NSVD: Normalized Singular Value Deviation Reveals Number of Latent Factors in Tensor Decomposition

Yorgos Tsitsikas * Evangelos E. Papalexakis *

Abstract

Tensor decomposition has been shown, time and time again, to be an effective tool in multi-aspect data mining, especially in exploratory applications where the interest is in discovering hidden interpretable structure from the data. In such exploratory applications, the number of such hidden structures is of utmost importance, since incorrect selection may imply the discovery of noisy artifacts that do not really represent a meaningful pattern. Albeit extremely important, selection of this number of latent factors, also known as low-rank, is very hard, and in most cases, practitioners and researchers resort to ad-hoc trial-and-error, or assume that somehow this number is known or is given via domain expertise.

There has been a considerable amount of prior work that proposes heuristics for selecting this low rank. However, as we argue in this paper, the state-of-the-art in those heuristic methods is rather unstable and does not always reveal the correct answer.

In this paper, we propose the Normalized Singular Value Deviation (*NSVD*), a novel method for selecting the number of latent factors in Tensor Decomposition, that is based on principled theoretical foundations. We extensively evaluate the effectiveness of *NSVD* in synthetic and real data and demonstrate that it yields a more robust, stable, and reliable estimation than state-of-the-art.

1 Introduction

Data analysis and pattern extraction have always been an important tool in science and everyday life, since they provide a fundamental way of understanding, organizing and solving problems that may arise. In fact, these problems are often characterized by a multidimensional profile which, in turn, can produce an explosion in the complexity of the problem, and, thus, create a temptation to resort towards simplified and biased solutions that are easier to figure out.

It is becoming increasingly apparent, however, that finding ways to tackle these issues in their original multi-aspect form, can provide answers of superior quality and unmatched insight [29, 28]. Therefore, it makes sense to search for methods that can encapsulate this characteristic in a way simple enough that will also enable us to make useful discoveries about the task at hand. Tensors, which have traditionally been a mathematical tool, offer exactly this capability, since they possess the simplicity of a multidimensional array, and hence are able to capture the multiple facets of the problem at hand. Albert Einstein's general relativity which was formulated using tensors, is a particularly interesting example that demonstrates their power. At the same time, a broad range of interesting results in tensors [20, 22] provides an elegant mathematical framework which allows us to derive important and insightful conclusions about the nature of the tensor data and their structure.

PARAFAC decomposition [9, 17] and Kruskal's uniqueness conditions [22], have laid out an important part of the foundation for using multidimensional arrays as a tool for multi-aspect data mining. PARAFAC decompositions express the data as the sum of other elementary rank one tensors, and when performed properly they can reveal a lot about the underlying structure of the data. However, finding the right number of components for a PARAFAC decomposition is not an easy task. To make this more clear, consider tensor rank, which is another important concept [6, 5, 12], and is closely tied to finding a proper number of components for a PARAFAC decomposition. Tensor rank calculation has been proven to be NP-Hard over $\mathbb{Q}$ [18, 12] and over any extension of it including $\mathbb{R}$ and $\mathbb{C}$, and NP-Complete over finite fields [19]. Therefore, it becomes understood that despite the solid mathematical foundations that have been set so far, we are still a long way from finding a broadly efficient solution that sufficiently tackles this problem.

For this reason, people have resorted to heuristic ways for discovering low-rank structure in data [8, 27, 10], and even though some of them might enjoy success in specific domains, they are usually unable to generalize robustly to a broader spectrum of applications. In this work, we are exploring some of these pathways, and we are suggesting a different way of looking at this problem. Our contributions are:

- **A novel low-rank structure detection method**: We propose a new technique for finding the appropriate number of PARAFAC components, by performing a simple but powerful transformation to the decomposition, which enables us the use of important linear algebra tools.
- **Extensive theoretical analysis**: We provide and

*Department of Computer Science and Engineering, University of California, Riverside. gtsit001@ucr.edu, epapalex@cs.ucr.edu

prove various advantageous theoretical properties of our technique, while pointing out the respective shortcomings in the theoretical formulation of an existing widely used method.

- **Thorough experimental evaluation**: Multiple experiments for various settings are carried out on real and artificial data, in order to study and evaluate the behavior of our method in comparison to other baselines.

Reproducibility: In order to promote reproducibility of our results, we make our code for *NSVD* and the synthetic tensor generator used in the paper publicly available[1].

2 Problem Formulation

2.1 Notation & Definitions Even though tensors are often defined as elements of specific tensor product spaces that correspond to multilinear maps, it is common in the field of data mining to define an N-mode tensor as an element of the tensor product of N arbitrary vector spaces. Similarly to matrices, if we choose a basis for each vector space, then the tensor can be represented as a multidimensional array of numbers. Without loss of generality, in this work we will study only 3-mode tensors $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I \times J \times K}$, where I, J and K are the dimensions of the respective vector spaces.

Additionally, we have the following definitions:

Mode-n Fiber: A column vector produced by fixing the indices in all of the dimensions of the tensor except from the n-th dimension. For example, the Mode-1 fibers of a $2\times2\times2$ tensor $\boldsymbol{\mathcal{X}}$ can be identified as $\boldsymbol{\mathcal{X}}(:, j, k)$ for all $j, k = 1, 2$.

n-Mode Product: It is denoted as $\boldsymbol{\mathcal{X}} \times_n \mathbf{M}$ where $\mathbf{M}$ is an $L \times I_n$ matrix and I_n is the n-th dimension of $\boldsymbol{\mathcal{X}}$. It modifies $\boldsymbol{\mathcal{X}}$ by transforming its mode-n fibers as $\mathbf{M}\boldsymbol{\mathcal{X}}(\cdots, i_{n-1}, :, \cdots)$.

Frobenius Norm: $||\boldsymbol{\mathcal{X}}|| = \sqrt{\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K} \boldsymbol{\mathcal{X}}(i,j,k)^2}$.

Vectorization: It is denoted as $\text{vec}\,\boldsymbol{\mathcal{X}}$ and is a column vector constructed by concatenating all mode-1 fibers $\boldsymbol{\mathcal{X}}(:, j, k)$, with the smaller j and k having higher priority in the concatenation, and similarly the second dimension has higher priority than the third dimension.

2.2 PARAFAC Decomposition As already discussed, tensor decompositions play an important role in discovering structure in multi-aspect data. Even though a plethora of decompositions have been proposed, in this work we will only concern ourselves with the PARAFAC decomposition since it has a very close connection to the rank of a tensor $\boldsymbol{\mathcal{X}}$. To see this, we first express $\boldsymbol{\mathcal{X}}$ in terms of its PARAFAC decomposition as follows

$$\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{I}} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C}$$

where $\mathbf{A} \in \mathbb{R}^{I \times R}$, $\mathbf{B} \in \mathbb{R}^{J \times R}$ and $\mathbf{C} \in \mathbb{R}^{K \times R}$ are the PARAFAC factor matrices, R is the number of PARAFAC components, also called CP-rank, and $\boldsymbol{\mathcal{I}} \in \mathbb{R}^{R \times R \times R}$ for which it holds that $\boldsymbol{\mathcal{I}}(i, j, k) = 1$ if $i = j = k$ and $\boldsymbol{\mathcal{I}}(i, j, k) = 0$ otherwise. Note that this expression can be reformulated as

$$\boldsymbol{\mathcal{X}} = \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r \tag{2.1}$$

where $\mathbf{a}_r$,$\mathbf{b}_r$ and $\mathbf{c}_r$ are the r-th columns of the factor matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$, respectively. Since this is the sum of R rank one tensors, it becomes evident that if we manage to find the minimum R for which (2.1) holds, then we have essentially found the rank of the tensor. Additionally, for fixed values of R, the PARAFAC decomposition is usually approximated by using alternating least squares algorithms [9, 17] which minimize the Frobenius norm of the error.

An important obstacle in finding the optimal R though, is the fact that even for a CP-rank less than the actual tensor rank, there is the possibility that a decomposition exists that produces an arbitrarily small error. This can occur due to the fact that a rank-R best approximation of a tensor might not even exist [11]. Thus, a common and intuitive idea is to calculate approximate PARAFAC decompositions for a range of CP-ranks, and then evaluate them with an effective diagnostic tool that will hopefully uncover the decomposition with the proper number of components.

Symbol	Definition
x	Scalar
$\mathbf{X}$	Matrix
$\boldsymbol{\mathcal{X}}$	Tensor
$\otimes$	Kronecker Product
$\odot$	Column-wise Khatri-Rao Product
$\circ$	Outer Product

Table 1: Table of Symbols

2.3 AutoTen & the Core Consistency Diagnostic As already discussed, rank estimation and low-rank trilinear structure discovery are very difficult problems, and there are currently no general purpose tools that can efficiently accomplish these tasks. It is worth elaborating, however, on some of the most effective tools, two of which are AutoTen [24] and the Core Consistency Diagnostic (CORCONDIA) [7, 8] .

[1] https://github.com/gtsitsik/NSVD

AutoTen is a tool which aims to provide unsupervised detection of multi-linear low-rank structure in tensors, and is currently considered state-of-the-art among its competitors who also attempt to automate the task. In closer inspection, however, one can observe that AutoTen's success can in large part be attributed to the power of CORCONDIA, which is one of its main building blocks, and, therefore, for all intents and purposes we can directly study and analyze the behavior of CORCONDIA instead of AutoTen's.

In order to make this claim concrete, we first remind that, given the PARAFAC factor matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ of $\boldsymbol{\mathcal{X}}$, CORCONDIA is defined as

$$\left(1 - \frac{||\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{G}}^*||^2}{||\boldsymbol{\mathcal{I}}||^2}\right) \cdot 100$$

where

$$\boldsymbol{\mathcal{G}}^* = \arg\min_{\boldsymbol{\mathcal{G}}} ||\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{G}} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C}|| \tag{2.2}$$

or equivalently

$$\operatorname{vec} \boldsymbol{\mathcal{G}}^* = \arg\min_{\operatorname{vec} \boldsymbol{\mathcal{G}}} || \operatorname{vec} \boldsymbol{\mathcal{X}} - (\mathbf{C} \otimes \mathbf{B} \otimes \mathbf{A}) \operatorname{vec} \boldsymbol{\mathcal{G}}||$$

which shows that the calculation of $\boldsymbol{\mathcal{G}}^*$ corresponds to a linear least squares problem.

Essentially, CORCONDIA evaluates how well a PARAFAC decomposition captures low-rank trilinear structure in the data by comparing it to how well the data are modeled when interactions among the components of the decomposition are allowed. These interactions are described by the off-diagonal elements of $\boldsymbol{\mathcal{G}}$, and therefore, we see that when a lot of them have non-zero values, then CORCONDIA will tend to have large values. Notice that for $\boldsymbol{\mathcal{G}} = \boldsymbol{\mathcal{I}}$ in (2.2) not only there exist no interactions between the components, but this is exactly how PARAFAC is formulated in the least squares sense. Since, CORCONDIA attains its goal by comparing $\boldsymbol{\mathcal{G}}^*$ to $\boldsymbol{\mathcal{I}}$, when their difference is small, CORCONDIA will have a value close to 100, which implies that the corresponding decomposition is appropriate and captures mostly trilinear structure in the data. On the other hand, if the difference is not trivial, CORCONDIA can get close to zero or even negative, which is a strong indication that the given decomposition is not capturing properly the trilinear variation in the data, and hence it should probably be discarded.

Since there could be multiple models which give a high value of CORCONDIA, the way that we choose between them is by selecting the one that also has the largest number of components. The reason that we choose this model is because it will most probably also provide the best fit in terms of the squared norm of the error. In this manner, we can explore a trade-off between the quality of the model based on CORCONDIA and its fit.

Based on this reasoning, AutoTen attempts to produce accurate estimates of the real number of components by examining CORCONDIA at CP-ranks where it starts dropping from 100 to zero. AutoTen calculates CORCONDIA in the least squares sense, but is generally able to generate finer estimates by further considering CORCONDIA based on the KL-divergence. In this work, we will focus on the least squares based CORCONDIA, and below we elaborate on some of its drawbacks, which can have a serious impact on the performance of AutoTen.

One of the weaknesses of CORCONDIA stems from the fact that even with a unique PAFAFAC decomposition, the factor matrices still suffer from scaling and permutation indeterminacies. To make this clear, consider the R-component PARAFAC of $\boldsymbol{\mathcal{X}}$ and the resulting $\boldsymbol{\mathcal{G}}^*$ from (2.2). Note also, that an equally valid set of factor matrices would be $\mathbf{APS_A}$,$\mathbf{BPS_B}$ and $\mathbf{CPS_C}$ where $\mathbf{P}$ is a permutation matrix and $\mathbf{S_A}$, $\mathbf{S_B}$ and $\mathbf{S_C}$ are diagonal scaling matrices for which it holds that $\mathbf{S_A S_B S_C} = \mathbf{I}$. In this case, (2.2) would give

$$\begin{aligned}\tilde{\boldsymbol{\mathcal{G}}}^* &= \arg\min_{\boldsymbol{\mathcal{G}}} ||\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{G}} \times_1 \mathbf{APS_A} \times_2 \mathbf{BPS_B} \times_3 \mathbf{CPS_C}|| \\ &= \arg\min_{\boldsymbol{\mathcal{G}}} ||\boldsymbol{\mathcal{X}} - f(\boldsymbol{\mathcal{G}}) \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C}||\end{aligned}$$

where $f(\boldsymbol{\mathcal{G}}) = \boldsymbol{\mathcal{G}} \times_1 \mathbf{PS_A} \times_2 \mathbf{PS_B} \times_3 \mathbf{PS_C}$. This is minimized only if

$$\begin{aligned}&f(\tilde{\boldsymbol{\mathcal{G}}}^*) = \boldsymbol{\mathcal{G}}^* \iff \\ &\quad \tilde{\boldsymbol{\mathcal{G}}}^* = \boldsymbol{\mathcal{G}}^* \times_1 \mathbf{S_A^{-1}P^{-1}} \times_2 \mathbf{S_B^{-1}P^{-1}} \times_3 \mathbf{S_C^{-1}P^{-1}}\end{aligned}$$

Therefore, there is an infinite number of valid $\boldsymbol{\mathcal{G}}^*$, which in turn implies that CORCONDIA will not have a unique value.

3 Proposed Method: *NSVD*

Since CORCONDIA can suffer from such indeterminacies which can create instabilities in the quality of its estimates, we propose a new method for discovering trilinear structure in tensor data and mitigating the issue above. Our method also possesses additional useful properties that provide further support for its power and robustness, as discussed in the following subsections.

3.1 Theoretical Formulation The first and probably the most important observation that we have to make is that the Khatri-Rao product $\mathbf{C} \odot \mathbf{B} \odot \mathbf{A}$ is a matrix which has the vectorized factors of the decomposition as its columns. This transformation is critical

for our analysis, since it allows us to manipulate and make inferences about the quality and the properties of a PARAFAC decomposition by employing a plethora of very well established tools in linear algebra.

Specifically, notice that we can use the singular values of $\mathbf{C} \odot \mathbf{B} \odot \mathbf{A}$ as a proxy for the behavior of any PARAFAC decomposition. This is a particularly interesting choice as explained in the following lemmas.

LEMMA 3.1. *The singular values of* $\mathbf{C} \odot \mathbf{B} \odot \mathbf{A}$ *are unaffected by the scaling and permutation indeterminacies of PARAFAC.*

Proof. Scaling issues are directly solved by the Khatri-Rao product as

$$\begin{aligned} \mathbf{CPS_C} \odot \mathbf{BPS_B} \odot \mathbf{APS_A} &= (\mathbf{CP} \otimes \mathbf{BP} \otimes \mathbf{AP})\mathbf{Z} \\ &= (\mathbf{CP} \odot \mathbf{BP} \odot \mathbf{AP}) \end{aligned}$$

where $\mathbf{Z} = \mathbf{S_C} \odot \mathbf{S_B} \odot \mathbf{S_A}$ is exactly the matrix that can transform a Kronecker product to a Khatri-Rao product when multiplied from the right [23].

Regarding the permutation indeterminacy, first we observe that performing identical column permutations on $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ and then computing their Khatri-Rao product produces the same result as when we first calculate their Khatri-Rao product and then we perform the same column permutations on this matrix. This is the same as saying that $\mathbf{CP} \odot \mathbf{BP} \odot \mathbf{AP} = (\mathbf{C} \odot \mathbf{B} \odot \mathbf{A})\mathbf{P}$, which in turn means that, if the Singular Value Decomposition (SVD) of $\mathbf{C} \odot \mathbf{B} \odot \mathbf{A}$ is $\mathbf{U\Sigma V}$, then the SVD of $\mathbf{CPS_C} \odot \mathbf{BPS_B} \odot \mathbf{APS_A}$ can be calculated as $\mathbf{U\Sigma V'}$ with $\mathbf{V'} = \mathbf{VP}$, and thus consists of the same singular values. ∎

Notice that these singular values can capture the essence of the decomposition in just a few parameters. This can prove a crucial factor in evaluating the decompositions, since using other quantities like the norm of the error, $||\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{I}} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C}||$, involves a huge number of parameters which can lead to greater inaccuracies. A specific example of this idea is presented experimentally in the next section, where it becomes obvious that in various cases a method based on the norm of the error can indeed be inadequate, as opposed to a method based on the singular values of $\mathbf{C} \odot \mathbf{B} \odot \mathbf{A}$. In particular, for any R-component PARAFAC decomposition of $\boldsymbol{\mathcal{X}}$, we only need to study R singular values, as opposed to an aggregation of the $I \cdot J \cdot K$ errors. Note that, even though these singular values are also obtained based on all the elements of $\boldsymbol{\mathcal{X}}$, they can provide a more stable means of evaluation, since they also have nice properties as discussed in the lemmas below.

LEMMA 3.2. *The Kronecker product* $\bigotimes_{i=1}^{n} \mathbf{A}_i$ *of any set of* n *matrices* $\mathbf{A}_i$ *is an orthogonal matrix if, and only if, all* $\mathbf{A}_i$ *are multiples of orthogonal matrices, such that* $\mathbf{A}_i^T \mathbf{A}_i = c_i \mathbf{I}$ *and* $\prod_{i=1}^{n} c_i = 1$.

(See appendix for proof)

LEMMA 3.3. *The singular values of* $\mathbf{C} \odot \mathbf{B} \odot \mathbf{A}$ *are identical, up to scaling, to the singular values produced by all tensors of the form* $\boldsymbol{\mathcal{X}} \times_1 \mathbf{R}_1 \times_2 \mathbf{R}_2 \times_3 \mathbf{R}_3$*, for* $\mathbf{R}_1$*,* $\mathbf{R}_2$ *and* $\mathbf{R}_3$ *multiples of arbitrary orthogonal matrices.*

Proof. Considering again the SVD $\mathbf{U\Sigma V}$ of $\mathbf{C} \odot \mathbf{B} \odot \mathbf{A}$, we observe that every other possible matrix of the same size that has the same singular values will have the form $\mathbf{U'U\Sigma VV'}$ where $\mathbf{U'}$ and $\mathbf{V'}$ are orthogonal matrices. However, not every orthogonal $\mathbf{U'}$ and $\mathbf{V'}$ results in a matrix $\mathbf{U'}(\mathbf{C} \odot \mathbf{B} \odot \mathbf{A})\mathbf{V'}$ that has a Khatri-Rao structure $\mathbf{C'} \odot \mathbf{B'} \odot \mathbf{A'}$. Therefore, if we set $\mathbf{V'} = \mathbf{I}$ and select a $\mathbf{U'}$ that is in the Kronecker form $\mathbf{R_3} \otimes \mathbf{R_2} \otimes \mathbf{R_1}$, then, following Lemma 3.2, $\mathbf{R_1}$, $\mathbf{R_2}$ and $\mathbf{R_3}$ have to be multiples of orthogonal matrices and the product of the norms of their corresponding columns has to be equal to one. However, observe that any multiple of $\mathbf{U'}$ is also a valid choice if we let the singular values absorb the scaling, i.e. $(c\mathbf{U'})\mathbf{U\Sigma VV'} = \mathbf{U'U}(c\mathbf{\Sigma})\mathbf{VV'}$. ∎

In other words, these singular values can be viewed as a more direct identifier of the structure of $\boldsymbol{\mathcal{X}}$ without being as prone to changes in the actual form of the tensor. Particularly, this implies that although methods like CORCONDIA or the norm of the error might report bad evaluations for orthogonal rotations of the decomposition, the singular values of $\mathbf{C} \odot \mathbf{B} \odot \mathbf{A}$ will signify that the decomposition is appropriate, since they will be identical, up to scaling, to the ones corresponding to the PARAFAC of the original unrotated $\boldsymbol{\mathcal{X}}$.

The importance of this becomes more evident if we consider the fact that the set of tensors consisting of $\boldsymbol{\mathcal{X}}$ and all of its orthogonal rotations contain in some sense the same underlying structure, even though directly looking at their elements on a high level might make them appear as completely different tensors. In fact, all these tensor representations correspond to the exact same abstract tensor in different bases, which implies that a sound rank estimation method should ideally produce similar rank estimates for all of them.

Additionally, it is not unreasonable to expect that multiple approximate solutions of an optimization algorithm for PARAFAC, produced by using different randomly generated initial points for example, will generally be similar to each other when the specified number of components captures the underlying structure properly. On the other hand, we can expect the solutions to

have greater deviations from each other when the given number of components cannot describe the structure of the data properly. Particularly, notice that for a fixed number of components, there could be a huge number of ways that extraneous components can be combined to produce the decomposition, when we decompose with more components than necessary.

Based on this observation, we propose the use of the variances of the singular values of $\mathbf{C} \odot \mathbf{B} \odot \mathbf{A}$ as a means to detecting and properly quantifying such deviations. Specifically, in order to provide fair comparisons, we divide these variances by the respective expected values, and, finally, we aggregate all these quantities by calculating the sum of their logarithms. Using the logarithm has a two-fold advantage since it provides numerical stability and leads to more interpretable plots. Numerical inaccuracies can occur if we only consider the product of the variances of the singular values since individual variances sometimes take values very close to zero. Also, considering that different CP-ranks can lead to a change of many orders of magnitude to this product, we can see that the logarithm provides a more suitable option.

More formally, considering a probability distribution for the random initializations of the algorithm that calculates the R-component PARAFAC, and given the variance $\sigma^2_{R,i}$ and the expected value $\mu_{R,i}$ of the i-th singular value of $\mathbf{C} \odot \mathbf{B} \odot \mathbf{A}$, we compute $\sum_{i=1}^{R} \log(\sigma^2_{R,i}/\mu_{R,i})$, which we will call *Normalized Singular Value Deviation* (*NSVD*). In this fashion, we are able to discover possible trilinear structure in the data and the number of PARAFAC components that best models it, with distinct dips in *NSVD* being an indication of more appropriate decompositions.

In summary, *NSVD* consists of the following steps:

Step 1 Compute multiple R-component PARAFAC decompositions by using multiple random initializations for the algorithm that calculates them.

Step 2 Form the Khatri-Rao product $\mathbf{C} \odot \mathbf{B} \odot \mathbf{A}$ for each decomposition.

Step 3 Compute the singular values of each $\mathbf{C} \odot \mathbf{B} \odot \mathbf{A}$ and estimate the variance $\sigma^2_{R,i}$ and the expected value $\mu_{R,i}$ of the i-th singular value, for all $i = 1 \dots R$.

Step 4 Compute $f(R) = \sum_{i=1}^{R} \log(\sigma^2_{R,i}/\mu_{R,i})$.

Step 5 Repeat **Steps 1-4** for multiple values of R.

Step 6 Estimate that k number of PARAFAC components appropriately model the underlying structure of the data if $f(k)$ is the lowest part of a distinct dip.

Notice that we did not specify a method for calculating the optimal expected value and variance estimates, since their optimality depends on their desired properties. However, in order to keep things simple and effective, in all our experiments we are going to make use of the sample mean and the unbiased sample variance.

4 Experimental Evaluation

In our experiments, at each number of components, R, we approximate *NSVD* by first calculating a number, K, of PARAFAC estimates using random initializations for the optimization algorithm. These PARAFACs are then used for the estimation of the necessary unbiased sample variances and sample means of the singular values of $\mathbf{C} \odot \mathbf{B} \odot \mathbf{A}$. Lastly, we aggregate all these estimates as explained in subsection 3.1 to get our *NSVD* estimate for R components.

For more accurate results, we calculate *NSVD* on the first k samples, and we repeat this for all k ranging from 2 to K. The final *NSVD* estimate is calculated based on these $K-1$ intermediate estimates. For further implementation details, we refer the interested reader to the appendix.

Finally, Tensor Toolbox [4, 3] was used for calculating all PARAFAC decompositions, and N-way Toolbox [1] was used only in cases where there exist missing data. Also, in all plots the horizontal axis always represents the number of PARAFAC components, while the errorbars signify the 25th and the 75th percentile of the $K-1$ intermediate *NSVD* estimates.

4.1 Artificial Data A first step in evaluating our method is to assess its performance on artificially generated data whose structure can be defined explicitly. In the following experiments, we create artificial tensors by generating three matrices with R columns each, where their elements are drawn from the standard normal distribution. Note that if we consider these matrices as the PARAFAC factor matrices of our tensor, then the rank of the tensor will be at most R, since it will be the sum of R rank-1 factors. Additionally, these factor matrices will be full column rank with very high probability, which in turn implies that the R factors will also be linearly independent. Therefore, the rank of the tensor will be exactly R.

In order to achieve a better approximation of real tensor data scenarios, however, our rank-R tensor, $\boldsymbol{\mathcal{X}}$, is distorted by an additive noise tensor $\boldsymbol{\mathcal{N}}$, which is a rank-100 tensor generated in the same way as $\boldsymbol{\mathcal{X}}$, but has its norm scaled to be p times less than the norm of $\boldsymbol{\mathcal{X}}$. In other words, the final tensor will be $\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{X}} + \boldsymbol{\mathcal{N}} \frac{||\boldsymbol{\mathcal{X}}||}{p||\boldsymbol{\mathcal{N}}||}$. In this manner, we generated the $20 \times 20 \times 20$ SynthSmall tensor having 5 components and $p = 2$, which is studied in subsection 4.3. Additional details about generating artificial noise can be found at the appendix.

Also, keep in mind that although there is not a perfect way of generating artificial data, our method is attempting to provide good approximations of real tensor data, without losing the flexibility and robustness provided by its mathematical foundation. The data

Name	Dimensions	Components
Chem1	$351 \times 19 \times 83$	6 (Chem. Verified)
Chem2	$351 \times 18 \times 83$	5 (Chem. Verified)
RealMining	$94 \times 94 \times 4$	6 (see [25])
Enron	$184 \times 184 \times 44$	4 & 7 (see [2, 26])
SynthSmall	$20 \times 20 \times 20$	5 (Synthetic)

Table 2: Tensors analyzed in the experimental section.

generator and the *NSVD* code are available at `https://github.com/gtsitsik/NSVD`.

4.2 Real Data Even though *NSVD* behaves nicely on artificial tensor data, indicating a distinctive dip exactly where the predefined number of components is, it is important to also study its performance and behavior on real tensor data in order to assess its practicality in real-world scenarios. For this reason, we analyze a range of real data sets as shown in Table 2.

4.2.1 Chemical Data First, we analyze the chemical datasets Chem1 and Chem2. These datasets are a small time interval taken from a large tensor dataset where 44 wine samples were measured on a gas chromatographic system with mass spectrometry detection. Each interval represents a subset on the time axis where a few specific chemical compounds appear. The aim of modeling the data is to separate the overlapping signals from the compounds. One additional chemical dataset of this type is studied at the appendix.

Note that data of this nature have generally been observed to have a strong trilinear structure. Therefore, when we are called to discover the individual chemical substances in the chemical samples, performing decompositions like PARAFAC and identifying the proper number of components become tasks of central importance. Moreover, many times it is possible to chemically identify the number of components in the samples. Therefore, these useful properties render chemical data a very attractive tool for the evaluation of methods like ours which aim to provide estimations for the rank of a tensor or detect low-rank multilinear structure in data. Unfortunately, however, the chemical data studied in this work have not become publicly available, which limits our ability to meaningfully study and explain their various components.

4.2.2 Social Network Data Next, we study the Reality Mining dataset [14], henceforth referred to as RealMining, which includes data gathered by the MIT Media Laboratory in the context of an experiment assessing the behavior of social communities. These communities consist of 100 subjects who participated in the study, 94 of which are included in our dataset. The experiment was conducted by special software that was installed in the phones of the participants, and among others it recorded four distinct communication aspects for each subject, i.e. calls, Bluetooth devices nearby, text messages and friendship with the rest of the subjects, which form a 3-mode tensor with four slices. Various studies have been conducted on this dataset [16, 15], and particularly in [25] the authors discover and elaborate on 6 dominant communities.

We are also evaluating our method on the famous Enron dataset which was made public after the corporation's scandal was uncovered. The dataset contains a large number of emails that were exchanged between around 150 employees of the company, and has attracted a lot of attention [21, 13], mainly due to the fact that it probably is the only publicly available large scale dataset of real-world emails. In our work, we study a smaller version of the dataset obtained as explained in [2], which consists of the communication between 184 email addresses in an interval of 44 weeks, and, therefore, can be represented as a 3-mode tensor. In the same study, the authors identify and elaborate on 4 distinct communities, while in [26] the authors, after running multiple iterations of their experiments, claim to have discovered a total of 7 communities with a standard deviation of 0.88.

4.3 Comparison to Baselines Detecting structure in data can be a daunting task, especially when one attempts to automate the process and make it as universally and domain agnostic as possible. Many heuristic methods have been proposed for this purpose [8, 27, 10], each with its own advantages and drawbacks, often making it necessary to resort to different structure finding techniques depending on the problem at hand.

It is crucial for the evaluation of *NSVD*, therefore, to put it into the test against other baselines, and in turn assess its capability in revealing the correct number of components for a range of artificial and real datasets. The comparison is carried out on the data described in subsection 4.1 and subsection 4.2 using 100 iterations for the calculation of all *NSVD* estimates, except for the RealMining dataset where 74 iterations were used due to limitations in computational resources. The following baselines are considered in the comparison:

- **CORCONDIA**: (see subsection 2.3)
- **Mean Squared Error** (MSE): It is estimated that R components best describe the trilinear variation in the data, if the R-component PARAFAC gives a low enough MSE compared to PARAFACs with different number of components. This usually manifests as a distinct dip at R components.

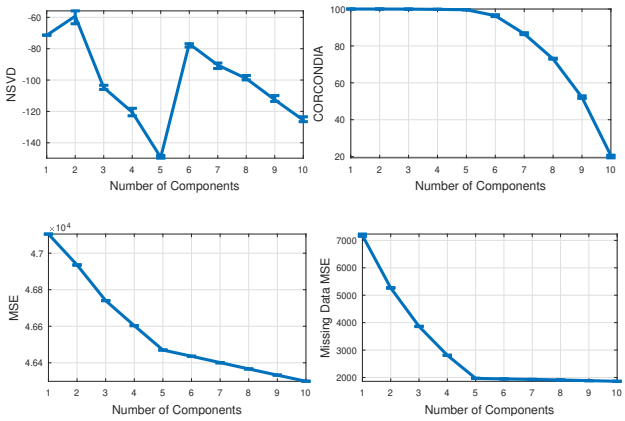


Figure 1: Baselines comparison on the SynthSmall dataset

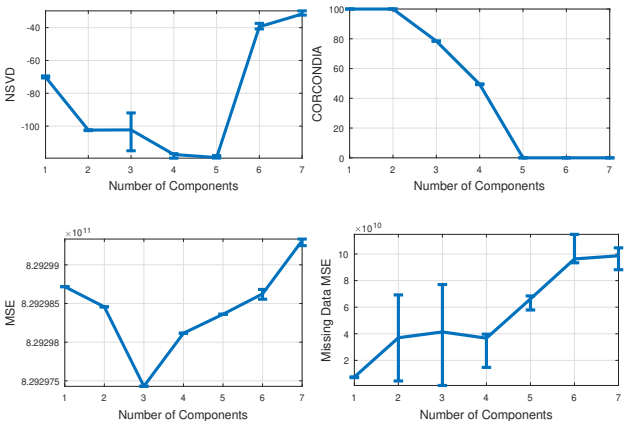


Figure 3: Baselines comparison on the Chem2 dataset

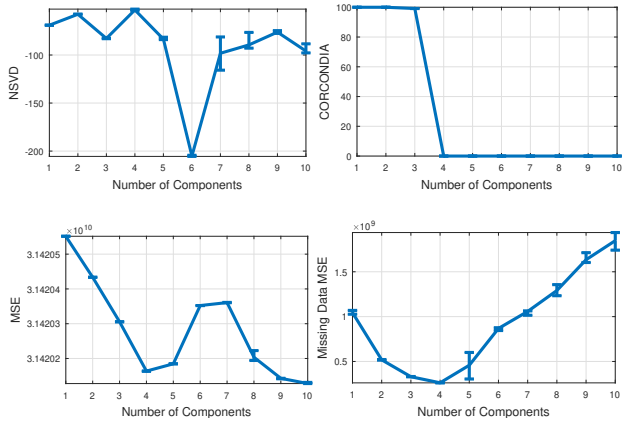


Figure 2: Baselines comparison on the Chem1 dataset

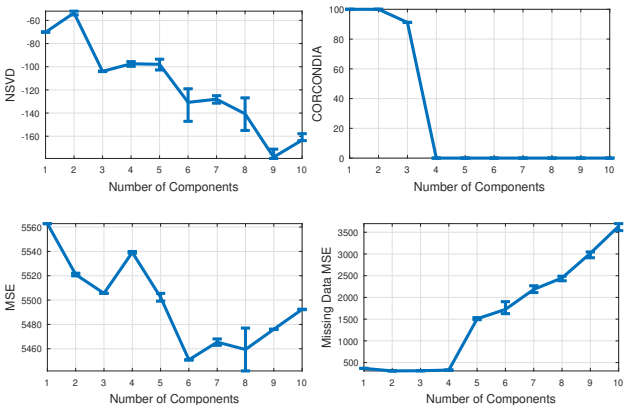


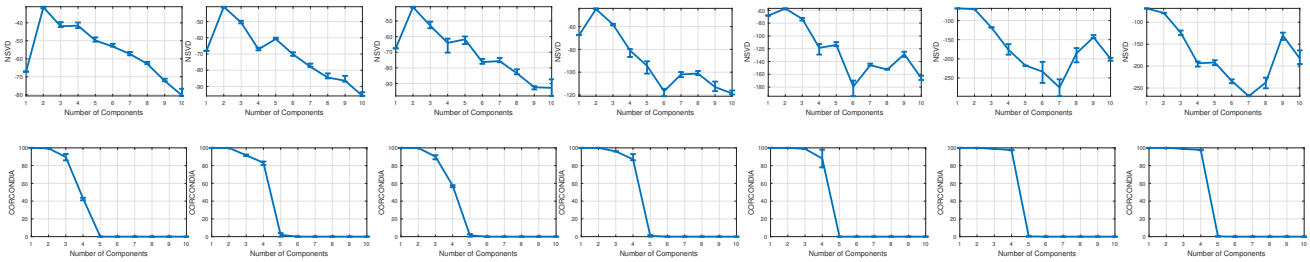
Figure 4: Baselines comparison on the RealMining dataset

- **Mean Squared Error on missing data**: Similar to the usual MSE, with the only difference being that the PARAFAC decompositions are estimated using only the known data of the tensor, and the MSE is calculated on the missing data. Note that if R components produce low MSE on the missing data, it can be interpreted as that the corresponding PARAFAC model can better predict them, and, hence, we can estimate that an R-component underlying structure is appropriate. In our work, we always assume 20% of missing data selected randomly in each iteration.

4.3.1 Synthetic Data For our synthetic dataset SynthSmall we observe in Figure 1 that *NSVD* presents a quite distinct dip at 5 components which is the correct answer. On the other hand, even though CORCONDIA seems to approximate a region around 5 components, it struggles to give a definitive answer and leaves open the possibility of up to 7 or even 8 components. Interestingly, even MSE seems to be working better than CORCONDIA in this case, showing a subtle indication at 5 components, which gets amplified when using MSE on the missing data.

4.3.2 Chemical Data In Figure 2, we can see the nice indication that *NSVD* provides for the 6 components of Chem1, as opposed to CORCONDIA which suggests only 3 components. MSE also fails to identify the correct answer by vaguely indicating only 4 or 5 components, and similarly for the MSE on missing data which suggests only 4 components.

The situation is different for Chem2 where all baselines fail to discover the 5 components in the data, as shown in Figure 3. CORCONDIA provides a vague estimate of around 3 components, while MSE shows a distinct dip also at 3 components. MSE on missing data also fails to give a definitive answer, even though it slightly hints at 4 components. On the other hand, *NSVD* provides a much clearer indication at 4 and 5 components, and although it is not as sharp as in the previous examples, it outperforms all the baselines.



(a) tolerance: 1e-2 (b) tolerance: 1e-3 (c) tolerance: 7e-4 (d) tolerance: 1e-4 (e) tolerance: 1e-7 (f) tolerance: 1e-10 (g) tolerance: 1e-12

Figure 5: *NSVD* & CORCONDIA on the Enron dataset for tighter tolerances from left to right. Observe how *NSVD* is able to reveal multiple structures in the data by using different tolerances.

4.3.3 Social Network Data In Figure 4 we see that the behavior of *NSVD* on the RealMining dataset is more convoluted, providing multiple indications at 3, 6 and 9 components. CORCONDIA provides an indication only at 3 components, while MSE hints at 3 and 6 components. Note that even though it is hard to obtain ground truth for this type of data, *NSVD* is able to at least suggest potential structure at all the points where both CORCONDIA and MSE do as well. It is also interesting to observe that the 6 components that the authors identify in [25] make sense considering that not only both *NSVD* and MSE provide an indication at this number of components, but also MSE takes its lowest value at that point too. Finally, MSE on missing data fails to provide a clear estimate since it could indicate anything from 1 to 4 components.

4.4 Fine-Tuning *NSVD* At this point, we are going to demonstrate an interesting property of *NSVD* which makes it even more flexible and versatile, allowing it to provide sharper and more accurate estimates, and even uncover different levels of structure in the data. Additionally, this improvement can be achieved by only fine-tuning a simple parameter; the tolerance for the termination criterion of the optimization algorithm that calculates the PARAFAC decompositions.

This interesting property can be observed in Figure 5 where *NSVD* is tested on the Enron dataset for multiple tolerance levels. At first, for the low tolerance of $1e-2$, *NSVD* provides no robust estimate at all. However, at a tolerance of $1e-3$ it shows a distinct indication at 4 components which is in agreement with the findings in [2]. Next, for a tolerance of $7e-4$ we see that *NSVD* starts transitioning from an estimate of 4 to an estimate of 6 components, which later becomes the only pronounced estimate for a tolerance of $1e-4$ and $1e-7$. In the end, *NSVD* seems to stabilize at 7 components for the even tighter tolerances of $1e-10$ and $1e-12$. Notice that the last estimates of 6 and 7 components are in perfect agreement with the findings in [26] where 7 components were identified with a standard deviation of 0.88. On the other hand, CORCONDIA is able to only discover 4 components, and remains virtually unaffected by the changes in the tolerance levels, except for a tolerance of $7e-4$ where its estimate gets distorted. Finally, additional experiments on fine-tuning *NSVD* for the Chem1 dataset are provided in the appendix.

5 Conclusions

We proposed a new method called *NSVD* for discovering low-rank multilinear structure in multiaspect data, which is based on the variance of the singular values of the Khatri-Rao product formed by the PARAFAC factor matrices. We have also shown various advantageous theoretical properties of our method, and we argued against a crucial theoretical shortcoming of CORCONDIA. Next, we offered extensive experimental evaluation on both artificial and real data, which verified that our method can be superior as compared to other widely used structure finding heuristics. Finally, we showed an interesting property of our method that allows it to discover different levels of structure in the data.

Acknowledgements

We are grateful to Rasmus Bro for his valuable feedback and for providing us with the chemical data. Research was supported by the National Science Foundation CDS&E Grant no. OAC-1808591 and by the Department of the Navy, Naval Engineering Education Consortium under award no. N00174-17-1-0005. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funding parties.

References

[1] Claus A. Andersson and Rasmus Bro. The n-way toolbox for matlab. *Chemometrics and Intelligent*

Laboratory Systems, 52(1):1–4, 2000. Available at: `https://www.mathworks.com/matlabcentral/fileexchange/1088-the-n-way-toolbox` (last checked January 2020).

[2] Brett W. Bader, Richard A. Harshman, and Tamara G. Kolda. Pattern analysis of directed graphs using dedicom: an application to enron email. Technical report, Sandia National Laboratories, 2006.

[3] Brett W. Bader and Tamara G. Kolda. Algorithm 862: MATLAB tensor classes for fast algorithm prototyping. *ACM Transactions on Mathematical Software*, 32(4):635–653, December 2006.

[4] Brett W. Bader, Tamara G. Kolda, et al. Matlab tensor toolbox version 2.6, February 2015. Available at: `https://gitlab.com/tensors/tensor_toolbox` (last checked January 2020).

[5] Dario A. Bini. The role of tensor rank in the complexity analysis of bilinear forms. *Presentation at ICIAM07, Zürich, Switzerland*, 2007.

[6] Markus Bläser. A 5/2n 2-lower bound for the multiplicative complexity of n× n-matrix multiplication. In *Annual Symposium on Theoretical Aspects of Computer Science*, pages 99–109. Springer, 2001.

[7] Rasmus Bro. Multi-way analysis in the food industry-models, algorithms, and applications. In *MRI, EPG and EMA," Proc ICSLP 2000*. Citeseer, 1998.

[8] Rasmus Bro and Henk A.L. Kiers. A new efficient method for determining the number of components in parafac models. *Journal of Chemometrics: A Journal of the Chemometrics Society*, 17(5):274–286, 2003.

[9] J. Douglas Carroll and Jih-Jie Chang. Analysis of individual differences in multidimensional scaling via an n-way generalization of "eckart-young" decomposition. *Psychometrika*, 35(3):283–319, 1970.

[10] Joao Paulo C.L. da Costa, Martin Haardt, and Florian Romer. Robust methods based on the hosvd for estimating the model order in parafac models. In *2008 5th IEEE Sensor Array and Multichannel Signal Processing Workshop*, pages 510–514. IEEE, 2008.

[11] Vin De Silva and Lek-Heng Lim. Tensor rank and the ill-posedness of the best low-rank approximation problem. *SIAM Journal on Matrix Analysis and Applications*, 30(3):1084–1127, 2008.

[12] Harm Derksen. Matrix completion and tensor rank. *Linear and Multilinear Algebra*, 64(4):680–685, 2016.

[13] Jana Diesner, Terrill L. Frantz, and Kathleen M. Carley. Communication networks from the enron email corpus "it's always about the people. enron is no different". *Computational & Mathematical Organization Theory*, 11(3):201–228, Oct 2005.

[14] Nathan Eagle and Alex Sandy Pentland. Reality mining: sensing complex social systems. *Personal and ubiquitous computing*, 10(4):255–268, 2006.

[15] Nathan Eagle and Alex Sandy Pentland. Eigenbehaviors: Identifying structure in routine. *Behavioral Ecology and Sociobiology*, 63(7):1057–1066, 2009.

[16] Nathan Eagle, Alex Sandy Pentland, and David Lazer. Inferring friendship network structure by using mobile phone data. *Proceedings of the National Academy of Sciences*, 106(36):15274–15278, 2009.

[17] Richard A. Harshman. Foundations of the parafac procedure: Models and conditions for an "explanatory" multimodal factor analysis. 1970.

[18] Johan Håstad. Tensor rank is np-complete. *Journal of Algorithms*, 11(4):644–654, 1990.

[19] Christopher J Hillar and Lek-Heng Lim. Most tensor problems are np-hard. *Journal of the ACM (JACM)*, 60(6):45, 2013.

[20] Thomas D. Howell. Global properties of tensor rank. *Linear Algebra and its Applications*, 22:9–23, 1978.

[21] Bryan Klimt and Yiming Yang. The enron corpus: A new dataset for email classification research. In *European Conference on Machine Learning*, pages 217–226. Springer, 2004.

[22] Joseph B. Kruskal. Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. *Linear algebra and its applications*, 18(2):95–138, 1977.

[23] Hanoch Lev-Ari et al. Efficient solution of linear matrix equations with application to multistatic antenna array processing. *Communications in Information & Systems*, 5(1):123–130, 2005.

[24] Evangelos E. Papalexakis. Automatic unsupervised tensor mining with quality assessment. In *Proceedings of the 2016 SIAM International Conference on Data Mining*, pages 711–719. SIAM, 2016.

[25] Evangelos E. Papalexakis, Leman Akoglu, and Dino Ience. Do more views of a graph help? community detection and clustering in multi-graphs. In *Proceedings of the 16th International Conference on Information Fusion*, pages 899–905. IEEE, 2013.

[26] Ravdeep Pasricha, Ekta Gujral, and Evangelos E. Papalexakis. Identifying and alleviating concept drift in streaming tensor decomposition. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 327–343. Springer, 2018.

[27] Saeed Pouryazdian, Soosan Beheshti, and Sridhar Krishnan. Candecomp/parafac model order selection based on reconstruction error in the presence of kronecker structured colored noise. *Digital Signal Processing*, 48:12–26, 2016.

[28] Jian-Tao Sun, Hua-Jun Zeng, Huan Liu, Yuchang Lu, and Zheng Chen. Cubesvd: a novel approach to personalized web search. In *Proceedings of the 14th international conference on World Wide Web*, pages 382–390. ACM, 2005.

[29] M. Alex O. Vasilescu and Demetri Terzopoulos. Multilinear analysis of image ensembles: Tensorfaces. In *European Conference on Computer Vision*, pages 447–460. Springer, 2002.