the true rank of 10 on the Full Digits dataset. Additionally, we show the clustering behavior at rank 11, which is a nearby rank that we suggest. This figure shows that the clustering behavior at both ranks is rather poor, and provides evidence for why all methods performed poorly on this dataset. Indeed, there appears to be no clear separation between the classes in the dataset. This poor clustering behavior was present in each rank we tested, and is indicative of the poor performance of all methods on this dataset.

For the image based datasets, MCI-RSIC and elbow returned the true rank on the Swimmer dataset. The permutation method consistently detected a rank greater than the true rank, across all image datasets. For the faces dataset, MCI-RSIC performed poorly in terms of the true rank and only elbow returned the true rank. On Full Digits, all methods performed poorly though the clustering behavior of this dataset is poor as seen in Figure 6. For Dig0246, MCI-RSIC, elbow, cophenetic, dispersion, and ARI all returned the true rank. We note that MADImput found the true rank in their paper Muzzarelli et al. (2019) and the discrepancy here is most likely due to their use of 1900 more initializations than allowed in our study.

6.3 Text-Based Datasets

No method was able to return the true rank for NewsGroup4000 although MCI-RSIC and dispersion both return a nearby rank. The others underestimate more substantially (elbow, cophenetic, dispersion, ARI), are infeasible to run (KS-CV, CV2K, MADImput), or failed to identify any rank (permutation).

For Web of Science, MCI-RSIC returns the true rank for the category as well as a nearby rank to the number of subcategories. Elbow detects a nearby rank for the category. All other methods either substantially

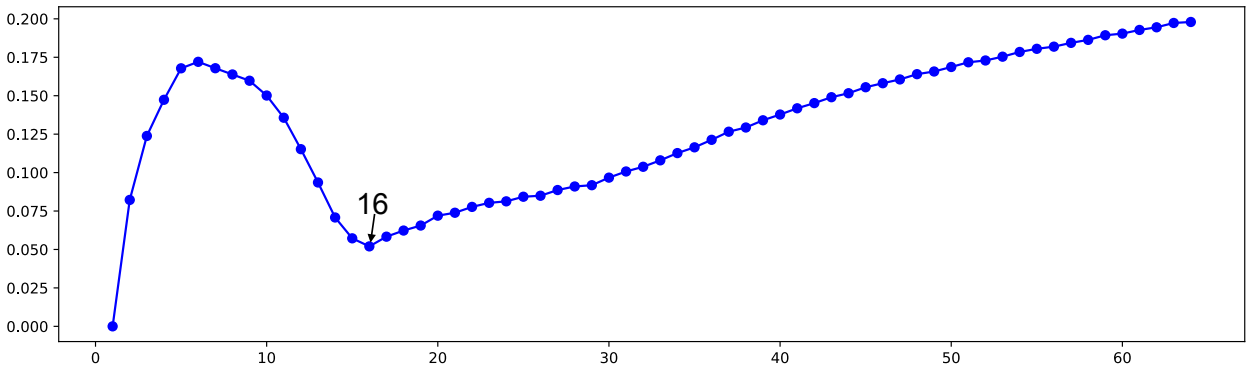
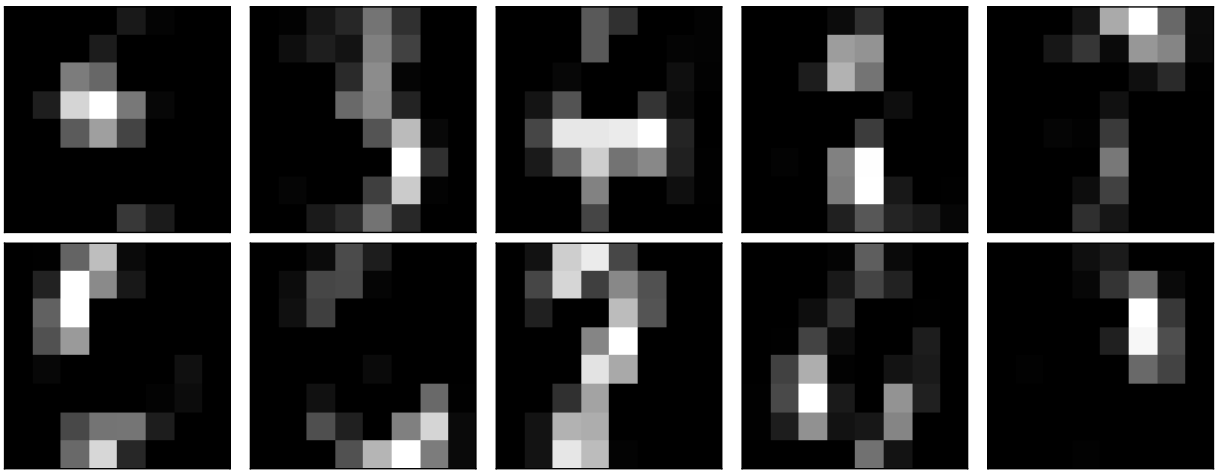
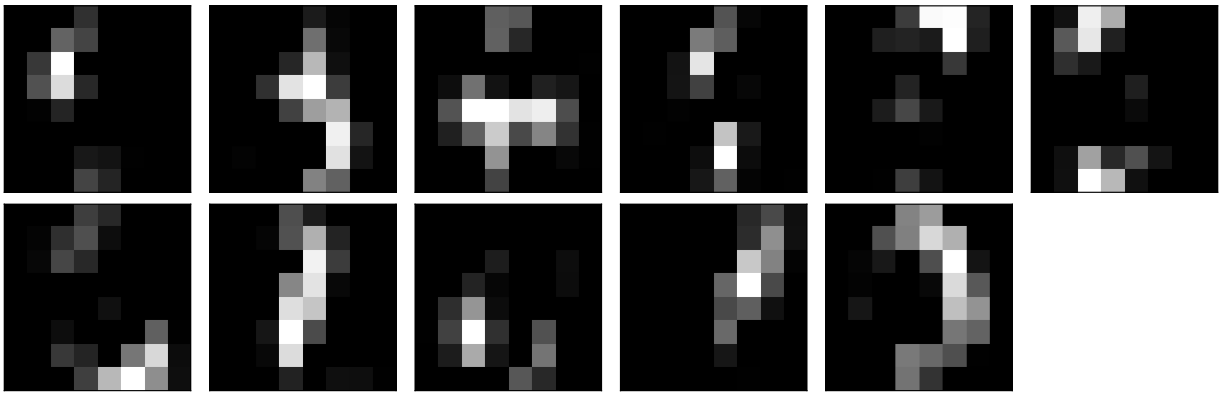


Figure 5: Mean Coordinatewise IQR (MCI) (y-axis) vs rank (x-axis) for the Swimmer dataset for ranks 1 through 64. We identify rank 16 as the only "island of stability" and thereby potential rank of interest.



(a) Rank 10



(b) Rank 11

Figure 6: Clustering behavior of the Full Digits dataset at ranks 10 and 11.

undershoot (cophenetic, dipsersion, ARI), are infeasible to run (KS-CV, CV2K, MADImput), or indicated the rank was greater than the tested range (permutation).

Table 2: Results for all evaluated rank determination methods in comparison to true rank on Swimmer, Faces, Full Digits, and Dig0246 image datasets.

Method	Swimmer	Faces	Full Digits	Dig0246
MCI-RSIC	16	{2, 6, 9, 18}	{3, 11, 21}	{4, 6}
Elbow	16	10	5 (Und.)	4
Cophenetic	Und.	2	2	4
Dispersion	64+	2	2	4
Permutation	18	50	22	17
ARI	13	2	5	4
KS-CV	14	51	12	12
CV2K	17	64	16	16
MADImput	13	60	2	11
True Rank	16	10	10	4

Table 3: Results for all evaluated rank determination methods in comparison to true rank on NewsGroup4000 and Web of Science text datasets.

Method	NewsGroup4000	Web of Science
MCI-RSIC	{4, 12, 24, 46}	{3, 7, 11?, 20, 39}
Elbow	9	6
Cophenetic	5	3
Dispersion	12	3
Permutation	Und.	64+
ARI	3	3
KS-CV	N/A	N/A
CV2K	N/A	N/A
MADImput	N/A	N/A
True Rank	20	{35, 7}

7 Discussion & Conclusion

In this paper, we introduced RSIC, a novel method for determining ranks of interest in NMF. Unlike traditional methods which aim to identify a single optimal rank—often requiring extensive parameter tuning and domain-specific knowledge—our approach identifies multiple, possibly relevant ranks by analyzing the sensitivity of the reconstruction residual to different initial conditions (random initializations in the case of this paper). This allows for a more nuanced understanding of the data's underlying structure and provides some flexibility in exploratory data analysis.

Our method identifies "islands of stability", which are ranks where the NMF solutions are less sensitive to initialization and, therefore, more likely to represent meaningful decompositions of the data. We quantified this stability using the Mean Coordinatewise Interquartile Range (MCI) of the relative reconstruction error across multiple initializations. By doing so, we highlighted ranks that consistently produce stable and interpretable factors, providing insights that single-rank methods may overlook.

We evaluated RSIC on a diverse set of datasets across various domains, including single-cell gene expression data, image datasets, and text corpora. Our experiments demonstrated that RSIC effectively identifies ranks of interest that are consistent or close to the true underlying ranks of the data. Of note, our method performed well on large-scale datasets where other methods tended to undershoot or where cross-validation-based approaches were infeasible due to their high computational complexity.

Comparative analysis with existing methods, including consensus-matrix methods like cophenetic correlation coefficient and dispersion coefficient, self-comparison methods like the adjusted Rand index, and cross-validation approaches, showed that RSIC is competitive and often superior in identifying meaningful ranks.

However, our method is not without limitations. In datasets where the underlying structure is less pronounced or when the data does not exhibit clear stability islands, RSIC may suggest multiple ranks, requiring further analysis to select the most appropriate one. For example, in the ALL-AML dataset, RSIC showed perfect stability for all ranks at or below 5, but we select 5 based on our selection criteria—this is a limitation of our method. Additionally, while our approach reduces the computational burden compared to some methods, it still requires multiple NMF computations across a range of ranks and initializations.

For future work, we plan to refine the RSIC metric to better handle datasets with subtle or hierarchical structures. Incorporating additional criteria, such as sparsity constraints or domain-specific knowledge, may help to further refine the ranks suggested by RSIC. Additionally, we believe that RSIC could benefit from other types of initialization schemes, such as those based on clustering, dimensionality reduction techniques, or other randomization schemes, to further explore the space of possible initializations. Further, the method could benefit from a window-based smoothing (e.g., Savitzky-Golay) of the MCI values to reduce the noise in the output, which could be particularly useful in datasets with high variability in the reconstruction error. We also aim to explore the theoretical underpinnings of the observed islands of stability to provide deeper insights into why certain ranks yield more stable decompositions.

RSIC offers a robust and significantly more scalable approach for rank suggestion in NMF, taking steps toward bridging the gap between the need for meaningful data decompositions and the practical constraints of computational resources. By providing a selection of relevant ranks and highlighting areas of stability, our method empowers practitioners to make more informed decision in exploratory data analysis and dimensionality reduction tasks.

Broader Impact Statement

removed for double blind review

Author Contributions

removed for double blind review

Acknowledgments

removed for double blind review

References

10x Genomics. Peripheral blood mononuclear cells (PBMCs) from a healthy donor (same donor as pbmc6k)., July 2016. URL cf.10xgenomics.com/samples/cell/pbmc3k/pbmc3k_filtered_gene_bc_matrices.tar.gz.

E. Alpaydin and C. Kaynak. Optical recognition of handwritten digits, 1998. URL https://doi.org/10.24432/C50P49.

Jean-Philippe Brunet, Pablo Tamayo, Todd R. Golub, and Jill P. Mesirov. Metagenes and molecular pattern discovery using matrix factorization. *Proceedings of the National Academy of Sciences*, 101(12):4164–4169, March 2004. doi: 10.1073/pnas.0308531101. URL https://doi.org/10.1073/pnas.0308531101.

Yun Cai, Hong Gu, and Toby Kenney. Rank selection for non-negative matrix factorization. 2022. URL https://api.semanticscholar.org/CorpusID:253254815.

AT&T Laboratories Cambridge. The database of faces, April 1994. URL https://cam-orl.co.uk/facedatabase.html.

Ali Taylan Cemgil. Bayesian inference for nonnegative matrix factorisation models. *Computational Intelligence and Neuroscience*, 2009:1–17, 2009. doi: 10.1155/2009/785152. URL https://doi.org/10.1155/2009/785152.

Vincent C. K. Cheung, Karthik Devarajan, Giacomo Severini, Andrea Turolla, and Paolo Bonato. Decomposing time series data by a non-negative matrix factorization algorithm with temporally constrained coefficients. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, August 2015. doi: 10.1109/embc.2015.7319146. URL https://doi.org/10.1109/embc.2015.7319146.

David F. Crouse. On implementing 2D rectangular assignment algorithms. *IEEE Transactions on Aerospace and Electronic Systems*, 52(4):1679–1696, 2016. doi: 10.1109/TAES.2016.140952.

Karthik Devarajan. Nonnegative matrix factorization: An analytical and interpretive tool in computational biology. *PLoS Computational Biology*, 4(7):e1000029, July 2008. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1000029. URL http://dx.doi.org/10.1371/journal.pcbi.1000029.

David Donoho and Victoria Stodden. When does non-negative matrix factorization give a correct decomposition into parts? In S. Thrun, L. Saul, and B. Schölkopf (eds.), *Advances in Neural Information Processing Systems*, volume 16. MIT Press, 2003. URL https://proceedings.neurips.cc/paper_files/paper/2003/file/1843e35d41ccf6e63273495ba42df3c1-Paper.pdf.

Yuheng Du, Qianhui Huang, Cedric Arisdakessian, and Lana X Garmire. Evaluation of STAR and kallisto on single cell RNA-seq data alignment. *G3 Genes|Genomes|Genetics*, 10(5):1775–1783, May 2020. doi: 10.1534/g3.120.401160. URL https://doi.org/10.1534/g3.120.401160.

Cédric Févotte, Nancy Bertin, and Jean-Louis Durrieu. Nonnegative matrix factorization with the Itakura-Saito divergence: With application to music analysis. *Neural Computation*, 21(3):793–830, March 2009. doi: 10.1162/neco.2008.04-08-771. URL https://doi.org/10.1162/neco.2008.04-08-771.

Vojtěch Franc, Václav Hlaváč, and Mirko Navara. Sequential coordinate-wise algorithm for the non-negative least squares problem. In *Computer Analysis of Images and Patterns*, pp. 407–414. Springer Berlin Heidelberg, 2005. doi: 10.1007/11556121_50. URL https://doi.org/10.1007/11556121_50.

Renaud Gaujoux and Cathal Seoighe. A flexible R package for nonnegative matrix factorization. *BMC Bioinformatics*, 11(1), July 2010. ISSN 1471-2105. doi: 10.1186/1471-2105-11-367. URL http://dx.doi.org/10.1186/1471-2105-11-367.

Renaud Gaujoux and Cathal Seoighe. NMF: Algorithms and framework for nonnegative matrix factorization (NMF), 2023. URL https://CRAN.R-project.org/package=NMF.

Gal Gilad, Itay Sason, and Roded Sharan. An automated approach for determining the number of components in non-negative matrix factorization with application to mutational signature learning. *Machine Learning: Science and Technology*, 2(1):015013, December 2020. doi: 10.1088/2632-2153/abc60a. URL https://doi.org/10.1088/2632-2153/abc60a.

Nicolas Gillis. Sparse and unique nonnegative matrix factorization through data preprocessing. *Journal of Machine Learning Research*, 13(108):3349–3386, 2012. URL http://jmlr.org/papers/v13/gillis12a.html.

Nicolas Gillis. Nonnegative matrix factorization. January 2020. doi: 10.1137/1.9781611976410. URL https://doi.org/10.1137/1.9781611976410.

Nicolas Gillis and François Glineur. Accelerated multiplicative updates and hierarchical ALS algorithms for nonnegative matrix factorization. *arXiv*, 2011. doi: 10.48550/ARXIV.1107.5194. URL https://arxiv.org/abs/1107.5194.

T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286(5439):531–537, October 1999. doi: 10.1126/science.286.5439.531. URL https://doi.org/10.1126/science.286.5439.531.

Lukas Grossberger, Francesco P. Battaglia, and Martin Vinck. Unsupervised clustering of temporal patterns in high-dimensional neuronal ensembles using a novel dissimilarity measure. *PLOS Computational Biology*, 14(7):e1006283, July 2018. doi: 10.1371/journal.pcbi.1006283. URL https://doi.org/10.1371/journal.pcbi.1006283.

D. Guillamet, B. Schiele, and J. Vitria. Analyzing non-negative matrix factorization for image classification. In *2002 International Conference on Pattern Recognition*, volume 2, pp. 116–119 vol.2, 2002. doi: 10.1109/ICPR.2002.1048251.

Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020. doi: 10.1038/s41586-020-2649-2. URL https://doi.org/10.1038/s41586-020-2649-2.

Ali Hassani, Amir Iranmanesh, and Najme Mansouri. Text mining using nonnegative matrix factorization and latent semantic analysis, 2019. URL https://arxiv.org/abs/1911.04705.

Cho-Jui Hsieh and Inderjit S. Dhillon. Fast coordinate descent methods with variable selection for non-negative matrix factorization. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, aug 2011.

Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, December 1985. doi: 10.1007/bf01908075. URL https://doi.org/10.1007/bf01908075.

Yu Ito, Shin ichi Oeda, and Kenji Yamanishi. Rank selection for non-negative matrix factorization with normalized maximum likelihood coding. In *Proceedings of the 2016 SIAM International Conference on Data Mining*. Society for Industrial and Applied Mathematics, June 2016. doi: 10.1137/1.9781611974348.81. URL https://doi.org/10.1137/1.9781611974348.81.

Bhargav Kanagal and Vikas Sindhwani. Rank selection in low-rank matrix approximations : A study of cross-validation for nmfs. *Proc Conf Adv Neural Inf Process*, 2010.

Hyunsoo Kim and Haesun Park. Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis. *Bioinformatics*, 23(12):1495–1502, May 2007. doi: 10.1093/bioinformatics/btm134. URL https://doi.org/10.1093/bioinformatics/btm134.

Jingu Kim and Haesun Park. Sparse nonnegative matrix factorization for clustering. Technical report, Georgia Institute of Technology, 2008. URL https://faculty.cc.gatech.edu/~hpark/papers/GT-CSE-08-01.pdf.

Keigo Kimura, Yuzuru Tanaka, and Mineichi Kudo. A fast hierarchical alternating least squares algorithm for orthogonal nonnegative matrix factorization. In Dinh Phung and Hang Li (eds.), *Proceedings of the Sixth Asian Conference on Machine Learning*, volume 39 of *Proceedings of Machine Learning Research*, pp. 129–141, Nha Trang City, Vietnam, 26–28 Nov 2015. PMLR. URL https://proceedings.mlr.press/v39/kimura14.html.

Kamran Kowsari, Donald E. Brown, Mojtaba Heidarysafa, Kiana Jafari Meimandi, Matthew S. Gerber, and Laura E. Barnes. HDLTex: Hierarchical deep learning for text classification. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 364–371, 2017. doi: 10.1109/ICMLA.2017.0-134.

Kamran Kowsari, Donald E. Brown, Mojtaba Heidarysafa, Kiana Jafari Meimandi, Matthew S. Gerber, and Laura E. Barnes. Web of science dataset, March 2019. URL https://data.mendeley.com/datasets/9rw3vkcfy4/6.

Ken Lang. 20 newsgroups, 1997. URL http://qwone.com/~jason/20Newsgroups/.

Daniel Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In T. Leen, T. Dietterich, and V. Tresp (eds.), *Advances in Neural Information Processing Systems*, volume 13. MIT Press, 2000. URL https://proceedings.neurips.cc/paper_files/paper/2000/file/f9d1152547c0bde01830b7e8bd60024c-Paper.pdf.

Daniel D. Lee and H. Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, October 1999. doi: 10.1038/44565. URL https://doi.org/10.1038/44565.

Chih-Jen Lin. Projected gradient methods for nonnegative matrix factorization. *Neural Computation*, 19 (10):2756–2779, October 2007. doi: 10.1162/neco.2007.19.10.2756. URL https://doi.org/10.1162/neco.2007.19.10.2756.

Xihui Lin and Paul C. Boutros. Optimization and expansion of non-negative matrix factorization. *BMC Bioinformatics*, 21(1), January 2020. doi: 10.1186/s12859-019-3312-5. URL https://doi.org/10.1186/s12859-019-3312-5.

Haifeng Liu, Zhaohui Wu, Xuelong Li, Deng Cai, and Thomas S. Huang. Constrained nonnegative matrix factorization for image representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1299–1311, 2012. doi: 10.1109/TPAMI.2011.217.

Laura Muzzarelli, Susanne Weis, Simon B. Eickhoff, and Kaustubh R. Patil. Rank selection in non-negative matrix factorization: systematic comparison and a new MAD metric. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2019. doi: 10.1109/IJCNN.2019.8852146.

Art B. Owen and Patrick O. Perry. Bi-cross-validation of the SVD and the nonnegative matrix factorization. *The Annals of Applied Statistics*, 3(2), June 2009. doi: 10.1214/08-aoas227. URL https://doi.org/10.1214/08-aoas227.

V. Paul Pauca, Farial Shahnaz, Michael W. Berry, and Robert J. Plemmons. Text mining using non-negative matrix factorizations. In *Proceedings of the 2004 SIAM International Conference on Data Mining*. Society for Industrial and Applied Mathematics, April 2004. doi: 10.1137/1.9781611972740.45. URL https://doi.org/10.1137/1.9781611972740.45.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Rafael A Rosales, Rodrigo D Drummond, Renan Valieris, Emmanuel Dias-Neto, and Israel T da Silva. signeR: an empirical Bayesian approach to mutational signature discovery. *Bioinformatics*, 33(1):8–16, September 2016. ISSN 1367-4811. doi: 10.1093/bioinformatics/btw572. URL http://dx.doi.org/10.1093/bioinformatics/btw572.

Mikkel N. Schmidt, Ole Winther, and Lars Kai Hansen. Bayesian non-negative matrix factorization. In *Independent Component Analysis and Signal Separation*, pp. 540–547. Springer Berlin Heidelberg, 2009. doi: 10.1007/978-3-642-00599-2_68. URL https://doi.org/10.1007/978-3-642-00599-2_68.

Aristeidis Sotiras, Jon B. Toledo, Raquel E. Gur, Ruben C. Gur, Theodore D. Satterthwaite, and Christos Davatzikos. Patterns of coordinated cortical remodeling during adolescence and their associations with functional specialization and evolutionary expansion. *Proceedings of the National Academy of Sciences*, 114(13):3527–3532, March 2017. doi: 10.1073/pnas.1620928114. URL https://doi.org/10.1073/pnas.1620928114.

V. Y. F. Tan and C. Févotte. Automatic relevance determination in nonnegative matrix factorization. In *Proc. Workshop on Signal Processing with Adaptative Sparse Structured Representations (SPARS)*, St-Malo, France, Apr. 2009. URL 10.1109/TPAMI.2012.240.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert

Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

Yu-Xiong Wang and Yu-Jin Zhang. Nonnegative matrix factorization: A comprehensive review. *IEEE Transactions on Knowledge and Data Engineering*, 25(6):1336–1353, 2013. doi: 10.1109/TKDE.2012.51.

Svante Wold. Cross-validatory estimation of the number of components in factor and principal components models. *Technometrics*, 20(4):397–405, November 1978. doi: 10.1080/00401706.1978.10489693. URL `https://doi.org/10.1080/00401706.1978.10489693`.

Sayaka Yamauchi, Masanori Kawakita, and Jun'ichi Takeuchi. Botnet detection based on non-negative matrix factorization and the MDL principle. In *Neural Information Processing*, pp. 400–409. Springer Berlin Heidelberg, 2012. doi: 10.1007/978-3-642-34500-5_48. URL `https://doi.org/10.1007/978-3-642-34500-5_48`.

Xiaohui Yang, Wenming Wu, Xin Xin, Limin Su, and Liugen Xue. Adaptive factorization rank selection-based NMF and its application in tumor recognition. *International Journal of Machine Learning and Cybernetics*, 12(9):2673–2691, May 2021. ISSN 1868-808X. doi: 10.1007/s13042-021-01353-1. URL `http://dx.doi.org/10.1007/s13042-021-01353-1`.

Marinka Zitnik and Blaz Zupan. Nimfa: A Python library for nonnegative matrix factorization. *Journal of Machine Learning Research*, 13:849–853, 2012.