Orthogonal Inductive Tensor Completion

a Bachelor's Thesis

by JUSTUS WILL

supervised by
Dr. Antoine Ledent,
Prof. Dr. Marius Kloft and
Prof. Dr. Jörn Saß

November 10, 2020

Department of Mathematics,
Department of Computer Science

TECHNISCHE UNIVERSITÄT
KAISERSLAUTERN

Abstract

Matrix and Tensor Completion are key techniques, e.g., in recommendation systems. We build on a recently developed matrix-completion method, BOMIC, which performs nuclear norm regularized matrix completion jointly with trainable user and item biases: Each predictor is a sum of a purely regression based model composed of user and item biases, and a low rank model free of any behavior that could be seen as the effect of user or item biases.

In this thesis, we extend this idea to 3-tensors. We propose the tensor completion method BOTIC where each predictor again consists of a sum of different bias terms. In addition to first degree bias terms for each of the dimensions and second degree terms consisting of tensors exhibiting (low rank) dependence only on two of the indices our model includes a low rank "pure" tensor free of effects which could be modelled by other terms.

We look at different ways to regularize this residual order 3 tensor, such that a low rank tensor capturing only the most obvious of the purely order 3 phenomena in the data can be obtained.

Since we believe many tensor completion problems involve mostly pairwise interactions, with purely three-way interactions playing a relatively minor role, our model should be better suited to model realistic low rank phenomena behind naturally occurring low rank tensors.

Finally, BOTIC is compared against both pure tensor-based and matrix-based baselines in experiments on synthetic data.

Contents

1 Introduction and Overview

1.1 Introduction

Tensor completion is the task of imputing values in tensors that are missing, unobserved, or corrupted. Many modern tasks and datasets are highly complex and often exhibit high dimensionality. Understanding the underlying structure of the potentially huge amounts of data is crucial in benefiting from the data and allows tackling important data-driven applications. Often, the collected data is incomplete, only partial observations have been made or some measurements have gone missing.

When dealing with this complex incomplete data, tensor completion as well as the closely related fields of tensor decomposition and tensor approximation are key techniques that enable many insights to be gained from collected data and aid prediction of the currently unknown while keeping the prediction highly interpretable.

A special case of tensor completion, matrix completion, has been studied extensively and is well understood. Advanced methods and frameworks like *Orthogonal Inductive Matrix Completion* OMIC [1] have been developed and analyzed. Matrix completion methods have already been used in many diverse applications.

Although many of the popular matrix completion methods can be indirectly used to complete tensors, e.g. by working with slices of tensors or unfolding tensors into big matrices, the inherent multidimensional structure present in the high dimensional data can get lost and additional redundancies can be included thus weakening interpretability and understanding. In addition to matrix-based tensor completion tensor-based tensor completion has attracted increasing research interest.

Tensor based methods can be used everywhere where data can be represented in multidimensional arrays, e.g. when the data depends on several categorical factors.

Applications include recommender systems, data mining, graph analysis,
computer vision, photo and video reconstruction, signal processing, psychometrics, chemometrics and neuroscience [2], [3].

Throughout this thesis we will take a closer look at tensor completion in the context of recommender systems where we are interested in learning how much a specific user likes a set of items, including items not yet seen by the user. We can use this knowledge to recommend new items and aid user guided searches by filtering and sorting possible matches with respect to their unique preferences. Specific applications include recommendations for movies, music, products or other content, i.e. in the context of social media.

All possible interactions between a set of users and a set of items can be represented by a matrix M where M_{ij} is a numerical value representing the affinity of user i to the item j, i.e. how user i would rate item j, e.g. on a scale from 1 to 5. Each row $M_{i\cdot}$ represents a user i and contains all ratings of that user i on all items. Each column $M_{\cdot j}$ on the other hand represents an item j and contains all its ratings from all users.

Not every entry of M is known since not every user has seen and rated every item there is, often each user only rated a handful of items leading to the case where only a (potentially very small) portion of the needed data is available. Our goal is to fill in values for the missing entries, since it would be beneficial to know how a user would rate an item they haven't already seen.

It is commonly believed that the preferences of a user can be largely attributed to a small set of influences. For example, the affinity of a user to an item strongly depends on their affinity to similar items and the ratings of users with similar taste on the specific item. This justifies the assumption that the data has a low rank which can be used to our advantage and allows predictions of unobserved data points.

In Tensor Completion we consider data that is influenced by more than two underlying factors. For example, one could consider a recommender system model that also describes how user preferences change over time. If a continuous time frame is divided into distinct time intervals, the preferences can be represented by a tensor $\mathbf{M}$.

This time M_{ijk} indicates how much the user i likes item j in the time interval k.

Again the data is only observed very sparesly and M will be a tensor of low rank.

1.2 Contribution

The goal of this thesis is to develop the comprehensive and interpretable tensor-based tensor completion framework OTIC closely build on the recently developed matrix-completion framework OMIC.

BOMIC, a special case of OMIC, models matrices as a sum of purely regression based terms including user and item biases and a low rank model free of any user or item biases. This enables great interpretability of the results.

In this thesis, we extend this idea to 3-tensors. We develop and analyze a new tensor completion method we name BOTIC.

While first order and second order terms are still an important part of our new model, we also include a higher order term capturing the purely order three phenomena in the data which can not be modelled by other terms. The first oder terms still consist of biases for each of the dimensions while second order terms consist of tensors exhibiting (low rank) dependence only on two of the indices.

We propose to use the overlap nuclear norm to dynamically select a tensor with a low Tucker rank. Thus, most of the representational capacity of our model will come from the matrix completion terms with an additional low rank tensor capturing the most important higher order interactions.

We will show that this addition will improve the performance of BOMIC, demonstrating that higher order interactions should not be neglected by tensor completion methods and enabling future tensor completion methods based on OTIC.

1.3 Notation

To better visualize the difference between scalars, vectors, matrices and tensors we consistently denote

scalars with small letters (e.g. $x \in \mathbb{R}$),

vectors with bold small letters (e.g. $\mathbf{x} \in \mathbb{R}^m$),

matrices with big letters (e.g. $X \in \mathbb{R}^{m \times n}$),

tensors with big bold letters (e.g. $\mathbf{X} \in \mathbb{R}^{m_1 \times \cdots \times m_d}$).

1.4 Outline

This thesis will be structured as follows:

Initially, in Chapter 2 the mathematical foundations needed for this thesis are laid, including a formal introduction to tensors and the task of Tensor Completion.
Chapter 3 summarizes related methods in Matrix and Tensor Completion. Most notably BOMIC and the OMIC framework will be described in detail.
In Chapter 4 we will be developing the OTIC framework and taking a thorough look at the proposed algorithm, BOTIC. Regularization is discussed and the convergence is proven.
Next, the capabilities of the algorithm are demonstrated on synthetic data and real world data. Matrix based and Tensor based baselines are evaluated and compared to our method. The experiments are described in Chapter 5 including a discussion of the results.
Finally, in Chapter 6, we will evaluate the performance of the proposed method and discuss its potential by giving an outlook on further work that could be done to elevate the capabilities of OTIC.

2 Background

In this chapter we will be introducing the mathematical concepts that are relevant for our tensor completion framework and our proposed algorithm BOTIC. We will start with an introduction to tensors, explore the singular value decomposition and take a look at tensor decompositions. Next we will formally introduce the task of Matrix and Tensor Completion and finally we will briefly define subgradients.

2.1 Tensors

An element $\mathbf{X} \in \mathbb{R}^{m_1 \times \cdots \times m_d}$ is called a real valued tensor of order d.
Tensors can thus be seen as a multidimensional generalization of vectors and matrices, since all tensors of order 2 are also matrices and all tensors of order 1 are also vectors.

For tensors $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m_1 \times \cdots \times m_d}$ an inner product can be defined as

$$\langle \mathbf{A}, \mathbf{B} \rangle := \sum_{i_1}^{m_1} \sum_{i_2}^{m_2} \cdots \sum_{i_d}^{m_d} \mathbf{A}_{i_1 i_2 \ldots i_d} \mathbf{B}_{i_1 i_2 \cdots i_d} \tag{1}$$

inducing the generalization of the Frobenius-norm through

$$\|\mathbf{A}\|_F^2 := \sqrt{\langle \mathbf{A}, \mathbf{A} \rangle}$$

making $\mathbb{R}^{m_1 \times \cdots \times m_d}$ a Banach space, i.e. a complete normed vector space.

For two vectors $\mathbf{u}_1$ and $\mathbf{u}_2$ we denote the outer product with $\mathbf{u}_1 \circ \mathbf{u}_2 := \mathbf{u}_1 \mathbf{u}_2^T$
such that $(\mathbf{u}_1 \circ \mathbf{u}_2)_{i_1 i_2} = (\mathbf{u}_1)_{i_1} (\mathbf{u}_2)_{i_2}$.
We generalize this concept to tensors and define:

$$(\mathbf{u}^{(1)} \circ \mathbf{u}^{(2)} \circ \cdots \circ \mathbf{u}^{(d)})_{i_1 i_2 \ldots i_d} = \mathbf{u}^{(1)}_{i_1} \mathbf{u}^{(2)}_{i_2} \cdots \mathbf{u}^{(d)}_{i_d}$$

In the following section $\mathbf{X}$ will always denote a tensor of order d, i.e. $\mathbf{X} \in \mathbb{R}^{m_1 \times \cdots \times m_d}$.
We now want to define an operation similar to the matrix product between a tensor and a matrix. First note that for two matrices A, B of appropiate sizes the rows of the matrix product $A \cdot B$ are linear combinations of the rows ('mode-1 vectors') of B. Likewise the columns of $B \cdot A^T$ are linear combinations of the columns ('mode-2 vectors') of B.
We can generalize the concept of linear combinations of 'mode-n vectors' to tensors.

Definition 2.1 (n-mode product).
Let $\boldsymbol{X} \in \mathbb{R}^{m_1 \times \cdots \times m_d}$ *and* $U \in \mathbb{R}^{k_n \times m_n}$.
We define $\boldsymbol{X} \times_n U \in \mathbb{R}^{m_1 \times m_2 \times \cdots \times m_{n-1} \times k_n \times m_{n+1} \times \cdots \times m_d}$ *as*
$(\boldsymbol{X} \times_n U)_{i_1 i_2 \ldots i_{n-1} j_n i_{n+1} \ldots i_d} := \sum_{i_n} \boldsymbol{X}_{i_1 i_2 \ldots i_{n-1} i_n i_{n+1} \ldots i_d} U_{j_n i_n}$

Note that for matrices A, B, $B \times_1 A = A \cdot B$ and $B \times_2 A = B \cdot A^T$.

The notion of a *tensor rank* similar to a *matrix rank* is ambiguous, there are two common ways to capture the concept of low-rank tensors:
The multilinear *n-rank* (Tucker rank) and the (CP-)*rank*.

First, by fixing all but one of the indices of a tensor we get the n-mode vectors:

Definition 2.2.
$\boldsymbol{X}_{i_1,i_2,\dots,i_{n-1},\,\cdot\,,i_{n+1},\dots,i_d} \in \mathbb{R}^{m_n}$ *is called a n-mode vector of R.*

Now the *n-rank* is defined as the dimension of the vector space spanned by the n-mode vectors, or more formally:

Definition 2.3.
The matrix unfolding $X_{(n)} \in \mathbb{R}^{m_n \times (m_{n+1} m_{n+2} \cdots m_d m_1 \cdots m_{n-1})}$ *is defined as* $(X_{(n)})_{i_n ind(i_1,\dots,i_d)} = (\boldsymbol{X})_{i_1 i_2 \dots i_d}$*, i.e. contains the element* $(\boldsymbol{X})_{i_1 i_2 \dots i_d}$ *at index* $(i_n, ind(i_1, \dots i_{n-1} i_{n+1} \dots, i_d))$ *where ind is a fixed indexing of* $\{1,\dots,m_1\} \times \cdots \times \{1,\dots,m_{n-1}\} \times \{1,\dots,m_{n+1}\} \times \cdots \times \{1,\dots,m_d\}$*.*
For notational simplicity we also define the operator $\boldsymbol{P}_n(\boldsymbol{X}) := X_{(n)}$*.*

Definition 2.4 (n-rank)**.**
We set $\operatorname{rank}_n(\boldsymbol{X}) = \operatorname{rank}(\boldsymbol{X}_{(n)})$*.*

On the other hand, one could define the rank of a tensor by the minimal number of rank -1 tensors needed to yield the tensor in a linear combination:

Definition 2.5.
$\boldsymbol{X}$ *has a* rank *of 1 iff it is the outer product of d vectors* $\boldsymbol{u}^{(1)}, \boldsymbol{u}^{(2)}, \dots, \boldsymbol{u}^{(d)}$*,*
i.e. $\boldsymbol{X} = \boldsymbol{u}^{(1)} \circ \boldsymbol{u}^{(2)} \circ \cdots \circ \boldsymbol{u}^{(d)}$*.*

Definition 2.6 (rank)**.**
We set $\operatorname{rank}(\boldsymbol{X}) = \min r$ *s.t.* $\sum_{k=1}^{r} \boldsymbol{u}^{(r,1)} \circ \boldsymbol{u}^{(r,2)} \circ \cdots \circ \boldsymbol{u}^{(r,d)}$ *for any set of vectors* $\boldsymbol{u}^{(r,j)}$*.*

Unfortunatly, finding rank($\mathbf{X}$) of an arbitrary tensor $\mathbf{X}$ is np-hard and thus often less practical than the n-rank [3].

2.2 Singular Value Decomposition

In data-driven tasks the singular value decomposition, *SVD* for short, is one of the most important mathematical tools building the foundation for many modern systems and applications. The SVD is a matrix decomposition that exists for every matrix. It can be used to compute optimal matrix approximations and to eliminate noise in low rank matrices. Moreover, it forms the basis of the influential *Principal Component Analysis* (PCA) that can be used to find the most statistically descriptive factors and dominant patterns in the data [4].

Mathematically, the SVD of a matrix $Z \in \mathbb{R}^{m \times n}$ is given by

$$Z = U \cdot D \cdot V^T$$

where $U \in \mathbb{R}^{m \times m}$ and $V \in \mathbb{R}^{n \times n}$ are orthogonal, i.e. whose columns form an orthonormal system. $D \in \mathbb{R}^{m \times n}$ is a diagonal matrix with non-negative entries, such that $D_{ii} \geq D_{jj}$ for $i \geq j$. $\sigma_i := D_{ii}$ is called the i-th singular value of Z and $U_{\cdot,i}$ ($V_{\cdot,i}$) is called the i-th left (right) singular vector of Z.

One of the defining properties of the SVD is the ability to provide optimal low rank matrix approximations.

Theorem 2.7 (Eckart-Young [5]).
The best rank-k approximation w.r.t. the Frobenius norm

$$\min_{\tilde{Z} \in \mathbb{R}^{m \times n}} \left\| Z - \tilde{Z} \right\|_F^2$$
$$s.t. \ \operatorname{rank}(\tilde{Z}) = k$$

is given by the rank-k truncation of the SVD of Z

$$\hat{Z} = U\hat{D}V^T$$

where $Z = UDV^T$ is the SVD of Z and $\hat{D}_{ii} = D_{ii}$ for $i \leq k$, $\hat{D}_{ii} = 0$ else.

On a site note we define two concepts closely related to the SVD that will be used later. In the following, let $Z \in \mathbb{R}^{m \times n}$ and denote its SVD with $Z = UDV^T$. First, we can use the singular values to define the nuclear norm, a tight convex relaxation of the matrix rank.

Definition 2.8.
Let $\|Z\|_* := \sum_{i=1}^{min(n,m)} \sigma_i$.

Finally we define the soft-treshold SVD which arises naturally in the context of Matrix Completion.

Definition 2.9.
We set $S_\lambda(Z) := UD_\lambda V^T$ where D_λ is a diagonal matrix with $(D_\lambda)_{ii} = (D_{ii} - \lambda)_+ = max(D_{ii} - \lambda, 0)$.

2.3 Tensor Decompositions

There are several types of tensor decompositions preserving properties of the SVD for matrices. The goal of Tensor Decomposition is to give insights into the patterns found in the tensors. Choosing only the most important patterns found in a decomposition will result in low rank approximations of tensors. Unfortunately an optimal tensor approximation w.r.t a given fixed n-rank or rank can not be found as elegantly as in the matrix case. We will take a closer look at two decompositions, the CP Decomposition and the

Tucker Decomposition / HOSVD.
A more complete overview on similar methods can be found in [3].
We also briefly note that there are other promising tensor decompositions and approximations, e.g. based on the *t-SVD* [6] or the *tensor train decomposition* [7] that try to capture a different kind of tensor rank.

2.3.1 CP Decomposition

One way to approach a generalization of the SVD is to note that the SVD of $Z \in \mathbb{R}^{m_1 \times m_2}$ can be rewritten as:

$$Z = UDV^T = \sum_{k=1}^{\min(n,m)} \sigma_k U_{\cdot,k} V_{\cdot,k}^T = \sum_{k=1}^{\min(n,m)} \sigma_k U_{\cdot,k} \circ V_{\cdot,k}$$

Each summand has the interpretation of a distinct influence or mechanism that contributed to Z where σ_k denotes the statistical importance of the factor.
The CP Decomposition generalizes the above to tensors. Historically it is also called the *canonical decomposition* (CANDECOMP) or *parallel factors model* (PARAFAC).

Definition 2.10 (CP).
Let $\boldsymbol{Z} \in \mathbb{R}^{m_1 \times \cdots \times m_d}$ be a d-order tensor.
A CP of $\boldsymbol{Z}$ is a decomposition $\boldsymbol{Z} = \sum_{r=1}^{R} \sigma_r \boldsymbol{u}_r^{(1)} \circ \boldsymbol{u}_r^{(2)} \circ \cdots \circ \boldsymbol{u}_r^{(d)}$
with minimal R where $\left\| \boldsymbol{u}_r^{(n)} \right\|_2 = 1$.
Depending on the application, some orthogonality constraints may apply.

Note that unlike the SVD, the CP is not unique.
A *CP* of $\mathbf{Z}$ can only be found numerically.
Given a CP of $\mathbf{Z}$, we are also interested in the *truncated CP*, a fixed rank approximation

$$\hat{\mathbf{Z}} = \sum_{r=1}^{k} \sigma_r \mathbf{u}_r^{(1)} \circ \mathbf{u}_r^{(2)} \circ \cdots \circ \mathbf{u}_r^{(d)}$$

with $\text{rank}(\hat{\mathbf{Z}}) = k < R$.
The problem of a 'best truncated CP' which minimizes $\left\| \hat{\mathbf{Z}} - \mathbf{Z} \right\|_F$ is studied in [8].
Without constraints a best *truncated CP* for a fixed rank may not even exist, altough by imposing orthogonality constraints on $\mathbf{u}_i^{(n)}$ the existence can be guaranteed and a solution can be found numerically.

2.3.2 Tucker Decomposition

An other way to approach a generalization of the SVD is to use the *higher order SVD* (HOSVD) [9] which computes a special case of the Tucker Decomposition. This special decomposition will also be called the HOSVD to avoid confusion.
First note that the SVD of $Z \in \mathbb{R}^{m_1 \times m_2}$ can be rewritten as:

$$Z = UDV^T = D \times_1 U \times_2 V$$

The HOSVD generalizes the concept of applying matrix products to a 'core matrix'. Instead tensor-matrix products are used.

Definition 2.11 (HOSVD).
Let $\boldsymbol{Z} \in \mathbb{R}^{m_1 \times \cdots \times m_d}$ be a d-order tensor.
The HOSVD of $\boldsymbol{Z}$ is given by the Tucker decomposition $\boldsymbol{Z} = \boldsymbol{\Sigma} \times_1 U^{(1)} \times_2 U^{(2)} \cdots \times_d U^{(d)}$ where additionally $U^{(k)} \in \mathbb{R}^{m_k \times m_k}$ are orthogonal matrices and $\boldsymbol{\Sigma} \in \mathbb{R}^{m_1 \times \cdots \times m_d}$ is all-orthogonal, i.e. all subtensors $\boldsymbol{\Sigma}_{i_n = \alpha}$ (where one index is fixed) are orthogonal, i.e. $\langle \boldsymbol{\Sigma}_{i_n=\alpha}, \boldsymbol{\Sigma}_{i_n=\beta} \rangle = 0$ for $\alpha \neq \beta$.

Note that we do not restrict ourselves to *pseudodiagonal* $\boldsymbol{\Sigma}$, i.e. where only $\boldsymbol{\Sigma}_{kk\ldots k}$ are nonzero because then too few degrees of freedom would remain and not every tensor in $\mathbb{R}^{m_1 \times \cdots \times m_d}$ could be represented (Since a pseudodiagonal $\boldsymbol{\Sigma}$ contains $m = min(m_1, \ldots, m_d)$ non-zero entries this decomposition would only have $m(1-d(m+1)/2+\sum_{i=1}^{d} m_i)$ degrees of freedom which for $d \geq 3$ is less than the $\Pi_{i=1}^{d} m_i$ independent entries of the original tensor [9]).
This unique decomposition has a lot of similar properties to the matrix SVD and is also easily computable because the n-mode-singular vectors $U^{(n)}$ of $\mathbf{Z}$ are just the left singular vectors of $\mathbf{Z}_{(n)}$, hence we only have to compute d matrix SVDs.
Unfortunately unlike the matrix SVD where Theorem 2.7 holds, for tensors in general the best approximation w.r.t a fixed Tucker rank $[k_1, \ldots, k_d]$

$$\min_{\tilde{\mathbf{Z}} \in \mathbb{R}^{m_1 \times \cdots \times m_d}} \left\| \mathbf{Z} - \tilde{\mathbf{Z}} \right\|_F^2 \qquad (2)$$
$$\text{s.t. } \operatorname{rank}_n(\tilde{Z}) = k_n$$

is not necessarily given by a *truncated HOSVD* where the respective values in $\boldsymbol{\Sigma}$ are set to zero that correspond to the n-mode singular vectors $U_{\cdot i}^{(n)}$ with $i > k_n$.
However the resulting approximation is very close to the real optimal solution and tight upper bounds on the resulting error can be shown.
Additionally, the *Higher Order Orthogonal Iteration* (HOOI) [10] can be used to iteratively improve the result obtained by the HOSVD. It can be shown that in most cases HOOI converges to an optimal solution of (2). HOOI is based on an alternating optimization of the factors $U^{(k)}$ by fixing all but one factor at a time.

Algorithm *HOOI*

INPUT: $\mathbf{Z}$, initial factors $U^{(k)}$ for $k \in \{1, \dots, d\}$

```
repeat
    for $i \in \{1, 2 \dots K\}$ do
        $\tilde{\mathbf{U}} \leftarrow \mathbf{Z}$
        for $j \in \{1, 2 \dots K\} \setminus \{i\}$ do
            $\tilde{\mathbf{U}} \leftarrow \tilde{\mathbf{U}} \times_j (U^{(j)})^T$
        end for
        $U^{(i)} \leftarrow \text{argmin}_U \left\| \tilde{\mathbf{U}} \times_i U^T \right\|$ s.t. $U$ is orthogonal.
    end for
until Convergence
```

Note that $\left\| \tilde{\mathbf{U}} \times_i U^T \right\|$ under these constraints is minimized by the left singular vectors of $\mathbf{P}_i(\tilde{\mathbf{U}})$ and can be obtained by calculating a matrix SVD [10].

2.4 Matrix Completion

Let $R \in \mathbb{R}^{m \times n}$ be a ground truth matrix whose entries are partially observed. The set of known entries is given by $\Omega \subseteq \{1, \dots, m\} \times \{1, \dots, n\}$ and the projection onto this set, setting all other values to zero, is denoted as P_Ω. The values that are not observed are denoted by $\Omega^\perp = \{1, \dots, m\} \times \{1, \dots, n\} \setminus \Omega$. Throughout the thesis we will write R_Ω as a shorthand for $P_\Omega(R)$, the observed entries of R.
The goal of matrix completion is to find a low rank matrix which also explains the observations R_Ω, i.e. while having a small loss (e.g. quadratic loss).
Using a simple model, a solution can be obtained through the solution of the following optimization problem:

$$\begin{aligned} \min_{Z \in \mathbb{R}^{m \times n}} \quad & \text{rank}(Z) \\ \text{s.t.} \quad & \| R_\Omega - P_\Omega(Z)) \|_F^2 \leq \delta \end{aligned} \tag{3}$$

If we approximate rank(Z) with the nuclear norm $\|Z\|_*$ the Lagrange form of (3) is a convex optimization problem

$$\min_{Z \in \mathbb{R}^{m \times n}} \frac{1}{2} \| R_\Omega - P_\Omega(Z)) \|_F^2 + \lambda \|Z\|_* \tag{4}$$

and thus has a unique solution.

2.5 Tensor Completion

Since Tensor Completion has the same goal of imputing missing values our considerations mostly carry over from *Matrix Completion.*

This time let $\mathbf{R} \in \mathbb{R}^{m_1 \times m_2 \times \cdots \times m_d}$ be a ground truth tensor whose entries are partially observed. The set of know entries is denoted by $\Omega \subseteq \{1, \ldots, m_1\} \times \cdots \times \{1, \ldots, m_d\}$, the projection onto them by $\mathbf{P}_\Omega$, $\mathbf{R}_\Omega := \mathbf{P}_\Omega(\mathbf{R})$.
Again we want to find a low rank tensor which explains our observations well.
For the sake of simplicity we will mainly focus on the case where $d = 3$, i.e. we observe data with 3 dimensions, capturing the interactions of three factors. Higher order Tensor Completion ($d > 3$) can however be solved with the same methods.
A basic model, similar to (3), is given by

$$\begin{aligned} \min_{\mathbf{Z} \in \mathbb{R}^{m_1 \times \cdots \times m_d}} \quad & r(\mathbf{Z}) \\ \text{s.t. } & \|\mathbf{R}_\Omega - P_\Omega(\mathbf{Z})\|_F^2 \leq \delta \end{aligned} \tag{5}$$

where $r(\mathbf{Z})$ can either be $\operatorname{rank}(\mathbf{Z})$ or the sum of the n-ranks ($\sum_{k=1}^d \operatorname{rank}_k(\mathbf{Z})$). When we choose a convex regularizer $\mathcal{R}$ (e.g. $\mathcal{R}(\mathbf{Z}) = \sum_{k=1}^d \|\mathbf{P}_k(\mathbf{Z})\|_*$) we can again make the problem convex:

$$\min_{\mathbf{Z} \in \mathbb{R}^{m_1 \times \cdots \times m_d}} \frac{1}{2}\|\mathbf{R} - \mathbf{Z}\|_F^2 + \lambda \mathcal{R}(\mathbf{Z}) \tag{6}$$

2.6 Subgradients

For later proofs we will need the subgradient, a handy tool for the optimization of convex functions. With the help of subgradients a simple optimality criterion can be given.

Let $f : \mathbb{R}^n \to \mathbb{R}, x \mapsto f(x)$ be a real valued function.

Definition 2.12 (Subgradient).
g is a subgradient of f at x if $f(y) \geq f(x) + g(x)^T(y - x)$ for all $y \in \mathbb{R}^n$.

Definition 2.13 (Subdifferential).
$\partial f([x]) = \partial f(x) = \{g | g \text{ is subgradient of } f \text{ at } x\}$ is called the subdifferential of f at x.

The subdifferential shares basic properties with the differential, e.g.

Remark 2.14 (Linearity).
$\partial(\alpha f(x) + h(x)) = \{\alpha g_1 + g_2 | g_1 \in \partial f(x), g_2 \in \partial h(x)\} = \alpha \partial f(x) + \partial h(x)$.

Subdifferentials are especially suited to describe convex functions, as can be seen by:

Proposition 2.15.
*If f is convex $\partial f(x)$ is nonempty for each $x \in \mathbb{R}^n$
and $\partial f(x) = \{\nabla f(x)\}$ if f is differentiable at $x \in \mathbb{R}^n$.*

Most importantly we obtain the following optimality criterion for convex functions that follows directly from Definition 21

Lemma 2.16.
If f is convex $0 \in \partial f(\hat{x}) \Leftrightarrow f(\hat{x}) \leq f(x)$ for all $x \in \mathbb{R}^n$.

3 Related Work

In this chapter we will review relevant works from the field of Matrix Completion and Tensor Completion. We will start with Matrix Completion methods including Soft-Impute and go into detail about OMIC and BOMIC. The chapter ends with a survey of current Tensor Completion models and methods.

3.1 Soft Impute

Since the problem (4) is convex it can be solved, for example, with (Sub-)Gradient Descent [11]. However a more elegant solution is given by the algorithm *Soft-Impute* [12] which will be explained below. Some concepts will still be relevant in our algorithm later on.

As a first step we consider the fully known case, i.e. $\Omega = \{1, \dots, m\} \times \{1, \dots, n\}$.
First note that (3) simplifies to

$$\begin{aligned} \min_{Z \in \mathbb{R}^{m \times n}} \quad & \operatorname{rank}(Z) \\ \text{s.t. } & \|R - Z\|_F^2 \leq \delta \end{aligned} \tag{7}$$

whose optimal solution can be found with a truncated SVD of R (see Theorem 2.7).
In the fully known case The nuclear norm regularized problem (4) simplifies to:

$$\min_{Z \in \mathbb{R}^{m \times n}} \frac{1}{2}\|R - Z\|_F^2 + \lambda \|Z\|_* \tag{8}$$

Proof A.1 in [12] shows that (8) can be solved with the soft-threshhold SVD $S_\lambda(R)$ (see Definition 2.9).

Finally, based on the above, we find a general solution to (4) including the case $\Omega \subsetneq \{1, \dots, m\} \times \{1, \dots, n\}$ with *Soft-Impute* [12]

```
Algorithm SOFT-Impute
INPUT: $\lambda_i$, $R_\Omega$, $\varepsilon$

Initialize $Z^{old}, Z^{old} \leftarrow \mathbf{0}$
for $i \in \{1, 2 \dots K\}$ do
    repeat
        $Z^{old} \leftarrow Z^{new}$
        $Z^{new} \leftarrow S_{\lambda_i}(R_\Omega(X) - P_{\Omega^\perp}(Z^{old}))$
    until $\frac{\|Z^{old} - Z^{new}\|_F}{\|Z^{old}\|_F} < \varepsilon$
end for
```

where different values for the regularization parameter λ can be tested to select the best one, e.g. using cross validation.

3.2 Orthogonal Inductive Matrix Completion

The model used in (4) can be improved with more sophisticated approaches like the recently developed *Orthogonal Inductive Matrix Completion* (OMIC) [1]. Its more complexe model allows for greater interpretability and the potential to incorporate side information to aid the prediction. In the context of recommender systems this side information could for example be additional information about the items, e.g. which genre a movie or song is in. OMIC imputes the missing values by finding a solution to

$$\min_{M} \frac{1}{2} \left\| R_\Omega - P_\Omega\Big(\sum_{k=1,l=1}^{K,L} X^{(k)} M^{(k,l)} (Y^{(l)})^T \Big) \right\|_F^2 + \sum_{k=1}^{K} \sum_{l=1}^{L} \lambda_{k,l} \left\| M^{(k,l)} \right\|_* \tag{9}$$

where the auxillary matrices $X^{(k)} \in \mathbb{R}^{m \times d_1^{(k)}}$ and $Y^{(l)} \in \mathbb{R}^{n \times d_2^{(l)}}$ form orthonormal bases of their respective spaces, i.e. $(X^{(k1)}_{\cdot,j_1})^T (X^{(k2)}_{\cdot,j_2}) = \delta_{k1,k2}\, \delta_{j_1,j_2}$ and $span_{k,j}(X^{(k)}_{\cdot,j}) = \mathbb{R}^n$ (likewise $(Y^{(l1)}_{\cdot,j_1})^T (Y^{(l2)}_{\cdot,j_2}) = \delta_{l1,l2}\, \delta_{j_1,j_2}$ and $span_{k,j}(Y^{(l)}_{\cdot,j}) = \mathbb{R}^m$).
This ensures the orthogonality of the subspaces $\{X^{(k)} M^{(k,l)} (Y^{(l)})^T | M \in \mathbb{R}^{d_1^{(k)} \times d_2^{(l)}}\}$ and allows R to have a unique representation as $R = \sum_{k=1,l=1}^{K,L} X^{(k)} R^{(k,l)} (Y^{(l)})^T$.
The $\lambda_{k,l}$ can be searched for by cross validation. Prior knowledge can reduce the computational costs by tying parameters with each other and setting others to zero.

Moreover, the noteworthy special case *BOMIC* allows for joint training of regression based bias terms and a low rank bias-free matrix completion term.
In BOMIC we set $X^{(1)} = \frac{1}{\sqrt{m}} \mathbf{1}^T$, $Y^{(1)} = \frac{1}{\sqrt{n}} \mathbf{1}^T$ and $X^{(2)}$ and $Y^{(2)}$ to their respective orthogonal complements. The resulting model is equivalent to optimizing

$$\begin{aligned} \min_{c,\mathbf{u},\mathbf{m},S} \frac{1}{2} \left\| R_\Omega - P_\Omega(c\mathbf{1}\mathbf{1}^T + \mathbf{u}\mathbf{1}^T + \mathbf{1}\mathbf{m}^T + S) \right\|_F^2 & + \\ \lambda_1 |c| + \lambda_2 \|\mathbf{u}\|_2 + \lambda_3 \|\mathbf{m}\|_2 + \lambda_4 \|S\|_* & \end{aligned} \tag{10}$$

under orthogonality constraints ($\mathbf{u}$, $\mathbf{m}$, $S_{\cdot,j}$ and $S_{i,\cdot}$ sum up to zero for all i, j).
The prediction $f_{i,j}$ can thus be written as a sum of a zero order term (constant), first order terms (individual biases) and a second order term:

$$f_{i,j} = c + \mathbf{u}_i + \mathbf{m}_j + S_{i,j}$$

3.3 Tensor Completion

There are many different approaches to Tensor Completion, a summary of recent methods can be found in [2]. Most approaches fall into two categories - there are methods which are based on tensor decompositions and methods that find fitting tensors directly, regularized by different regularizations.

First, Tensor Decomposition methods can be extended to also handle missing data. One common way to solve the resulting optimization problems is to use repeated Imputation, very similar to Soft-Impute. The decomposition is used indirectly by repeatedly alternating between imputation of the known entries and approximation of the current iterate until convergence is reached.
There are also more direct optimization approaches, simply ignoring the missing values. For example there are several methods based on the CP decomposition (Section 2.3.1), including the gradient-based *CP-WOPT* [13] and the Bayesian approach *FBCP* [14].
In [15] a method based on the Tucker decomposition (similar to Section 2.3.2) was used to find tensors with low n-rank.

An other approach is to solve optimization problems similar to (6) directly, often using some generalization of the nuclear norm. A popular generalization is the *overlap nuclear norm* which is a sum of the nuclear norms of the matrix unfoldings $\sum_{k=1}^{d} \gamma_k \|\mathbf{P}_k(bM)\|_*$ [16]. Solutions to the resulting problem can be found using *Block Cordinate Descent* [17], the *Alternating Direction Method of Multipliers* (ADMM) [18] or the *Frank-Wolfe Algorithm* [19]. [18] also explores the *latent nuclear norm* which instead of finding a tensor that exhibits a low rank in each dimension, searches for tensors that can be represented as a sum of several tensors each having a low rank in only one dimension.
As mentioned in Chapter 2.3, we briefly note that there also methods based on the Tensor Train Decomposition [20] and the tSVD [21].

4 Orthogonal Inductive Tensor Completion

In this section we will present the new tensor completion framework OTIC which extends OMIC [1] to tensors. An algorithmic solution to the general model is developed, putting a special focus on the proposed algorithm BOTIC. Finally, different regularizations and warm starts are discussed.

4.1 The Model

We now introduce a generalization of *OMIC* that is called *Orthogonal Inductive Tensor Completion* (OTIC) which for $d = 3$ is given by

$$\min_{\mathbf{M}} \mathcal{L}(\mathbf{R}_\Omega, \mathbf{M}, \Lambda) \text{ with} \tag{11}$$

$$\mathcal{L}(\mathbf{R}_\Omega, \mathbf{M}, \Lambda) := \frac{1}{2}\left\| R_\Omega - P_\Omega(\sum_{k_1,k_2,k_3=1}^{K_1,K_2,K_3} \mathbf{M}^{(k_1,k_2,k_3)} \times_1 X^{(k_1)} \times_2 Y^{(k_2)} \times_3 Z^{(k_3)})) \right\|_F^2 + \sum_{k_1,k_2,k_3=1}^{K_1,K_2,K_3} \lambda_{k_1 k_2 k_3} \mathcal{R}(\mathbf{M}^{(k_1,k_2,k_3)})$$

where $\mathcal{R}$ is a convex regularizer, $\Lambda \in \mathbb{R}^{K_1 \times K_2 \times K_3}$, $\Lambda_{k_1 k_2 k_3} = \lambda_{k_1 k_2, k_3}$ are the regularization parameters and $\mathbf{M}^{(k_1,k_2,k_3)} = \mathbf{M} \times_1 (X^{(k_1)})^T \times_2 (Y^{(k_2)})^T \times_3 (Z^{(k_3)})^T$. Further explanations are given in the next section.

For now we will focus on the special case *BOTIC* (an extension of *BOMIC*) by setting $X^{(1)} = \frac{1}{\sqrt{m_1}}\mathbf{1}^T$, $Y^{(1)} = \frac{1}{\sqrt{m_2}}\mathbf{1}^T$, $Z^{(1)} = \frac{1}{\sqrt{m_3}}\mathbf{1}^T$ and $X^{(2)}$, $Y^{(2)}$, $Z^{(2)}$ to their respective orthogonal complements. This allows for the joint training of individual biases, matrix completion terms and an aditional tensor completion term modelling pure order three interactions between all dimensions.

We will use a regularizer that is an extension of the nuclear norm. This means $\mathcal{R}(\mathbf{Z}) = \|\mathbf{Z}\|_*$ for $\mathbf{Z} \in \mathbb{R}^{k_1 \times k_2 \times k_3}$ where $k_1 = 1$, $k_2 = 1$ or $k_3 = 1$, i.e. when $\mathbf{Z}$ can be be understood as a matrix, vector or scalar respectively.

In that case the model is equivalent to optimizing a prediction function

$$f_{i,j,k} = c + \mathbf{b}_i^1 + \mathbf{b}_j^2 + \mathbf{b}_k^3 + S_{i,j}^1 + S_{j,k}^2 + S_{i,k}^3 + \mathbf{T}_{i,j,k} \tag{12}$$

under orthogonality constraints ($\mathbf{b}^l$, $S^l_{\cdot,j}$, $S^l_{i,\cdot}$, $\mathbf{T}_{\cdot,j,k}$, $\mathbf{T}_{i,\cdot,k}$ and $\mathbf{T}_{i,j,\cdot}$ sum up to zero for all l, i, j, k) and w.r.t an regularizer that is given by

$$\bar{\lambda}_0|c| + \sum_{l=1}^{3} \bar{\lambda}_{1,l} \left\| \mathbf{b}^l \right\|_2 + \sum_{l=1}^{3} \bar{\lambda}_{2,l} \|S\|_* + \bar{\lambda}_3 \mathcal{R}(t).$$

In addition to first order terms ($\mathbf{b}$) and second order terms (S) this model also takes a third order term ($\mathbf{T}$) into account.
To reduce the need for cross validation and thus computation time we will set the regularization for zeroth and first order terms zero and tie the second order regularization parameters together,
i.e. $\bar{\lambda}_0 = \bar{\lambda}_{11} = \bar{\lambda}_{12} = \bar{\lambda}_{13}$ $(= \lambda_{111} = \lambda_{211} = \lambda_{121} = \lambda_{112}) = 0$, $\bar{\lambda}_{2i} = c_i\lambda_1$ and $\lambda_2 = \bar{\lambda}_3$ $(= \lambda_{222})$ where c_i is a universal constant depending on the size of the ground truth tensor.
As a result only two regularization parameters have to be choosen, λ_1 and λ_2.

4.2 The Algorithm

The first step to a solution of (11) is to find a solution for the fully known case were all entries of a tensor $\mathbf{R} \in \mathbb{R}^{m_1 \times m_2 \times m_3}$ are observed.
Note that the subspaces

$$\mathcal{S}_{k_1,k_2,k_3} := \{\mathbf{M} \times_1 X^{k_1} \times_2 Y^{k_2} \times_3 Z^{k_3} | \mathbf{M} \in \mathbb{R}^{d_1^{k_1} \times d_2^{k_2} \times d_3^{k_3}}\}$$

are orthogonal w.r.t the inner product defined in (1).
The projection onto those subspaces is given by

$$\mathbf{\Pi}^{k_1k_2k_3}(\mathbf{R}) := \Pi_{\mathcal{S}_{k_1,k_2,k_3}}(\mathbf{R}) = \Big[\mathbf{R} \times_1 X^{(k_1)} \times_2 Y^{(k_2)} \times_3 Z^{(k_3)}\Big] \times_1 (X^{(k_1)})^T \times_2 (Y^{(k_2)})^T \times_3 (Z^{(k_3)})^T$$

where

$$\mathbf{P}^{k_1k_2k_3}(\mathbf{R}) := \mathbf{R} \times_1 X^{(k_1)} \times_2 Y^{(k_2)} \times_3 Z^{(k_3)} \in \mathbb{R}^{d_1^{k_1} \times d_2^{k_2} \times d_3^{k_3}}$$

is the unique representation of $\mathbf{\Pi}^{k_1k_2k_3}(\mathbf{R})$ w.r.t. the base vectors $X^{(k_1)}_{\cdot j_1}$, $Y^{(k_2)}_{\cdot j_2}$ and $Z^{(k_3)}_{\cdot j_3}$:

$$\mathbf{\Pi}^{k_1k_2k_3}(\mathbf{R}) = \sum_{j_1, j_2, j_3=1}^{d_1^{k_1}, d_2^{k_2}, d_3^{k_3}} \mathbf{P}^{k_1k_2k_3}(\mathbf{R})(\mathbf{M})_{j_1j_2j_3} \, X^{(k_1)}_{\cdot j_1} \circ Y^{(k_2)}_{\cdot j_2} \circ Z^{(k_3)}_{\cdot j_3}$$

The problem (11) can be rewritten as:

$$\min_{\mathbf{M} \in \mathbb{R}^{m_1 \times m_2 \times m_3}} \frac{1}{2} \left\| \mathbf{R} - \sum_{k_1,k_2,k_3=1}^{K_1,K_2,K_3} \mathbf{\Pi}^{k_1k_2k_3}(\mathbf{M}) \right\|_F^2 + \sum_{k_1,k_2,k_3=1}^{K_1,K_2,K_3} \lambda_{k_1k_2k_3} \mathcal{R}(\mathbf{P}^{k_1k_2k_3}(\mathbf{M})) \quad (13)$$

Because the $\mathcal{S}_{k_1,k_2,k_3}$ are mutually orthogonal we can uniquely decompose
$\mathbf{R} = \sum_{k_1,k_2,k_3=1}^{K_1,K_2,K_3} \mathbf{\Pi}^{k_1k_2k_3}(\mathbf{R})$ and thus find an optimal solution by solving
$K_1 \cdot K_2 \cdot K_3$ independent optimization problems of the form

$$\begin{aligned} &\min_{\mathbf{M}\in\mathbb{R}^{m_1\times m_2\times m_3}} \frac{1}{2}\left\|\mathbf{\Pi}^{k_1k_2k_3}(\mathbf{R}) - \mathbf{\Pi}^{k_1k_2k_3}(\mathbf{M})\right\|_F^2 + \lambda_{k_1k_2k_3}R(\mathbf{\Pi}^{(k_1k_2k_3)}(\mathbf{M})) \\ = &\min_{\mathbf{M}\in\mathbb{R}^{d_1^{k_1}\times d_2^{k_2}\times d_3^{k_3}}} \frac{1}{2}\left\|\mathbf{\Pi}^{k_1k_2k_3}(\mathbf{R}) - \mathbf{M}\right\|_F^2 + \lambda_{k_1k_2k_3}R(\mathbf{M}). \end{aligned} \tag{14}$$

Denote the optimal solution to (14) with $\mathbf{S}_{\lambda_{k_1k_2k_3}}(\mathbf{\Pi}^{k_1k_2k_3}(\mathbf{R}))$
and the optimal solution to (13) with $\mathbf{S}_\Lambda(\mathbf{R})$.
Note that:

$$\mathbf{S}_\Lambda(\mathbf{M}) = \sum_{k_1,k_2,k_3=1}^{K_1,K_2,K_3} S_{\lambda_{k_1k_2k_3}}(\mathbf{M} \times_1 (X^{k_1})^T \times_2 (Y^{k_2})^T \times_3 (Z^{k_3})^T) \times_1 X^{k_1} \times_2 Y^{k_2} \times_3 Z^{k_3} \tag{15}$$

How to find the solution $\mathbf{S}_{\lambda_{k_1k_2k_3}}(\mathbf{\Pi}^{(k_1k_2k_3)}(\mathbf{R}))$ to (14) will be described in Section 4.4. Building on the solution of the fully known case (13), a solution to the partially observed case ($\Omega \subsetneq \{1,\dots,m_1\} \times \cdots \times \{1,\dots,m_d\}$) can be found by first setting all unobserved values to zero (or any other constant value) and then repeatedly alternating between imputation of the observed values and application of the solution of the fully known case, i.e.

$$\begin{aligned} \mathbf{M}^0 &= \mathbf{0} \\ \mathbf{M}^{i+1} &= \mathbf{S}_\Lambda(\mathbf{R}_\Omega + \mathbf{P}_{\Omega^\perp}(\mathbf{M}^i)). \end{aligned}$$

Proposition 4.1.
If $\mathcal{R}$ is the overlap nuclear norm, i.e. $\mathcal{R}(\boldsymbol{M}) = \sum_{k=1}^d \gamma_k \|\boldsymbol{P}_k(bM)\|_*$,
the sequence $\boldsymbol{M}^i$ defined as above converges to an optimal solution $\boldsymbol{M}^\infty$ of (11).

Proposition 4.2 (Worst Case Asymptotic Convergence)**.**
In this case, for a fixed Λ we get the following bound:

$$\mathcal{L}(\boldsymbol{M}^i) - \mathcal{L}(\boldsymbol{M}^\infty) \leq \frac{\left\|\boldsymbol{M}^0 - \boldsymbol{M}^\infty\right\|_F^2}{i+1}$$

These propositions are generalizations of similiar theorems that also hold for OMIC.
A proof of Proposition 4.1 can be found in Section 7.2 while the proof of Proposition 4.2 is exactly the same as the proof of the similar Theorem 2.2 in [1] for the matrix case.

Using Proposition 4.1 a solution to (11) can be computed using the following pseudocode

Algorithm *OTIC (simple)*

INPUT: $\mathcal{V}$, R_Ω, ε

OUTPUT: $\mathbf{M}^\Lambda$ and $\mathbf{P}^{k_1k_2k_3}(\mathbf{M}^\Lambda)$ for all k_1, k_2, k_3 and $\Lambda \in \mathcal{V}$

```
Initialize $\mathbf{M}^{new} \leftarrow \mathbf{0}$
for $\Lambda \in \mathcal{V}$ do
    repeat
        $\mathbf{M}^{old} \leftarrow \mathbf{M}^{new}$
        $\mathbf{M}^{new} \leftarrow \mathbf{S}_\Lambda(\mathbf{R}_\Omega(X) + \mathbf{P}_{\Omega^\perp}(\mathbf{M}^{old}))$
    until $\frac{\left\|\mathbf{M}^{old} - \mathbf{M}^{new}\right\|_F}{\left\|\mathbf{M}^{old}\right\|_F} < \varepsilon$
    $\mathbf{M}^\Lambda \leftarrow \mathbf{M}^{new}$
end for
```

where a solution will be calculated for all $\Lambda \in \mathcal{V}$.

Each subproblem is initialized with the result of the preceding subproblem.

The best solution can be selected using cross validation.

If $\mathcal{V}$ is a cross product, i.e. $\mathcal{V} = \bigtimes_{k_1,k_2,k_3} \mathcal{V}_{k_1,k_2,k_3}$ where $\mathcal{V}_{k_1,k_2,k_3}$ is a finte set of all possible values for $\lambda_{k_1k_2k_3}$, the runtime can be further improved by warm starts similar to [1]. First a set of tensors is computed using $\Lambda = p_{k_1,k_2,k_3}(\lambda), \lambda \in \mathcal{V}_{k_1,k_2,k_3}$ where

$$(p_{k_1,k_2,k_3}(\lambda))_{i_1i_2i_3} \coloneqq \begin{cases} \lambda \text{ for } i_1 = k_1, i_2 = k_2, i_3 = k_3 \\ \infty \text{ else} \end{cases}$$

that can be combined later.

```
Algorithm OTIC
INPUT: $\mathcal{V} = \bigtimes_{k_1,k_2,k_3} \mathcal{V}_{k_1,k_2,k_3}$, $R_\Omega$, $\varepsilon$
OUTPUT: $\mathbf{M}^\Lambda$ and $\mathbf{P}^{k_1 k_2 k_3}(\mathbf{M}^\Lambda)$ for all $k_1$, $k_2$, $k_3$ for all $\Lambda \in \mathcal{V}$
for $k_1 \in \{1, \dots, K_1\}$ do
    for $k_2 \in \{1, \dots, K_2\}$ do
        for $k_3 \in \{1, \dots, K_3\}$ do
            Initialize $\mathbf{M}^{new} \leftarrow \mathbf{0}$
            for $\lambda \in \mathcal{V}_{k_1,k_2,k_3}$ do
                repeat
                    $\mathbf{M}^{old} \leftarrow \mathbf{M}^{new}$
                    $\mathbf{M}^{new} \leftarrow \mathbf{S}_{p_{k_1 k_2 k_3}(\lambda)}(\mathbf{R}_\Omega(X) + \mathbf{P}_{\Omega^\perp}(\mathbf{M}^{old}))$
                until $\frac{\|\mathbf{M}^{old} - \mathbf{M}^{new}\|_F}{\|\mathbf{M}^{old}\|_F} < \varepsilon$
                $\mathbf{M}^\lambda_{(k_1,k_2,k_3)} \leftarrow \mathbf{M}^{new}$
            end for
        end for
    end for
end for
for $\Lambda \in \mathcal{V}$ do
    Initialize $\mathbf{M}^{new} \leftarrow \sum_{k_1,k_2,k_3=1}^{K_1,K_2,K_3} \mathbf{M}^{\Lambda_{k_1 k_2 k_3}}_{(k_1,k_2,k_3)}$
    repeat
        $\mathbf{M}^{old} \leftarrow \mathbf{M}^{new}$
        $\mathbf{M}^{new} \leftarrow \mathbf{S}_\Lambda(\mathbf{R}_\Omega(X) + \mathbf{P}_{\Omega^\perp}(\mathbf{M}^{old}))$
    until $\frac{\|\mathbf{M}^{old} - \mathbf{M}^{new}\|_F}{\|\mathbf{M}^{old}\|_F} < \varepsilon$
    $\mathbf{M}^\Lambda \leftarrow \mathbf{M}^{new}$
end for
```

Again remember that for *BOTIC* (12) only two hyperparameters have to be choosen and many $\Lambda_{k_1 k_2 k_3}$ are set to zero and can thus be ignored because $\mathbf{S}_{\lambda_{k_1,k_2,k_3}}(\mathbf{M})$ is the identity for $\lambda_{k_1 k_2 k_3} = 0$.
Since the only auxillary matrices used are the scaled $\mathbf{1}$ and its ortogonal complement we can simplify (15) further. Note that $\mathbf{M} \times_k \mathbf{1}^T$ has the effect of summing all entries over the k-th mode of $\mathbf{M}$, effectively reducing the number of dimensions by one. Similarly $\mathbf{M} \times_k \mathbf{1}$ repeats the elements multiple times along the k-th mode, effectively increasing the number of dimensions by one.

For example, the zeroth order term

$$\mathbf{C} = \left[\mathbf{M} \times_1 \frac{1}{\sqrt{m_1}} \mathbf{1}^T \times_2 \frac{1}{\sqrt{m_2}} \mathbf{1}^T \times_3 \frac{1}{\sqrt{m_3}} \mathbf{1}^T\right] \times_1 \frac{1}{\sqrt{m_1}} \mathbf{1} \times_2 \frac{1}{\sqrt{m_2}} \mathbf{1} \times_3 \frac{1}{\sqrt{m_3}} \mathbf{1}$$

can be calculated using

$$\mathbf{C}_{ijk} = \frac{1}{m_1 m_2 m_3} \sum_{k_1,k_2,k_3=1}^{K_1,K_2,K_3} M_{k_1k_2k_3} \text{ for every } i, j, k.$$

Similar simplifications allow the calculation of $\mathbf{S}_\Lambda$ via the following algorithm

Algorithm *BOTIC (fully known)*
INPUT: $\Lambda = (\lambda_1,\ \lambda_2)$, $\mathbf{M}$
OUTPUT: $\mathbf{S}_\Lambda(\mathbf{M}) \coloneqq \mathbf{M}^\Lambda$

```
$c \leftarrow \frac{1}{m_1m_2m_3}\sum_{k_1,k_2,k_3=1}^{K_1,K_2,K_3}\mathbf{M}_{k_1k_2k_3}$    ▷ zeroth order
$\mathbf{M}_{k_1k_2k_3} \leftarrow \mathbf{M}_{k_1k_2k_3} - c$
$\mathbf{b}^1_{k_1} \leftarrow \frac{1}{m_2m_3}\sum_{k_2,k_3=1}^{K_2,K_3}\mathbf{M}_{k_1k_2k_3}$    ▷ first order
$\mathbf{b}^2_{k_2} \leftarrow \frac{1}{m_1m_2}\sum_{k_1,k_2=1}^{K_1,K_2}\mathbf{M}_{k_1k_2k_3}$
$\mathbf{b}^3_{k_3} \leftarrow \frac{1}{m_1m_3}\sum_{k_1,k_3=1}^{K_1,K_3}\mathbf{M}_{k_1k_2k_3}$
$\mathbf{M}_{k_1k_2k_3} \leftarrow \mathbf{M}_{k_1k_2k_3} - \mathbf{b}^1_{k_1} - \mathbf{b}^2_{k_2} - \mathbf{b}^3_{k_3}$
$S^1_{k_1k_2} \leftarrow \frac{1}{m_3}\sum_{k_3=1}^{K_3}\mathbf{M}_{k_1k_2k_3}$    ▷ second order
$S^2_{k_2k_3} \leftarrow \frac{1}{m_1}\sum_{k_1=1}^{K_1}\mathbf{M}_{k_1k_2k_3}$
$S^3_{k_1k_3} \leftarrow \frac{1}{m_2}\sum_{k_2=1}^{K_2}\mathbf{M}_{k_1k_2k_3}$
$\mathbf{M}_{k_1k_2k_3} \leftarrow \mathbf{M}_{k_1k_2k_3} - S^1_{k_1k_2} - S^2_{k_2k_3} - S^3_{k_1k_2}$
$\mathbf{T} \leftarrow \mathbf{M}$    ▷ third order
$\mathbf{M}^\Lambda_{k_1k_2k_3} \leftarrow c + \mathbf{b}^1_{k_1} + \mathbf{b}^2_{k_2} + \mathbf{b}^3_{k_3} + S_{\lambda_1}(S^1)_{k_1k_2} + S_{\lambda_1}(S^2)_{k_1k_2} + S_{\lambda_1}(S^3)_{k_1k_2} + \mathbf{S}_{\lambda_2}(\mathbf{T})_{k_1k_2k_3}$
```

where $S_{\lambda_1}(S^l)$ is the soft-threshhold SVD of S^l as defined in Definition 2.9.

4.3 Regularization

We will now discuss the regularizer $\mathcal{R}$ used in our model (10). This choice heavily influences the optimal solution $\mathbf{S}_\lambda(\mathbf{M})$ to (14) and thus the quality of our results.
As discussed above $\mathbf{S}_{\lambda_{k_1k_2k_3}}(\mathbf{M})$ should reduce to the soft-tresholding SVD if $\mathbf{M}$ corresponds to a term of order 2 or less.
Our goal is to only capture the most important of the purely order 3 phenomena in the data. Hence, a first idea would be to fix the Tucker rank to a low value $[r_1, r_2, r_3]$ (e.g. $[2, 2, 2]$) and to simply set $\mathbf{S}_\lambda(\mathbf{M})$ to the solution of

$$\begin{aligned} \min_{\hat{\mathbf{M}}}\ & \frac{1}{2}\left\|\mathbf{M} - \hat{\mathbf{M}}\right\|_F^2 \\ \text{s.t. } & \operatorname{rank}(\mathbf{P}_k(\hat{\mathbf{M}})) = r_k \text{ for } k \in \{1, 2, 3\}. \end{aligned} \tag{16}$$

This approximate can be found with HOOI (see Section 2.3.2). As can be seen in Chapter 5, this simple approach suffices when $[r_1, r_2, r_3]$ is close to the true Tucker rank of the respective ground truth term $\mathbf{P}^{k_1k_2k_3}(\mathbf{R})$ that has to be approximated. Although, even in that case, it is prone to overfitting, in particular when the respective component

only has a small influence in $\mathbf{R}$.
When $[r_1, r_2, r_3]$ can not be estimatated beforehand this approach quickly becomes useless, under or overfiting dramatically.
One potential way to combat these flaws is to always overestimate the Tucker rank and use a simple additional regularizer like $\hat{\lambda}\left\|\hat{\mathbf{M}}\right\|_F$ to force the third order term to tend to zero. Unfortunatly the solution to (16) stays the same even with this additional penalty term (until $\hat{\lambda}$ gets to big and the all-zero tensor becomes the optimum).

A better way would be to use a regularizer that enables the dynamic selection of the rank best fit to the data. This can be achieved by using any generalization of the nuclear norm. We choose a scaled variant of the overlap nuclear norm $\mathcal{R}(\mathbf{M}) := \sum_{k=1}^{d} \gamma_k \|\mathbf{P}_k(\mathbf{M})\|_*$ since it is relatively easy to compute and forces the order three term to have a low rank in each dimension. With this norm $\mathbf{S}_\lambda(\bar{\mathbf{R}})$ is the optimal solution to

$$\min_{\mathbf{M}} \ \frac{1}{2}\left\|\bar{\mathbf{R}} - \mathbf{M}\right\|_F^2 + \lambda \sum_{k=1}^{d} \gamma_k \|\mathbf{P}_k(\mathbf{M})\|_*. \tag{17}$$

4.4 The Fully Known Case

In this section, the tensor that has to be approximated is denoted with $\bar{\mathbf{R}}$.
With the choosen regularization we can finally discuss how $\mathbf{S}_\lambda(\bar{\mathbf{R}})$ can be computed. Although the overlap nuclear norm is a sum of nuclear norms it can't be solved directly by applying the soft-threshhold SVD because of the high dependence of the summands.

There are however several methods to solve the optimization problem including ([18], [17], [19]). We will use a simpler version of the alogrithm from [18] which uses the *Alternating Direction Method of Multipliers* (ADMM).

4.4.1 ADMM

ADMM can be used to solve problems of the form

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^n, \mathbf{z}\in\mathbb{R}^m} \ & f(\mathbf{x}) + g(\mathbf{x}) \\ \text{s.t. } & A\mathbf{x} = \mathbf{z} \end{aligned} \tag{18}$$

where f and g are convex functions by minimizing the Augemented Lagrangian (AL) of the problem defined as

$$L_\eta(\mathbf{x}, \mathbf{z}, \alpha) = f(\mathbf{x}) + g(\mathbf{x}) + \alpha^T(A\mathbf{x} - \mathbf{z}) + \frac{\eta}{2}\|A\mathbf{x} - \mathbf{z}\|^2.$$

Under some mild conditions (which are fulfilled when f is the quadratic loss [22]) a solution can be found iteratively with the update step

$$\begin{aligned}(\mathbf{x}^{t+1}, \mathbf{z}1t+1) &= \operatorname{argmin}_{\mathbf{x},\mathbf{z}} L_\eta(\mathbf{x}, \mathbf{z}, \alpha^t) \\ \alpha^{t+1} &= \alpha^t + \eta(A\mathbf{x}^{t+1} - \mathbf{z}^{t+1}).\end{aligned}$$

In *ADMM* we alternate between the optimizing for $\mathbf{x}$ and $\mathbf{z}$

$$\begin{aligned}\mathbf{x}^{t+1} &= \operatorname{argmin}_{\mathbf{x}} L_\eta(\mathbf{x}, \mathbf{z}^t, \alpha^t) \\ \mathbf{z}1t+1 &= \operatorname{argmin}_{\mathbf{z}} L_\eta(\mathbf{x}^{t+1}, \mathbf{z}, \alpha^t) \\ \alpha^{t+1} &= \alpha^t + \eta(A\mathbf{x}^{t+1} - \mathbf{z}^{t+1})\end{aligned}$$

which converges to a solution of (18) for every $\eta > 0$. [23]

4.4.2 Tensor Approximation

In this section we will be following the ideas of [18] and adapting them to our setting and notations. Before ADMM can be applied to (17) we first have to introduce several auxiallary tensors $\mathbf{Z}_1, \dots, \mathbf{Z}_d$ and rewrite our problem to

$$\begin{aligned}\min_{\mathbf{M}, \mathbf{Z}_1, \dots, \mathbf{Z}_d} \ & \frac{1}{2}\|\bar{\mathbf{R}} - \mathbf{M}\|_F^2 + \lambda \sum_{k=1}^d \gamma_k \|\mathbf{P}_k(\mathbf{Z}_k)\|_* \\ & \text{s.t. } \mathbf{Z}_i = \mathbf{M} \text{ for } i \in \{1, \dots, d\}\end{aligned}$$

which for $\lambda > 0$ is the same as

$$\begin{aligned}\min_{\mathbf{M}, \mathbf{Z}_1, \dots, \mathbf{Z}_d} \ & \frac{1}{2\lambda}\|\bar{\mathbf{R}} - \mathbf{M}\|_F^2 + \sum_{k=1}^d \gamma_k \|\mathbf{P}_k(\mathbf{Z}_k)\|_* \\ & \text{s.t. } \mathbf{Z}_i = \mathbf{M} \text{ for } i \in \{1, \dots, d\}.\end{aligned} \tag{19}$$

The Augemented Lagrangian is given by

$$L_\eta(\mathbf{M}, \{\mathbf{Z}_k\}_{k=1}^d, \{\mathbf{A}_k\}_{k=1}^d) = \frac{1}{2\lambda}\|\bar{\mathbf{R}} - \mathbf{M}\|_F^2 + \sum_{k=1}^d \gamma_k \|\mathbf{P}_k(\mathbf{Z}_k)\|_* + \sum_{k=1}^d (\langle \eta\mathbf{A}_k, \mathbf{M} - \mathbf{Z}_k\rangle + \frac{\eta}{2}\|\mathbf{M} - \mathbf{Z}_k\|^2) \tag{20}$$

where we also scaled the $\mathbf{A}_k$ by η.
Since L_η is a combination of convex functions it is also convex and we can find the global minima w.r.t $\mathbf{M}$ by finding the critical points of L_η, i.e. setting the respective derivative of L_η to zero:

$$
\begin{aligned}
\frac{\partial L_\eta}{\partial \mathbf{M}} &= \frac{1}{\lambda}(\mathbf{M}-\bar{\mathbf{R}}) + \sum_{k=1}^{d} \eta \mathbf{A}_k + \eta(\mathbf{M}-\mathbf{Z}_k) = 0 \\
&\Leftrightarrow (\frac{1}{\lambda} + \eta d)\mathbf{M} = \frac{1}{\lambda}\bar{\mathbf{R}} - \eta \sum_{k=1}^{d} \mathbf{A}_k + \mathbf{Z}_k \\
&\Leftrightarrow \mathbf{M} = \frac{\bar{\mathbf{R}} + \lambda\eta \sum_{k=1}^{d} \mathbf{Z}_k - \mathbf{A}_k}{1+\lambda\eta d}
\end{aligned}
$$

On the other hand we see that minimizing L_η w.r.t. $\mathbf{Z}_k$ is the same as finding

$$
\begin{aligned}
&\min_{\mathbf{Z}_k} \; -\eta\langle \mathbf{A}_k, \mathbf{Z}_k\rangle + \frac{\eta}{2}\|\mathbf{M}-\mathbf{Z}_k\|^2 + \gamma_k \|\mathbf{P}_k(\mathbf{Z}_k)\|_* \\
=&\min_{\mathbf{Z}_k} \; -\langle \mathbf{A}_k, \mathbf{Z}_k\rangle + \frac{1}{2}\|\mathbf{M}-\mathbf{Z}_k\|^2 + \frac{\gamma_k}{\eta} \|\mathbf{P}_k(\mathbf{Z}_k)\|_* \\
=&\min_{\mathbf{Z}_k} \; \frac{1}{2}\|\mathbf{A}_k\|^2 + \langle \mathbf{A}_k, \mathbf{M}\rangle - \langle \mathbf{A}_k, \mathbf{Z}_k\rangle + \frac{1}{2}\|\mathbf{M}-\mathbf{Z}_k\|^2 + \frac{\gamma_k}{\eta} \|\mathbf{P}_k(\mathbf{Z}_k)\|_* \\
=&\min_{\mathbf{Z}_k} \; \frac{1}{2}\|\mathbf{A}_k\|^2 + \langle \mathbf{A}_k, \mathbf{M}-\mathbf{Z}_k\rangle + \frac{1}{2}\|\mathbf{M}-\mathbf{Z}_k\|^2 + \frac{\gamma_k}{\eta} \|\mathbf{P}_k(\mathbf{Z}_k)\|_* \\
=&\min_{\mathbf{Z}_k} \; \frac{1}{2}\|\mathbf{A}_k + \mathbf{M}-\mathbf{Z}_k\|^2 + \frac{\gamma_k}{\eta} \|\mathbf{P}_k(\mathbf{Z}_k)\|_* \\
=&\min_{\mathbf{Z}_k} \; \frac{1}{2}\|\mathbf{P}_k(\mathbf{A}_k + \mathbf{M}) - \mathbf{P}_k(\mathbf{Z}_k)\|^2 + \frac{\gamma_k}{\eta} \|\mathbf{P}_k(\mathbf{Z}_k)\|_* \\
=&\min_{\mathbf{P}_k(\mathbf{Z}_k)} \; \frac{1}{2}\|\mathbf{P}_k(\mathbf{A}_k + \mathbf{M}) - \mathbf{P}_k(\mathbf{Z}_k)\|^2 + \frac{\gamma_k}{\eta} \|\mathbf{P}_k(\mathbf{Z}_k)\|_*
\end{aligned}
$$

where the unique solution $\mathbf{P}_k(\hat{\mathbf{Z}}_k)$ is given by $S_{\gamma_k/\eta}(\mathbf{P}_k(\mathbf{A}_k + \mathbf{M}))$ (see Section 3.1).
To find the optimal $\hat{\mathbf{Z}}_k$, we have to fold $S_{\gamma_k/\eta}(\mathbf{P}_k(\mathbf{A}_k + \mathbf{M}))$ back to a tensor, reversing the effect of $\mathbf{P}_k$. This reverse folding operation will be denoted by $\mathbf{P}_k^{-1}$.
Note that $\mathbf{M} = \mathbf{P}^{-1}(\mathbf{P}_k(\mathbf{M}))$ and

$$
\text{argmin}_{\mathbf{Z}_k} L_\eta(\mathbf{M}, \{\mathbf{Z}_k\}_{k=1}^d, \{\mathbf{A}_k\}_{k=1}^d) = \mathbf{P}_k^{-1}(S_{\frac{\gamma_k}{\eta}}(\mathbf{P}_k(\mathbf{A}_k + \mathbf{M}))).
$$

Thus we can iteratively repeat the two steps above until $\mathbf{M}^t$ converges
(see [18] for an explanation of the convergence criterium). So finally,

Algorithm *Tensor Aproximation with ADMM*
INPUT: $\bar{\mathbf{R}}$, λ, γ_k for k in $\{1, \ldots, d\}$, ε

```
Initialize $\mathbf{M}, \mathbf{Z}_k, \mathbf{A}_k$ for each $k$
repeat
    $\mathbf{M} \leftarrow \frac{1}{1+\lambda\eta d}(\bar{\mathbf{R}} + \lambda\eta \sum_{k=1}^{d} \mathbf{Z}_k - \mathbf{A}_k)$
    for $k \in \{1, \ldots, d\}$ do
        $\mathbf{Z}_k \leftarrow \mathbf{P}_k^{-1}(S_{\gamma_k/\eta}(\mathbf{P}_k(\mathbf{A}_k + \mathbf{M})))$
        $\mathbf{A}_k \leftarrow \mathbf{A}_k + \eta(\mathbf{M} - \mathbf{Z}_k)$
    end for
until $\|\mathbf{M} - \mathbf{Z}_k\| < \varepsilon$ and $\left\|\frac{1}{\lambda}(\mathbf{M} - \mathbf{R}) + \sum_{k=1}^{d} \mathbf{A}_k\right\| < \varepsilon$
```

where $\mathbf{M}$ converges to a solution of (13) for every $\eta > 0$. η can influence the number of iterations needed, [18] suggests using $\eta = \frac{\eta_0}{std(\mathbf{M})}$. A few experiments show that setting $\eta = 1$ performs similarly well in our context.

4.5 Unbalanced Tensors

The results in Chapter 5 show that with the regularization choosen in Section 4.3, BOTIC can achieve great results on synthetic data $\mathbf{R} \in \mathbb{R}^{m \times m \times m}$.
However, without further precautions, good results are only obtained for balanced data where each dimension/axis of our data has a similiar size, i.e. $\mathbf{R} \in \mathbb{R}^{m_1 \times m_2 \times m_3}$ where $m_1 \approx m_2 \approx m_3$. When working with real world data this assumption is often not fullfilled. For example in the context of recommender systems where $\mathbf{R}_{ijk}$ is the affinity of user i to movie j at time k. In that case we are given have a huge pool of users and movies but are often only interested in the change over bigger time intervals like weeks or months. This results in m_3 being several magnitudes smaller. Generally, when using BOTIC on unbalanced data, the second order terms have varying sizes and thus are likely to have different ranks as well. While the user-item second order term may have a (comparably) high rank (e.g. 50) the user-time term will be of a much lower rank (e.g. 5). A solution would be to use a different regularization parameter for each second order term. Since increasing the number of cross validated parameters also exponentially increases the running time other solutions have to be found.
To mitigate this problem we will instead only use one parameter for second order terms but scale according to the size of the tensor

$$\lambda_{3i} = c_i \lambda_1 = \sqrt{\frac{\min_{j \in \{1,2,3\} \setminus \{i\}} m_j}{\min_{j \in \{1,2,3\}} m_j}} \lambda_1 .$$

Similarly, the scaling factor for the overlap nuclear norm is set to $\gamma_k = \sqrt{m_k}$ to allow the tensor to have a low rank in each mode but also adapt to the size of the tensor.

4.6 Warm Starts and other Heuristics

With many regularization pairs to cross validate, our method has to be evaluated many times and should thus be as optimized as possible. The main contributer to the runtime of our algorithm are the many SVDs that have to be computed in each iteration. For BOTIC we need three soft-threshold SVDs for each second order term and three more soft-threshold SVDs at each iteration of ADMM when approximating the third order term.

This leaves us with two chances for improvement. Either make the SVD faster or use less SVDs overall. To compute the SVD we will use the highly optimized *FORTRAN* routine *ARPACK* [24] that only computes the k largest singular vectors. Thus singular vectors coresponding to small singular values that don't contribute to the soft-threshhold SVD are never computed. We use the soft-threshold SVD computed in the last iteration to choose the value of k in the current iteration. One could potentially use the SVD of the last iterate as a strating point for iterative improvements, however since *ARPACK* usually converges in only very few iterations anyway the benefits would only be small.

With a fast SVD the next step is to reduce the number of SVDs that have to be computed. This can be done by reducing the number of iterations BOTIC needs until it converges. Since the number of iterations for each $\Lambda \in \mathcal{V}$ strongly depends on $\left\|\mathbf{M}^0 - \mathbf{M}^\infty\right\|_F^2$ (Proposition 4.2), choosing a good estimate $\mathbf{M}^0$ of the solution $\mathbf{M}^\infty$ can speed up the convergence significantly.

We will now discuss several possibilities of these so called warm starts and evaluate their performance. $\mathcal{V} = \mathcal{V}^{(1)} \times \mathcal{V}^{(2)}$ is assumed where $\lambda_1 \in \mathcal{V}^{(1)}$ is the second order regularization and $\lambda_2 \in \mathcal{V}^{(2)}$ the third order regularization. As a baseline the warm starting method based on the OMIC framework will be used **[omic]**. As detailed in Section 4.2, a series of intermediate results are computed for $\bar{\Lambda} = p_{k_1,k_2,k_3}(\lambda), \lambda \in \mathcal{V}_{k_1,k_2,k_3}$ such that $\sum_{k_1,k_2,k_3=1}^{K_1,K_2,K_3} \mathbf{M}_{(k_1,k_2,k_3)}^{\Lambda_{k_1k_2k_3}}$ can be used as a warm start for Λ.

An other approach would be to use already computed $\mathbf{M}^\Lambda$ as a warm start for similar $\hat{\Lambda}$. We propose several warm starting schemes:

last:

for $(\mathcal{V}_i^{(1)}, \mathcal{V}_j^{(2)}), j > 1$ use $M^{(\mathcal{V}_i^{(1)}, \mathcal{V}_{(j-1)}^{(2)})}$

last+:

for $(\mathcal{V}_i^{(1)}, \mathcal{V}_j^{(2)}), j > 1$ use $M^{(\mathcal{V}_i^{(1)}, \mathcal{V}_{(j-1)}^{(2)})}$,
for $(\mathcal{V}_i^{(1)}, \mathcal{V}_j^{(2)}), j = 1, i > 1$ use $M^{(\mathcal{V}_{(i-1)}^{(1)}, \mathcal{V}_j^{(2)})}$

last mean:

for $(\mathcal{V}_i^{(1)}, \mathcal{V}_j^{(2)}), i > 1, j > 1$ use $\frac{1}{2}(M^{(\mathcal{V}_i^{(1)}, \mathcal{V}_{(j-1)}^{(2)})} + M^{(\mathcal{V}_{(i-1)}^{(1)}, \mathcal{V}_j^{(2)})})$,
for $(\mathcal{V}_i^{(1)}, \mathcal{V}_j^{(2)}), i = 1, j > 1$ use $M^{(\mathcal{V}_i^{(1)}, \mathcal{V}_{(j-1))}^{(2)}}$,
for $(\mathcal{V}_i^{(1)}, \mathcal{V}_j^{(2)}), i > 1, j = 1$ use $M^{(\mathcal{V}_{(i-1)}^{(1)}, \mathcal{V}_j^{(2)})}$

An evaluation of warm starting methods can be found in table 1. BOTIC was evaluated on synthetic data (described in 5.2) of the size $10 \times 10 \times 10$ with 10% observed entries. $\mathcal{V}^{(1)}$ and $\mathcal{V}^{(2)}$ were equidistantly sampled in the log domain of real intervals.

	omic	**last**	**last+**	**last mean**	**last+** without heuristic
SVD	43.424.172	11.261.118	**7.660.158**	31.615.980	25.852.464
BOTIC	699.152	170.623	116.063	479.030	**115.099**
ADMM	672.604	170.628	**116.063**	479.030	668.309
time	2830 s	706 s	**464** s	2041 s	1604 s

Table 1: Several warm starting methods and the number of ARPACK(SVD) iterations, ADMM iterations and BOTIC iterations needed as well as the total computation time. All but the last method use the described heuristic, i.e. only evaluate one ADMM iteration per BOTIC iteration.

Additionally, we can use the heuristic that the first few iterations of BOTIC don't need high precision results since the iterates change quite a bit from iteration to iteration anyway. As long as the direction is approximately correct, we will eventually start converging and can start using a higher precision to guarantee convergence.

This heuristic can be applied to the overlap nuclear norm regularization of the third order terms. Instead of waiting for ADMM to converge we can only computed very few ADMM iterations at each iteration of BOTIC. The state of the ADMM from this approximate result is saved and used as a starting point for the next few ADMM iterations. This also means that the third order terms are not very accurate at first but will get better and better at each step until ADMM converges in the few iterations it is given.

The best results were obtained when only one ADMM iteration is performed in each iteration of BOTIC. In this case, only six SVDs are computed in each iteration of BOTIC.

5 Experiments

In this chapter several experiments are designed to test the capabilities of *BOTIC*. The synthetic is introduced, followed by a description of the evaluation metrics and the baselines we used to compare the obtained results. Next, the experimental setup is detailed. The results are described and discussed thoroughly. The final evaluation of BOMIC will then be given in the next chapter.

5.1 Baselines and Metrics

For the synthetic data experiments we compared the performance of BOTIC (and the BOTIC variant with fixed Tucker ranks described in Section 4.3) against several tensor-based and matrix-based completion methods. As mentioned in Chapter 3, we can use an Imputation procedure (similar to Soft-Impute in Section 3.1) to find the best rank-r approximation and the best rank-$[r, r, r]$ approximation even with missing data. The best rank-r approximation was found with *Alternating Least Squares* (ALS) [25] and the best rank-$[r, r, r]$ was found with HOOI (see Section 2.3.2). Additionally, we used cross validation to dynamically choose a Tucker rank from a restricted set of possibilities (HOOI-sel). optTR [15] is used as a representative for methods that are not build on the idea of Imputation. Instead a gradient-based approach was used. Finally, in avg-SI, Soft-Impute was applied to 3 matrices each obtained by averaging over one mode of the tensor. After these matrices have been completed the three matrices were recombined to form a tensor. To sum up, we investigated:

- **[BOTIC]** with overlap nuclear norm regularization,
- **[BOTIC-TR]** with third order terms fixed at a Tucker rank of $[2, 2, 2]$,
- **[CP-ALS-r]** Imputation-based, best r-rank,
- **[HOOI-r]** Imputation-based, best $[r, r, r]$-rank,
- **[HOOI-sel]** Imputation-based, best rank-$[r, r, r]$ with $r \in \{3, \ldots, 7\}$,
- **[optTR]** Gradient-based, low Tucker rank,
- **[avg-SI]** Soft-Impute on the averaged tensor.

Internal regularization parameters are searched for with 5 fold cross validation, other parameters (e.g. which n-ranks to use for HOOI-sel) with 10 fold cross validation. All methods are given 5000 iterations per set of investigated parameters. If no parameters are to be selected the method is given 5000 iterations in total. The optimization tolerance is set to $\varepsilon = 10^{-4}$. Second order regularizations (BOTIC, BOTIC-TR, avg-SI) are selected from 8 samples in the log domain of $[0.001, 2]$. Third order regularizations (BOTIC) are selected from 8 samples in the log domain of $[0.01, 5]$.

We will now discuss the metrics used to evaluate and compare the results from our methods. Since the goal of the OTIC framework is to minimize w.r.t to the Frobenius norm, the most important metric to evaluate the quality of a tensor completion is the *root mean square error* (RMSE).

Definition 5.1.
The RMSE between two tensors $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{m_1 \times \cdots \times m_d}$
on a set of entries $\Omega \subset \{1, \ldots, m_1\} \times \cdots \times \{1, \ldots, m_d\}$ *is definied as*

$$RMSE(\boldsymbol{A}, \boldsymbol{B}, \Omega) := \sqrt{\frac{1}{|\Omega|} \|\boldsymbol{P}_\Omega \boldsymbol{A} - \boldsymbol{P}_\Omega \boldsymbol{B}\|_F^2}.$$

If the ground truth is known we can decompose the RMSE into several orthogonal components of the OTIC model. We can then analyze how well sepecific terms are completed. The resulting metric is called *tensor bias deviation* (TBD).

Definition 5.2.
The TBD between two tensors $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{m_1 \times \cdots \times m_d}$ *w.r.t.* $\mathcal{S}_{k_1,k_2,k_3}$ *is definied as*

$$TBD_{k_1 k_2 k_3}(\boldsymbol{A}, \boldsymbol{B}) := \sqrt{\frac{1}{m_1 m_2 \cdots m_d} \left\|\boldsymbol{\Pi}^{k_1 k_2 k_3}(\boldsymbol{A}) - \boldsymbol{\Pi}^{k_1 k_2 k_3}(\boldsymbol{B})\right\|_F^2}.$$

Note that under the BOTIC model TBD_{111} meassures the deviation in the mean of the tensors, TBD_{211}, TBD_{121}, TBD_{112} the deviation of the specific biases, TBD_{122}, TBD_{212}, TBD_{221} the RMSE of the second order terms and TBD_{222} the RMSE on the third order residuals. Using TBD we can explain the error meassured by the RMSE of three dimensional tensors $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m_1 \times m_2 \times m_3}$ more precisely:

$$\begin{aligned} RMSE(\mathbf{A}, \mathbf{B})^2 &= \frac{1}{m_1 m_2 m_3} \|\mathbf{A} - \mathbf{B}\|_F^2 \\ &= \frac{1}{m_1 m_2 m_3} \left\| \sum_{k_1,k_2,k_3=1}^{2,2,2} \mathbf{\Pi}^{k_1 k_2 k_3}(\mathbf{A}) - \sum_{k_1,k_2,k_3=1}^{2,2,2} \mathbf{\Pi}^{k_1 k_2 k_3}(\mathbf{A}) \right\|_F^2 \\ &= \sum_{k_1,k_2,k_3=1}^{2,2,2} TBD_{k_1 k_2 k_3}(\mathbf{A}, \mathbf{B})^2 \end{aligned}$$

Additionally, we define $B0 := TBD_{111}$, $B1 := \sqrt{TBD_{211}^2 + TBD_{121}^2 + TBD_{112}^2}$,
$B2 := \sqrt{TBD_{122}^2 + TBD_{212}^2 + TBD_{221}^2}$ and $B4 := TBD_{222}$,
comprehensively contributing the error to zeroth, first, second and third order terms:

$$RMSE(\mathbf{A}, \mathbf{B})^2 = B0(\mathbf{A}, \mathbf{B})^2 + B1(\mathbf{A}, \mathbf{B})^2 + B2(\mathbf{A}, \mathbf{B})^2 + B3(\mathbf{A}, \mathbf{B})^2$$

Finally we define the *Spearman Correlation* (SPC), the corellation between the scores of two ordered sets.

Definition 5.3.
The SPC between two tensors $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{m_1 \times \cdots \times m_d}$ is defined as

$$SPC(\boldsymbol{A}, \boldsymbol{B}) := \rho_{rg_{\boldsymbol{A}}, rg_{\boldsymbol{B}}} = \frac{cov(rg_{\boldsymbol{A}}, rg_{\boldsymbol{B}})}{std(rg_{\boldsymbol{A}})std(rg_{\boldsymbol{B}})}$$

where $rg_{\boldsymbol{M}}$ are the scores of $\boldsymbol{M}$, i.e. $\boldsymbol{M}_{i_1 i_2 \ldots i_d}$ is the $(rg_{\boldsymbol{M}})_{i_1 i_2 \ldots i_d}$-th biggest value of $\boldsymbol{M}$.

5.2 Synthetic Data Experiments

The goal of this set of experiments is to design synthetic data that allow us to control how much of the data can be attributed to first, second and third order interactions. We then investigated the influence of the portion of observed entires and the influence of second and third order terms on the quality of the prediction.

5.2.1 Generation Procedure

The data consists of three orthogonal summands, $\mathbf{T}^{(1)}, \mathbf{T}^{(2)}, \mathbf{T}^{(3)} \in \mathbb{R}^{m_1 \times m_2 \times m_3}$.
Every summand is normalized to have the same Frobenius norm as a tensor which has ones everywhere, i.e. $\left\| \mathbf{T}^{(l)} \right\|_F = \sqrt{m_1 m_2 m_3}$.
The first term $\mathbf{T}^{(1)}$ consists of a sum of pure (first order) biases, i.e.
$\mathbf{T}^{(1)}_{k_1,k_2,k_3} = \mathbf{b}^{(1)}_{k_1} + \mathbf{b}^{(2)}_{k_2} + \mathbf{b}^{(3)}_{k_3}$ where $\mathbf{b}^{(l)}$ are centered normalized random gaussian vectors.
The second summand $\mathbf{T}^{(2)}$ consists of a sum of purely second order terms that caputure the interactions of pairs of items, free from any item specific bias. These effects can be represented with low rank matrices, i.e. $\mathbf{T}^{(2)}_{k_1,k_2,k_3} = S^{(1)}_{k_1,k_2} + S^{(2)}_{k_2,k_3} + S^{(3)}_{k_1,k_3}$ where $S^{(l)}$ has a (low) fixed rank r_1 and is free of order one phenomena ($S^{(l)}_{i,\cdot}$ and $S^{(l)}_{\cdot,j}$ sum up to zero).
More precisely,

$$S^{(l)} = \sum_{i=1}^{r_1} \mathbf{u}^{(l,i)} (\mathbf{u}^{(l,i)})^T$$

$$\text{where } \mathbf{u}^{(l,i)} = \mathbf{v}^{(l,i)} - \sum_{j=0}^{i-1} \frac{(\mathbf{v}^{(l,i)})^T (\mathbf{v}^{(l,j)})}{\left\| \mathbf{v}^{(l,j)} \right\|_2^2} \mathbf{v}^{(l,j)}$$

where $\mathbf{v}^{(l,j)}$ are again centered normalized random gaussian vectors.
$\mathbf{T}^{(3)}$ is a purely third order term. This tensor has a (low) fixed Tucker rank $[r_2, r_2, r_2]$ and is free of lower biases, i.e. $\mathbf{T}^{(1)}_{\cdot,k_2,k_3}$, $\mathbf{T}^{(1)}_{k_1,\cdot,k_3}$ and $\mathbf{T}^{(1)}_{k_1,k_2,\cdot}$ sum up to zero.
More preciesly,

$$S^{(l)} = \sum_{i=1}^{r_2} \mathbf{w}^{(1,i)} \circ \mathbf{w}^{(2,i)} \circ \mathbf{w}^{(3,i)}$$

$$\text{where } \mathbf{w}^{(l,i)} = \mathbf{x}^{(l,i)} - \sum_{j=0}^{i-1} \frac{(\mathbf{x}^{(l,i)})^T \mathbf{x}^{(l,j)}}{\left\| \mathbf{x}^{(l,j)} \right\|_2^2} \mathbf{x}^{(l,j)}.$$

We combine the three tensors with a weighted sum so we can control the influence of each factor,

$$\mathbf{T} = \frac{\mathbf{T}^{(1)} + \alpha \mathbf{T}^{(2)} + \beta \mathbf{T}^{(3)}}{\sqrt{1 + \alpha^2 + \beta^2}}$$

where α controls the influence of second order terms and β controls to influence of third order terms. The resulting data has norm $\|\mathbf{T}\|_F = \sqrt{m_1 m_2 m_3}$ and a Tucker rank of $[r, r, r]$ where $r = 1 + 1 + 2r_1 + r_2$.

5.2.2 Results

We ran a set of experiments on the synthetic data described in the previous section investigating the capabilities of BOTIC and the baselines under changing conditions. The second order terms were set to be of rank 2 ($r_1 = 2$) and the third order term had Tucker rank $[1, 1, 1]$ ($r_2 = 1$). For $\alpha > 0$, $\beta > 0$, the data thus has a Tucker rank of $[7, 7, 7]$ and a rank of $\approx$ 11-13 (shown by numerical experiments).
The plotted results can be found in the Appendix (Section 7.1). All plots show the RMSE, the SPC, B0, B1, B2 and B3. The graphs are smoothed out with a moving average by replacing each point with the mean of the point itself and its direct neighbors. Note that for CP-ALS-r, $r > 9$ no useful predictions could be made because the Imputation did not converge.

Percentage of Observed Entries

First, we will investigate how the methods perform with different percentages of observed data. Synthetic data is generated for 20 different percentages p_i which are sampled equidistantly in the log domain of $[0.5\%, 50\%]$. For each p_i, 5 tensors are generated and $(1 - p_i)$ % of the values are hidden from the methods. All tensors have an equal portion of first, second and third order terms, i.e. $\alpha = \beta = 1$. Each method is started on each tensor and the quality of the result is measured with the metrics described in Section 5.1. Figure 1 and Figure 2 show the performance of HOOI-r and CP-ALS-r for different values of r while in Figure 3 a complete overview of all methods is given. In all figures the percentage of observed entries (p) is plotted against RMSE, B0, B1, B2, B3 and SPC. For BOTIC we excpect low values of RMSE, B0, B1, B2 and B3 as well as a SPC close to 1, indicating a small prediction error.
As to be excpected, when p gets smaller all methods perform worse until they approach a RMSE of 1 where they become useless, not finding any of the patterns in the data (The all-zero tensor has a RMSE of 1 on the synthetic data).
Both HOOI-r and CP-ALS-r perform increasingly well with r increasingly closer to $r = 8$ because when using a rank that is too low the data can't be represented well and the methods will underfit. When the ranks are choosen too high both methods are prone to overfitting, finding patterns in the observed values that are not present in the ground truth.

Note that in Figure 1, in particular on RMSE and B3, when the percentage of observed entries is low enough the optimal performance was not always achieved by choosing the true ranks of the ground truth. Instead smaller ranks gain an advantage because of the lower number of values that have to be fitted. Around $p_i = 5\%$ HOOI-4 performed best. Generally CP and Tucker based methods show very similiar performance.

The gradient-based optTR can reconstruct the ground truth near pearfectly for high p but is worse than the other methods for $p < 15\%$ and 'useless' for $p < 10\%$.

BOITC and BOTIC-TR generally perform better than all other methods, especially for small p. The RMSE in Figure 3 is smaller then 1 at all times and only really high for $p < 5\%$ ($p < 2\%$ for BOTIC-TR).

When comparing BOTIC and BOTIC-TR we first note that for high p ($> 11\%$) BOTIC-TR is the only method that is not able to score a RMSE near 0. As discussed in 4.3 this is likely due to overfitting of the third order term. Still BOTIC-TR outperforms BOTIC for $3\% \leq p \leq 10\%$. This can be explained by the additional information that BOTIC-TR is given about the data. BOTIC only searches for a rank $[2, 2, 2]$ third order term, not considering all possibilities. While this is definitely a great disadvantage for higher values of p, when only a few entires are observed this limits the overfitting that can occur.

Second Order Terms

Next, we will investigate how the methods perform with an increasing influence of second order terms in the observed data. We will steadily increase α, increasing the proportion of second order term in the data.

Synthetic data is generated for 20 different percentages of α_i which are sampled equidistantly from $[0, 5]$. For each α_i, 10 tensors are generated. Five tensors have $p = 5\%$ while the other five have $p = 10\%$. Additionally, we set $\beta = 0$, i.e. don't allow any order three interactions in the data.

Figure 4 and Figure 5 show the performance of HOOI-r and CP-ALS-r for different values of r and $p = 5\%$. Figure 6 and Figure 7 show an overview of selected methods for $p = 5\%$ and $p = 10\%$ respectively. optTR is excluded from the graph for $p = 5\%$ due to its terrible performance.

Similar to the first set of experiments, HOOI-r and CP-ALS-r show the best performance when the estimated ranks are close to the ranks of the ground truth.

In Figure 6 we can see the superiority of BOTIC over all other methods when $p = 5\%$, i.e. when the percentage of observed entries is low. While HOOI-r and CP-ALS-r show high variance in the quality of the approximation, BOTIC has a consistent RMSE smaller than 0.1 and a perfect SPC very close to 1 outperforming any other method despite not having any rank information about the ground truth. Although BOTIC-TR and avg-SI are more consistent than HOOI-r and CP-ALS-r, Figure 6 shows that their RMSE grows as the influence of second order terms increases, surpassing both HOOI and CP-ALS based methods. Note that HOOI-sel, which used less information about the ground truth data performs considerably worse than HOOI-6 which knew the exact Tucker rank

of the data. Figure 6 also shows that the matrix based avg-SI has problems in finding the first and second oder bias terms making it the worst performing method after optTR, despite its perfect B3 score.
When the percentage of observed entries is increased to $p = 10\%$, optTR and HOOI-sel can close the gap reaching similar near-perfect scores on all metrics. This can be seen in Figure 7. BOTIC-TR doesn't reach an RMSE below 0.1, again indicating an overfitting due to the overestimation of third order ranks.

Third Order Terms

Finally, we will investigate how the methods perform with an increasing influence of third order terms in the observed data. We will steadily increase β, increasing the proportion of third order term in the data.
Most of the experimental setup is the same as in the above paragraph, this time sampling 20 values β_i from $[0, 5]$ and setting $\alpha = 0$.

Again, Figure 8 and Figure 9 show the performance of HOOI-r and CP-ALS-r for different values of r and $p = 5\%$. Figure 10 and Figure 11] show an overview of selected methods for $p = 5\%$ and $p = 10\%$ respectively. avg-SI is excluded from the graph for $p = 10\%$ due to its (expected) terrible performance (Because avg-SI is based on the avergages of the tensor, it can't find any order three interactions).
Similar to the other sets of experiments, HOOI-r and CP-ALS-r show the best performance when the estimated ranks are close to the ranks of the ground truth. Note that unlike in the previous paragraph, BOTIC is not achieving near-perfect results for $p = 5\%$. Instead, Figure 10 shows that the methods based on CP-ALS and HOOI perform best. BOTIC-TR has the best RMSE and is the only method that has a consistent SPC of 1 even as the influence of third order terms incerases. This can be attributed to the fact that BOTIC-TR is using a very accurate estimation of the true ranks of the tensor. Even tough BOTIC has a relatively high RMSE it also shows good results on B0 and a SPC score which is not that far of from 1. This indicates that BOTIC can find many relevant patterns in the data without any known rank-information.
When the percentage of observed entries is increased to $p = 10\%$, BOTIC shows an increased relative performance as well, beating all methods except optTR. Figure 7 shows that it is the only method that detects that the data doesn't contain any second order interactions, resulting in a perfect B2 score. Most notably, BOTIC-TR is outperformed by all other methods demonstrating that the disadvantage of its inflexibility becomes more notable with higher percentage of observed entries.

6 Conclusion

In this thesis we developed the flexible and interpretable OTIC framework which allows the completion of tensors with missing entries. We examined the special case BOTIC and compared its performance against other baselines. The results of the synthetic data experiments in the previous chapter show that BOTIC can dynamically find the most important patterns in the data even when the percentage of observed entries is so low that the other baselines could not be used in a meaningful way, this is particularly true when the second order terms have the biggest influence. We note that although the performance of BOTIC on tensors consisting of mainly third-order terms is great it can't match the performance of some of the other methods which use an good estimate of the rank structure of the ground truth. However, BOTIC doesn't need any rank information about the ground truth. This means that it has a big advantage in real world applications where the true ranks can't be estimated since the performance of methods like HOOI-r and CP-ALS-r whose performance strongly depends on the quality of the rank estimste. Evaluating BOTIC on a large real-world dataset requires some ingenious tricks in the implementation and paralelisation which will be left to future work. We will conclude this thesis by noting that other methods could be build on top of the developed foundations. For example, the OTIC framework has the potential to also include side information which could further improve the performance, especially in real world applications.

7 Appendix

7.1 Figures

All plots and other visualisations from Chapter 5 as discussed in Section 5.2.2. Starting with the results under changing percentage of observed entries, followed by the results under increasing influence of second order terms. Finally, the results of a changing importance of third order terms are presented.

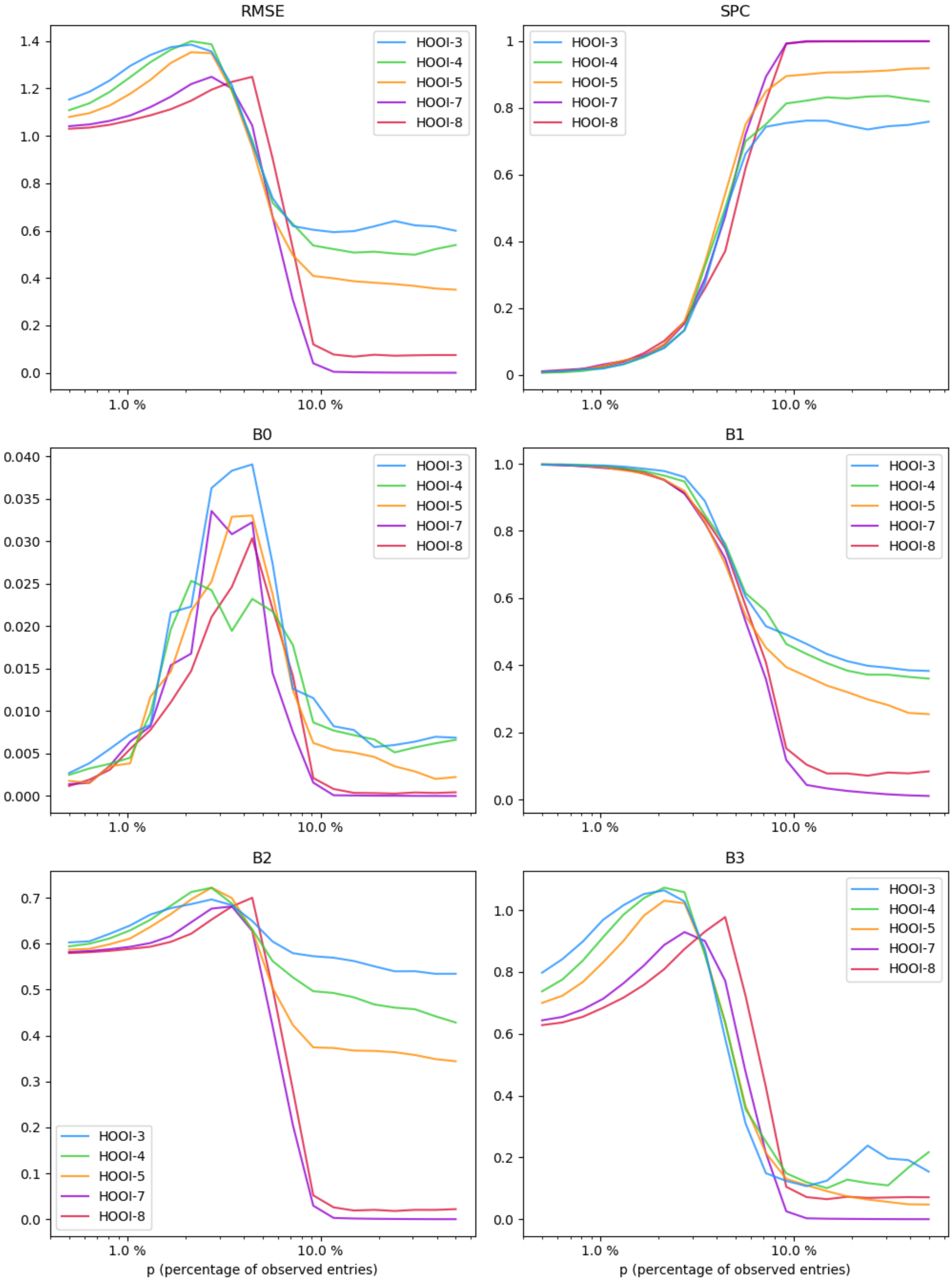

Figure 1: Tucker Decomposition based Imputation using HOOI for different n-ranks. Choosing the true n-ranks ($r = 7$) performs best for high p, while underestimating works even better for small p.

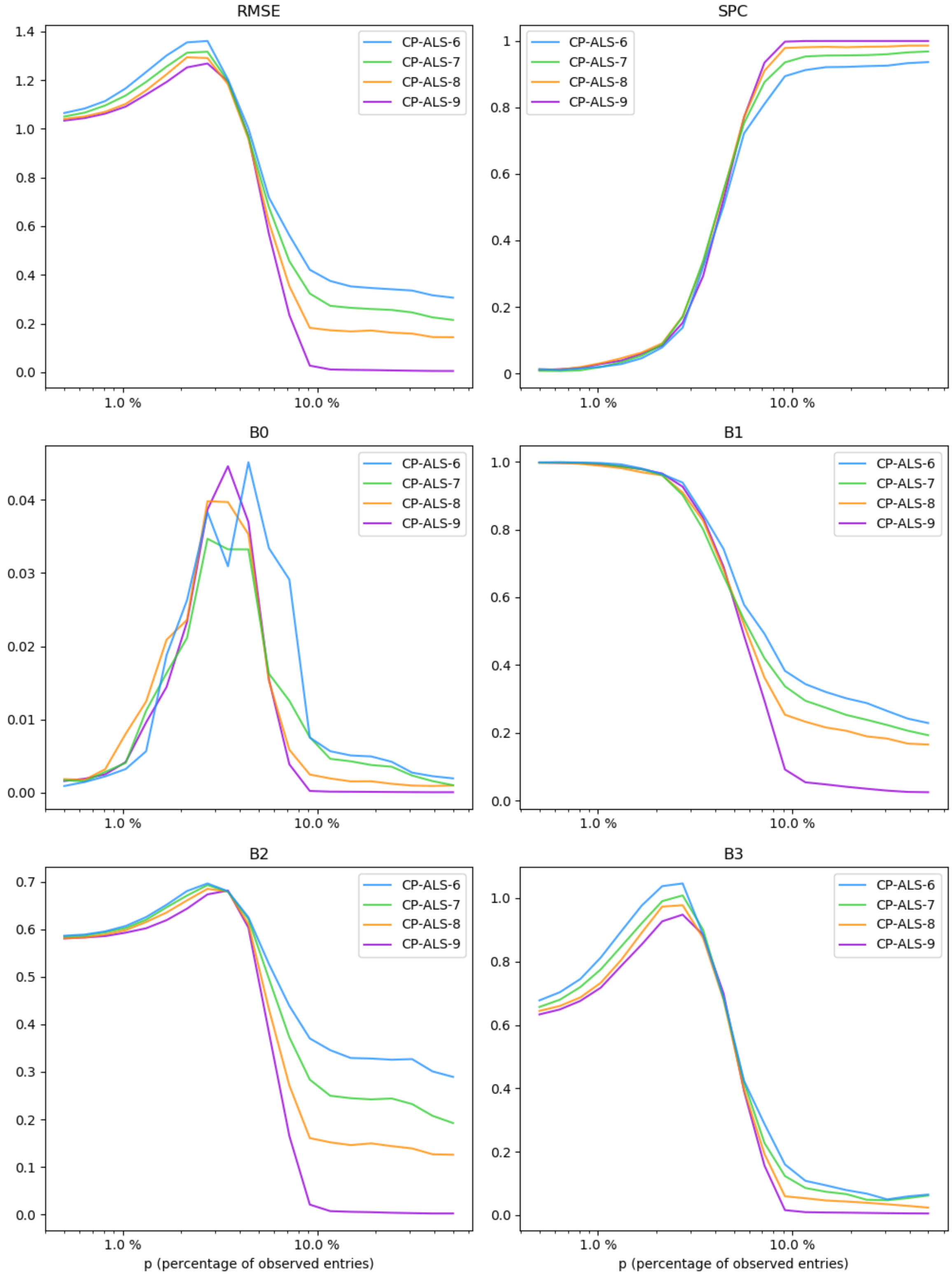

Figure 2: CP Decomposition based Imputation using ALS for different ranks. Choosing a rank close to the true rank performs best.

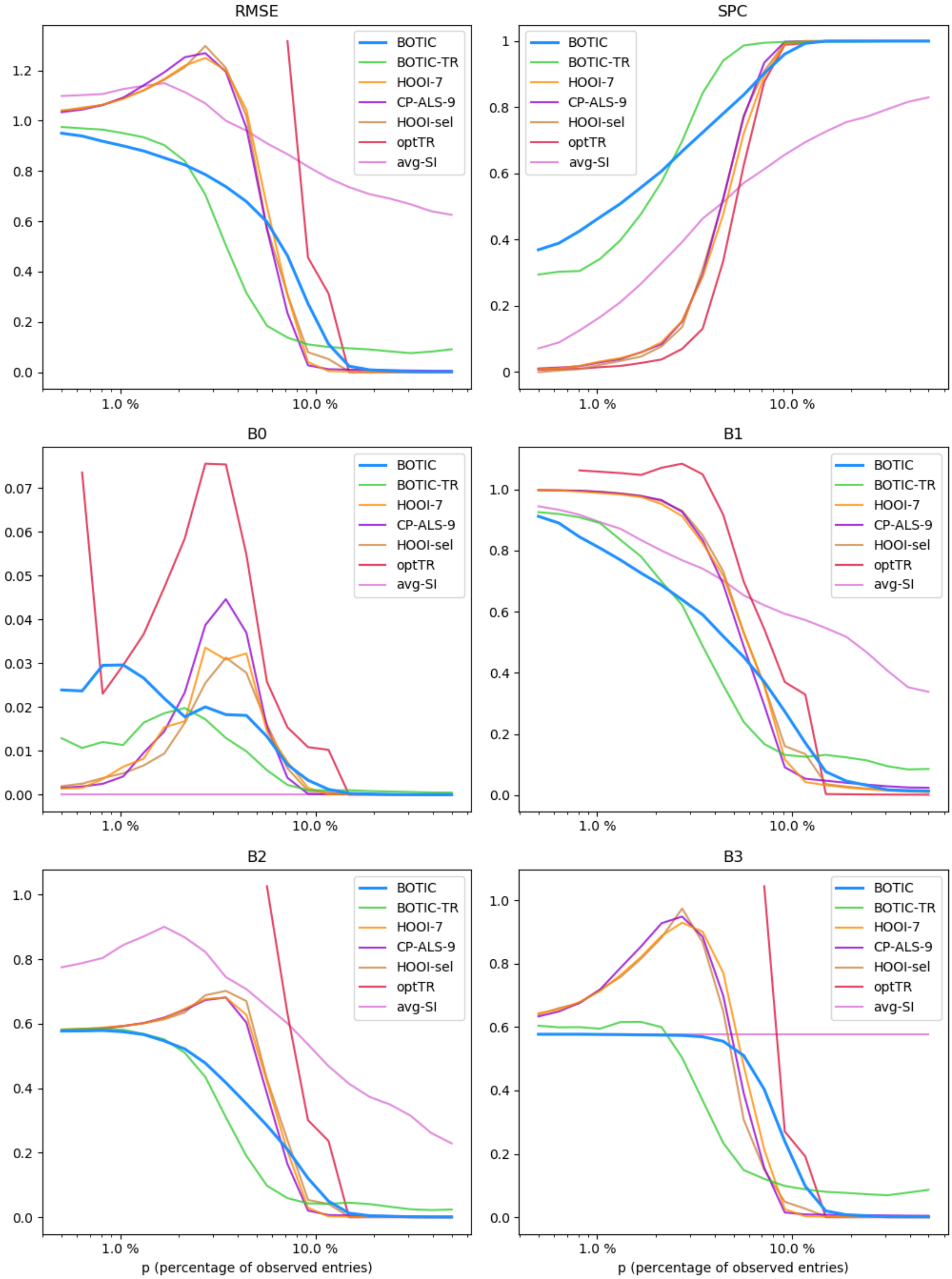

Figure 3: All methods under a changing percentage of observed entries. Note that not all points can be seen for optTR.

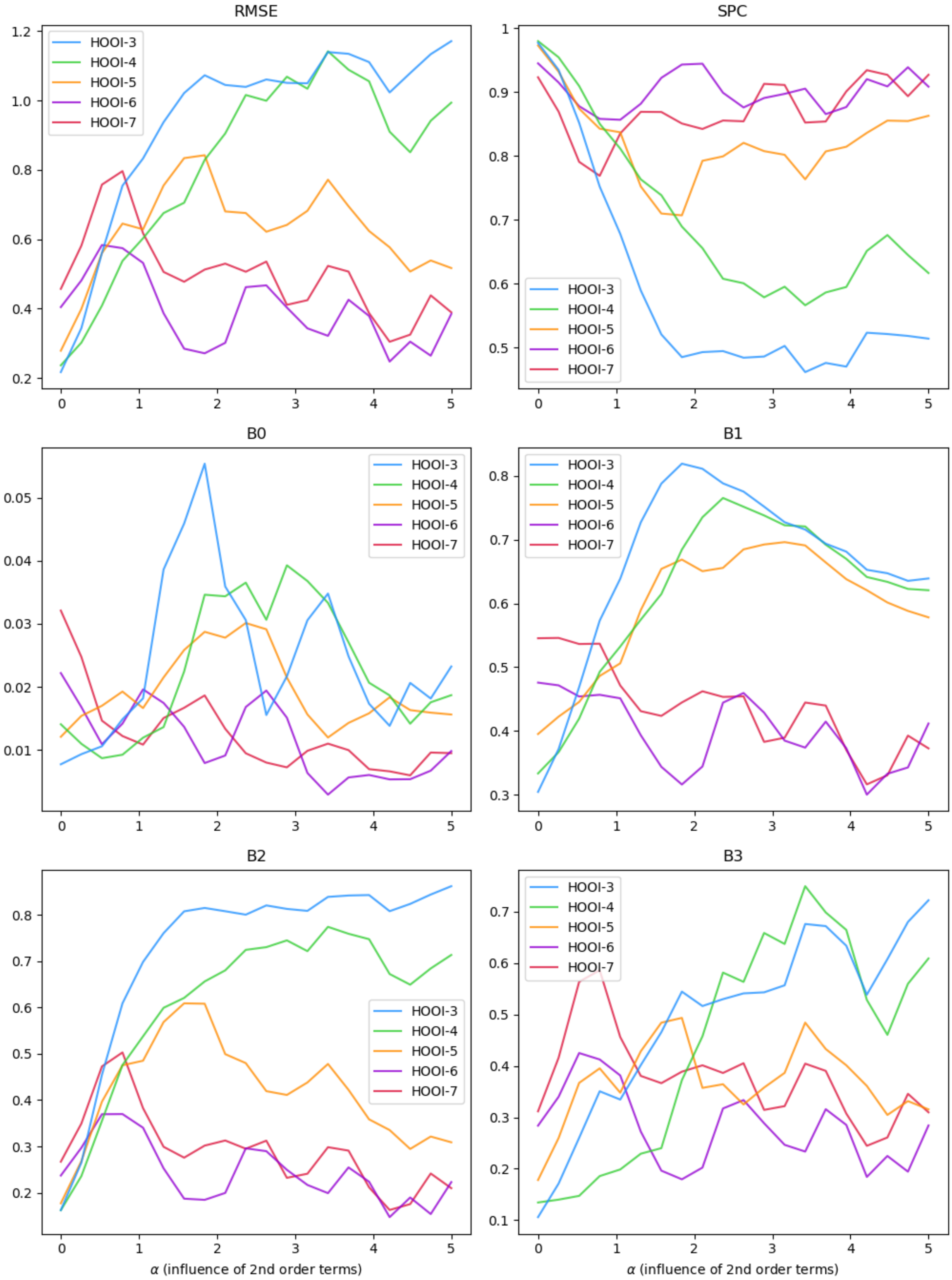

Figure 4: Tucker Decomposition based Imputation using HOOI for different n-ranks, p=5%. Choosing the true n-ranks ($r = 6$) performs best.

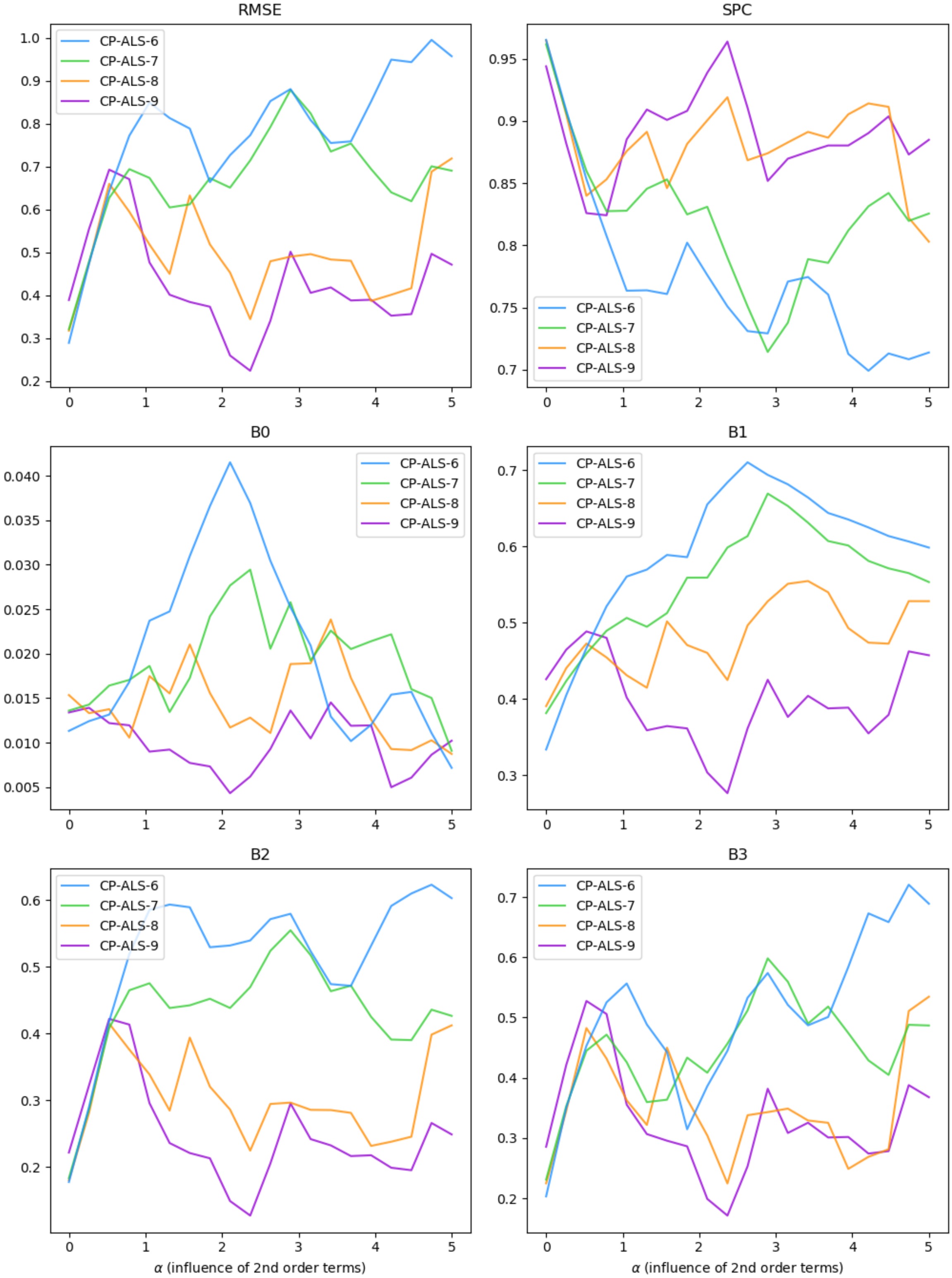

Figure 5: CP Decomposition based Imputation using ALS for different ranks, p=5%. Choosing a rank close to the true rank performs best.

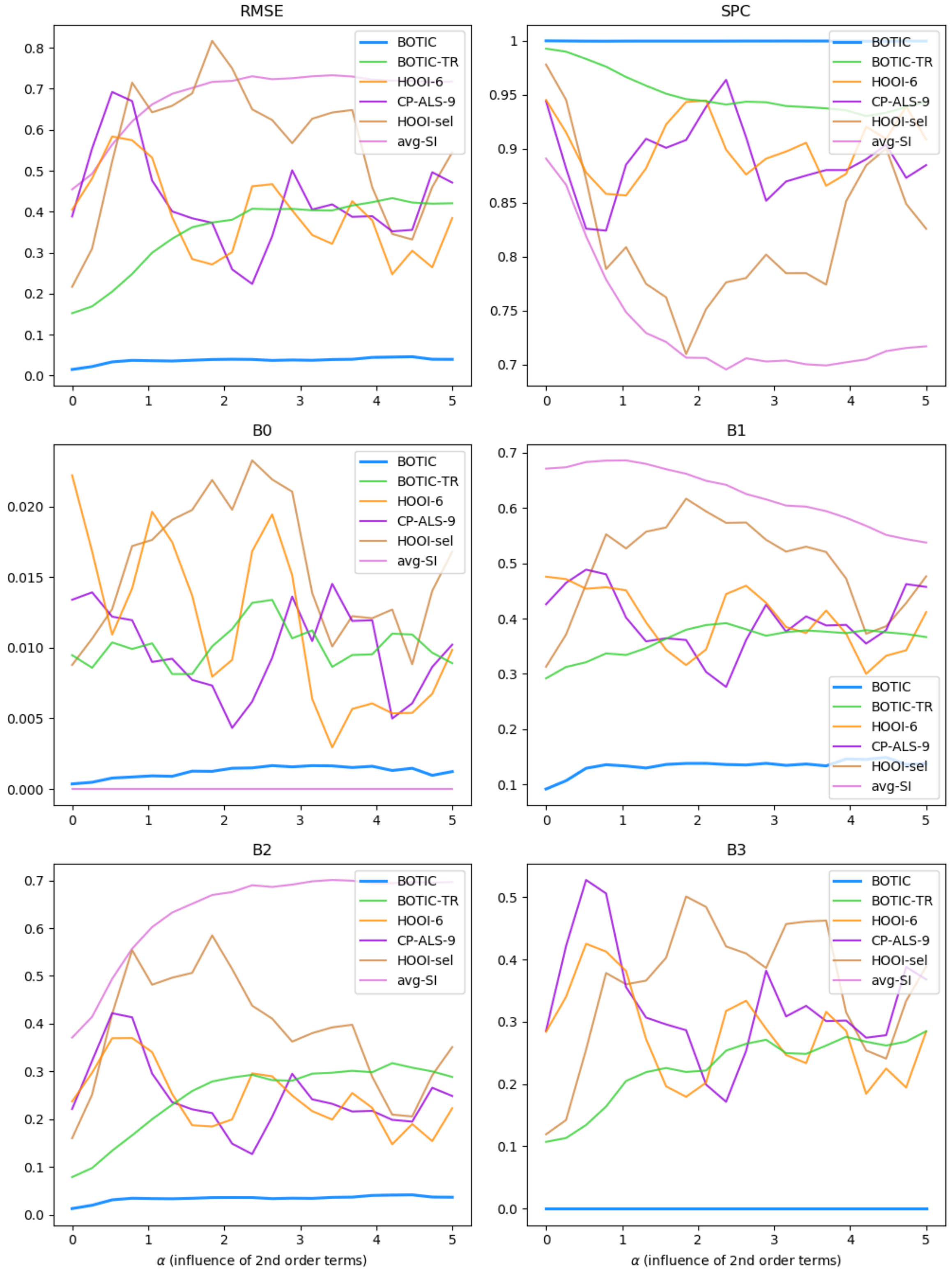

Figure 6: A selection of methods under an increasing relevance of second order terms, p=5%.
Note that optTR is excluded due to poor performance and avg-SI has a consistent B3 of zero.

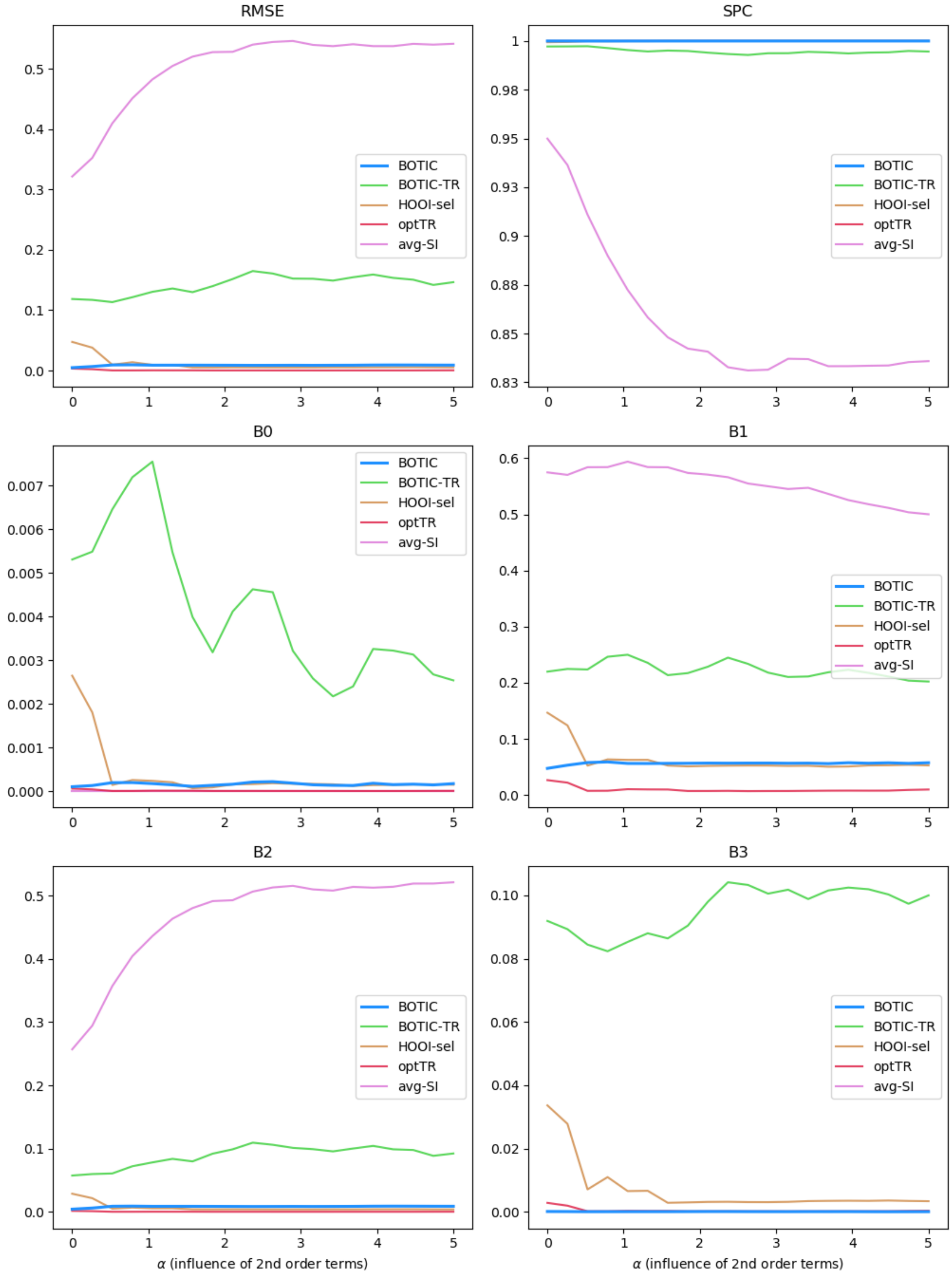

Figure 7: A selection of methods under an increasing relevance of second order terms, p=10%.

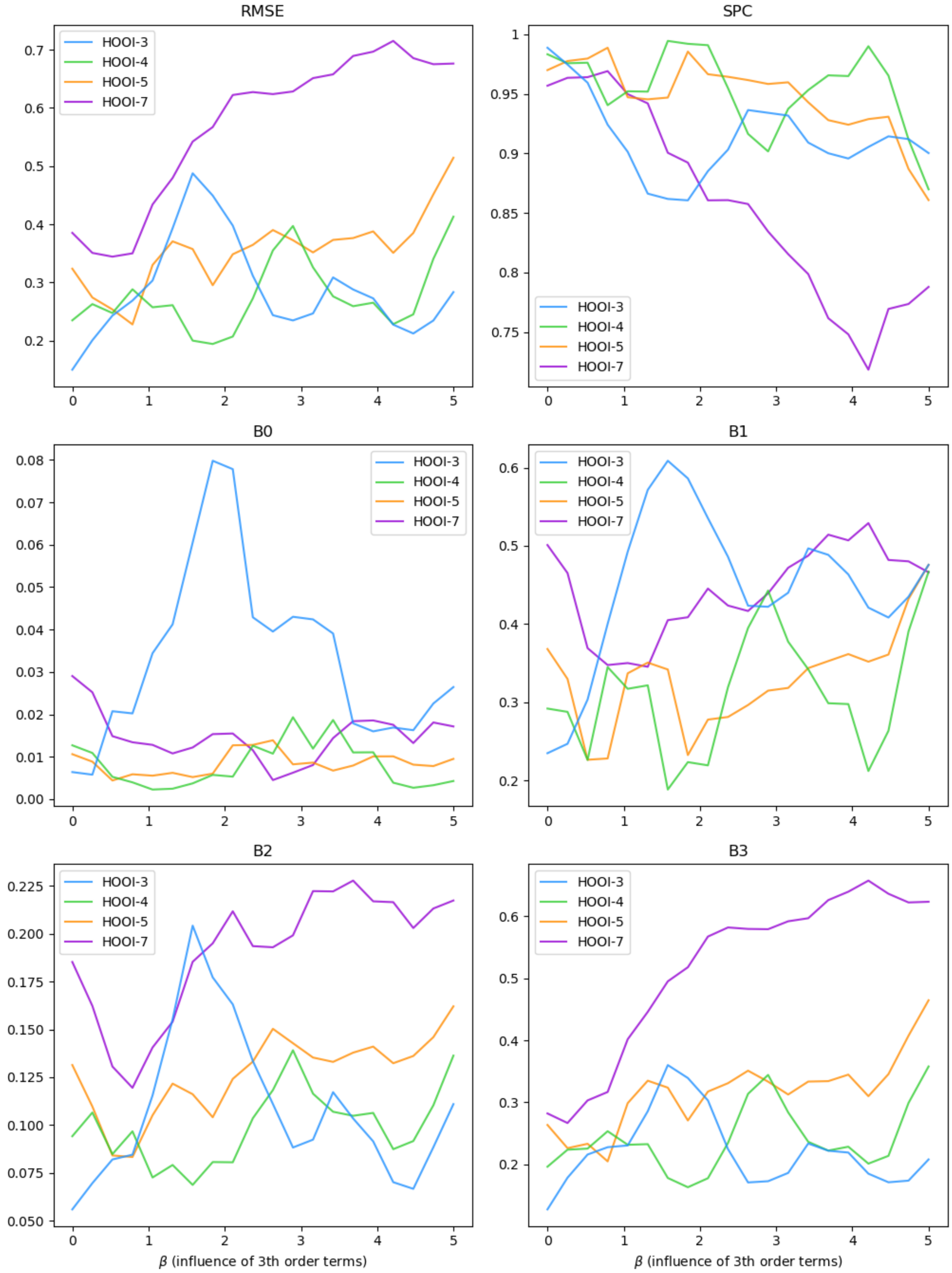

Figure 8: Tucker Decomposition based Imputation using HOOI for different n-ranks, p=5%. Choosing n-ranks close to the true n-ranks performs best.

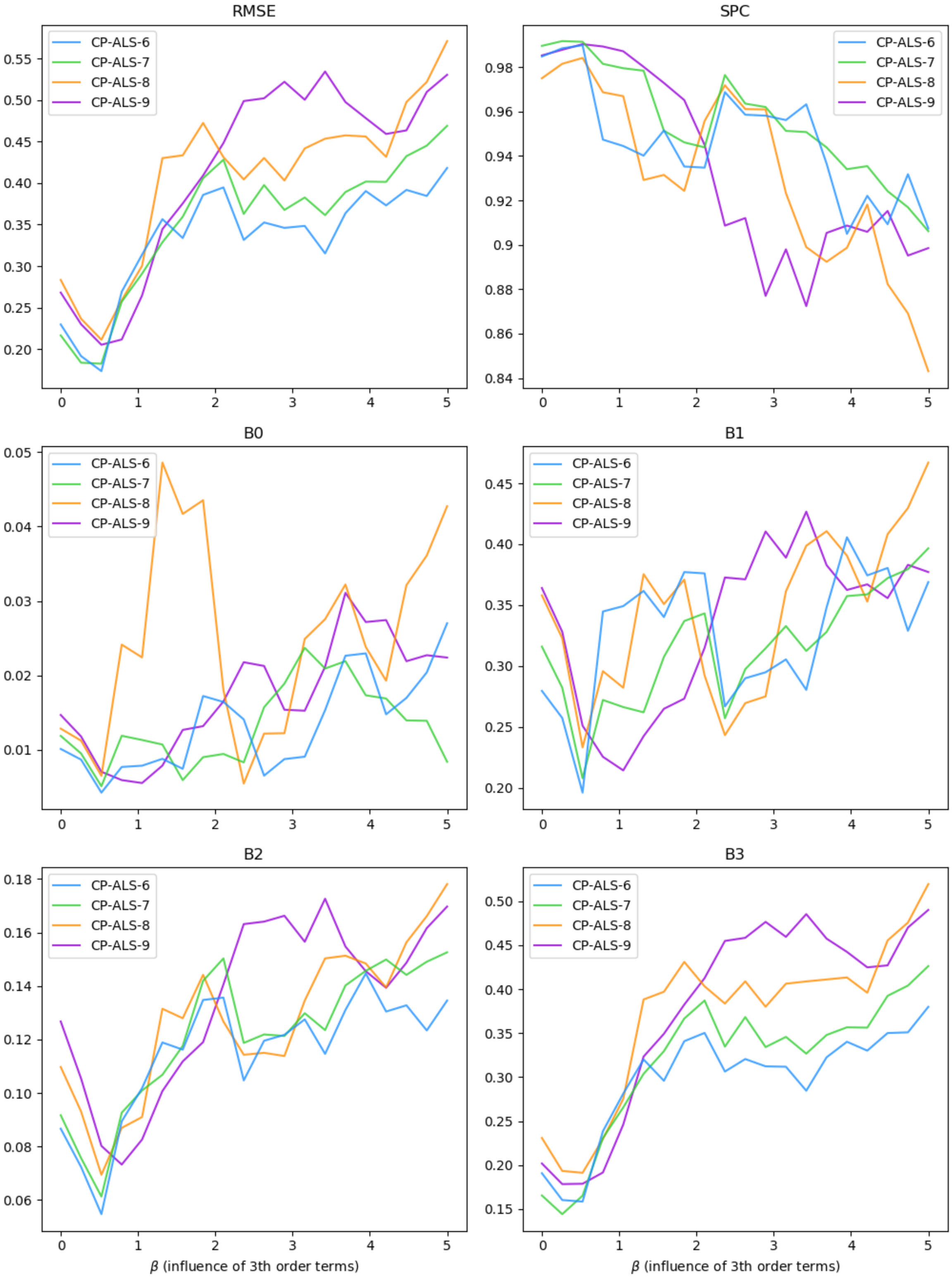

Figure 9: CP Decomposition based Imputation using ALS for different ranks, p=5%. Choosing a rank close to the true rank performs best.

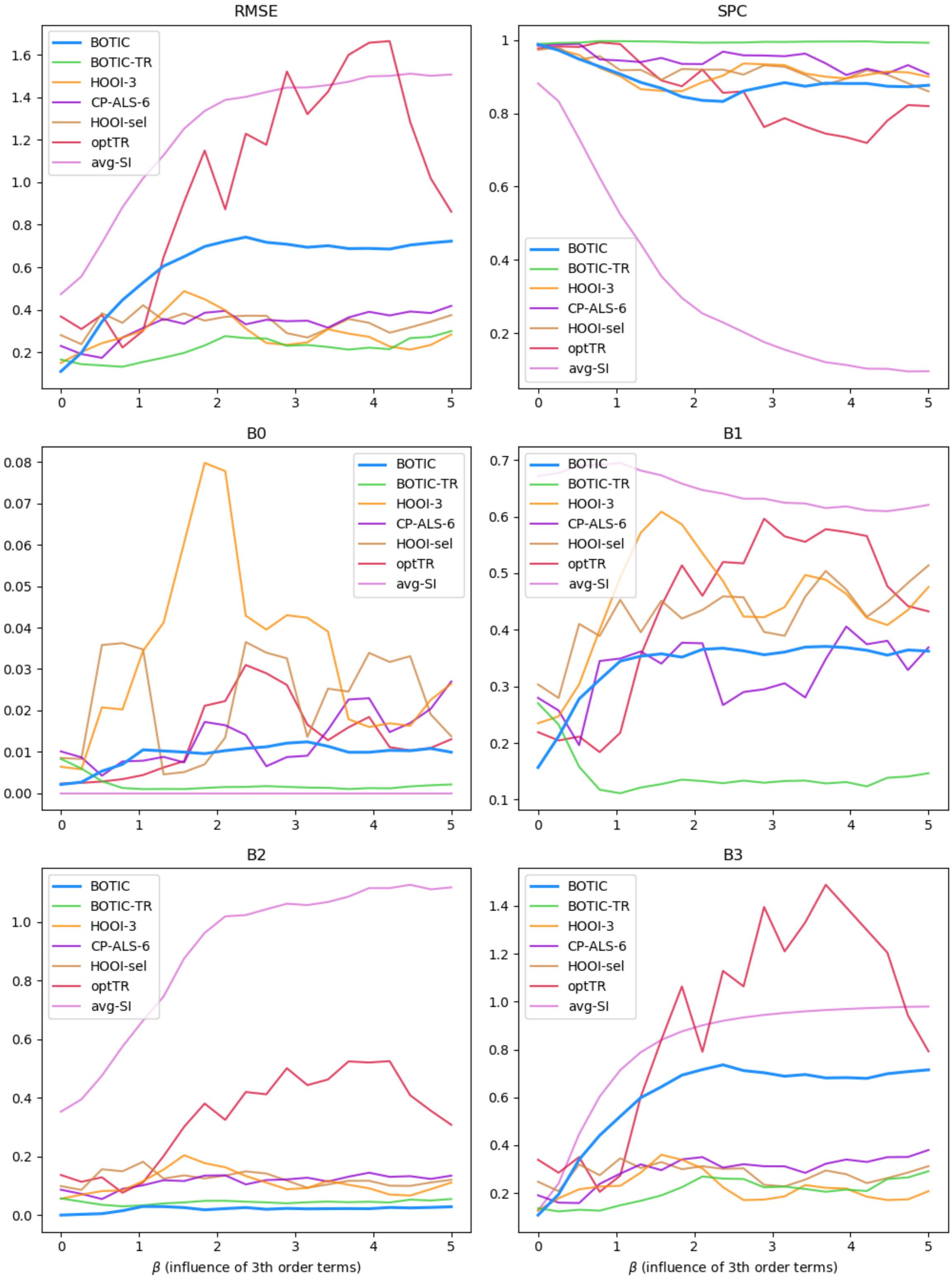

Figure 10: All methods under an increasing relevance of third order terms, p=5%.

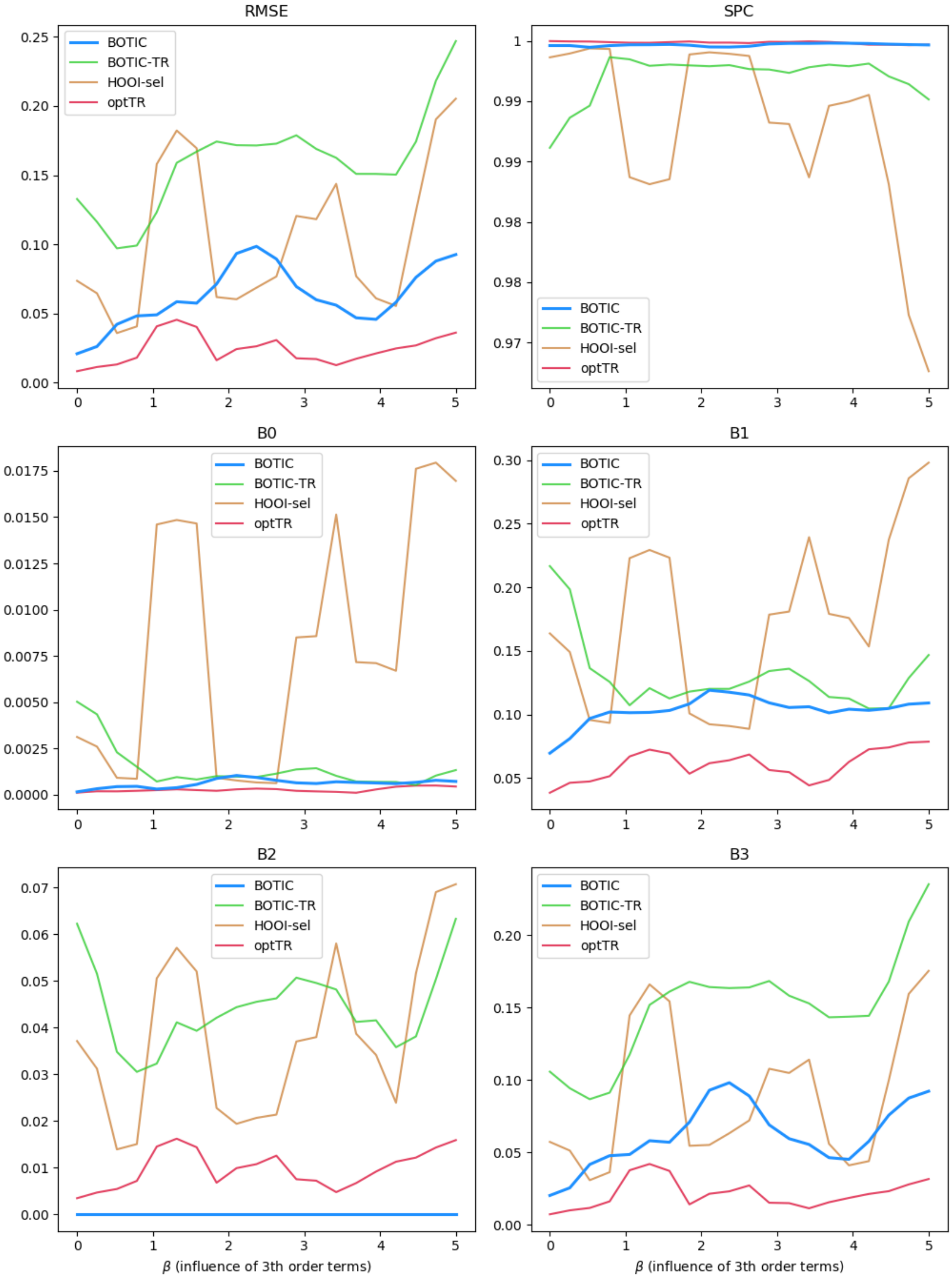

Figure 11: A selection of methods under an increasing relevance of third order terms, p=10%. Note that avg-SI is excluded due to poor performance.

7.2 Convergence

We will now proof Proposition 4.1 and thus show that the proposed algorithm converges. We will largely follow the structure of the proof of convergence for OMIC [1].
Remember that

$$\mathbf{S}_\Lambda(\mathbf{R}) = \sum_{k_1,k_2,k_3=1}^{K_1,K_2,K_3} S_{\lambda_{k_1k_2k_3}}(\mathbf{R} \times_1 (X^{k_1})^T \times_2 (Y^{k_2})^T \times_3 (Z^{k_3})^T) \times_1 X^{k_1} \times_2 Y^{k_2} \times_3 Z^{k_3}$$

is the optimal solution to

$$\min_{\mathbf{M}} \frac{1}{2}\left\|\mathbf{R} - \sum_{k_1,k_2,k_3=1}^{K_1,K_2,K_3} \mathbf{\Pi}^{k_1k_2k_3}(\mathbf{M})\right\|_F^2 + \sum_{k_1,k_2,k_3=1}^{K_1,K_2,K_3} \lambda_{k_1k_2k_3}\mathcal{R}(\mathbf{P}^{k_1k_2k_3}(\mathbf{M}))$$

and $\mathbf{S}_\lambda(\mathbf{R})$ is the optimal solution to

$$\min_{\mathbf{M}} \frac{1}{2}\|\mathbf{R} - \mathbf{M}\|_F^2 + \lambda\mathcal{R}(\mathbf{M}).$$

We will only show Proposition 4.1 for the regularization $\mathcal{R}(\mathbf{M}) = \sum_{k=1}^{d} \gamma_k \|\mathbf{P}_k(\mathbf{M})\|_*$ choosen in Section 4.3. First we need to proof a property of the operator $\mathbf{S}_\lambda$. This proof is based on Proof A.2 in [12].

Lemma 7.1. *For any two tensors $\boldsymbol{R}_1, \boldsymbol{R}_2 \in \mathbb{R}^{m_1\times m_2\times m_3}$*
$\|\boldsymbol{S}_\lambda(\boldsymbol{R}_1) - \boldsymbol{S}_\lambda(\boldsymbol{R}_2)\|_F \leq \|\boldsymbol{R}_1 - \boldsymbol{R}_2\|_F$.

Proof.
Let $\hat{\mathbf{M}}_i := \mathbf{S}_\lambda(\mathbf{M}_i)$ for i $\in \{1,2\}$.
Note that $\hat{\mathbf{M}}_i$ is an optimal solution to (14) (for $\mathbf{R} = \mathbf{R}_i$). Because the objective function of (14) is convex we know by Lemma 2.16 that for $i \in \{1,2\}$:

$$0 \in \hat{\mathbf{M}}_i - \mathbf{R} + \sum_{k=1}^{3} \gamma_k \partial \left\|\mathbf{P}_k(\hat{\mathbf{M}})\right\|_*$$

Let $f_k : \mathbb{R}^{m_1\times m_2\times m_3} \to \mathbb{R}, \mathbf{Z} \mapsto \|\mathbf{P}_k(\mathbf{Z})\|_*$ and $\hat{f}_k : \mathbb{R}^{m_k\times(m_1m_2m_3/m_k)} \to \mathbb{R}, Z \mapsto \|Z\|_*$.
Let $p_k(\hat{\mathbf{M}}_i)$ denote an element from $\partial f_k(\hat{\mathbf{M}}_i)$. We get

$$\hat{\mathbf{M}}_i - \mathbf{M}_i + \sum_{k=1}^{3} \gamma_k p_k(\hat{\mathbf{M}}_i) = 0$$

and thus

$$(\hat{\mathbf{M}}_1 - \hat{\mathbf{M}}_2) - (\mathbf{M}_1 - \mathbf{M}_2) + \sum_{k=1}^{3} \gamma_k(p_k(\hat{\mathbf{M}}_1) - p_k(\hat{\mathbf{M}}_2)) = 0$$

which implies

$$\begin{aligned}\langle \hat{\mathbf{M}}_1 - \hat{\mathbf{M}}_2, \hat{\mathbf{M}}_1 - \hat{\mathbf{M}}_2\rangle - \langle \mathbf{M}_1 - \mathbf{M}_2, \hat{\mathbf{M}}_1 - \hat{\mathbf{M}}_2\rangle +\\ \sum_{k=1}^{3} \gamma_k \langle p_k(\hat{\mathbf{M}}_1) - p_k(\hat{\mathbf{M}}_2), \hat{\mathbf{M}}_1 - \hat{\mathbf{M}}_2\rangle = 0.\end{aligned}$$

Using the definition of the subgradient we see that

$$\begin{aligned}&p_k(\hat{\mathbf{M}}_i) \text{ is a subgradient of } f_k \text{ at } \hat{\mathbf{M}}_i\\ \Leftrightarrow\ &\forall \mathbf{Z} \in \mathbb{R}^{m_1\times m_2\times m_3} : f_k(\mathbf{Z}) \geq f_k(\hat{\mathbf{M}}_i) + \langle p_k(\hat{\mathbf{M}}_i), \mathbf{Z} - \hat{\mathbf{M}}_i\rangle\\ \Leftrightarrow\ &\forall \mathbf{Z} \in \mathbb{R}^{m_1\times m_2\times m_3} : f_k(\mathbf{Z}) \geq f_k(\hat{\mathbf{M}}_i) + \langle \mathbf{P}_k(p_k(\hat{\mathbf{M}}_i)), \mathbf{P}_k(\mathbf{Z} - \hat{\mathbf{M}}_i)\rangle\\ \Leftrightarrow\ &\forall \mathbf{Z} \in \mathbb{R}^{m_1\times m_2\times m_3} : \hat{f}_k(\mathbf{P}_k(\mathbf{Z})) \geq \hat{f}_k(\mathbf{P}_k(\hat{\mathbf{M}}_i)) + \langle \mathbf{P}_k(p_k(\hat{\mathbf{M}}_i)), \mathbf{P}_k(\mathbf{Z}) - \mathbf{P}_k(\hat{\mathbf{M}}_i)\rangle\\ \Leftrightarrow\ &\mathbf{P}_k(p_k(\hat{\mathbf{M}}_i)) \text{ is a subgradient of } \hat{f}_k \text{ at } \mathbf{P}_k(\hat{\mathbf{M}}_i).\end{aligned} \tag{21}$$

From Proof A.2 in [12] we know that for two matrices Z_1, Z_2 and $p_{Z_i} \in \partial\|Z_i\|_*$

$$\langle p_{Z_1} - p_{Z_2}, Z_1 - Z_2\rangle \geq 0 \tag{22}$$

and thus

$$\begin{aligned}&\langle p_k(\hat{\mathbf{M}}_1) - p_k(\hat{\mathbf{M}}_2), \hat{\mathbf{M}}_1 - \hat{\mathbf{M}}_2\rangle\\ =\ &\langle \mathbf{P}_k(p_k(\hat{\mathbf{M}}_1)) - \mathbf{P}_k(p_k(\hat{\mathbf{M}}_2)), \mathbf{P}_k(\hat{\mathbf{M}}_1) - \mathbf{P}_k(\hat{\mathbf{M}}_2)\rangle\\ \geq\ &0. && \text{Using (21) and (22)}\end{aligned}$$

Therefore,

$$\sum_{k=1}^{3} \gamma_k \langle p_k(\hat{\mathbf{M}}_1) - p_k(\hat{\mathbf{M}}_2), \hat{\mathbf{M}}_1 - \hat{\mathbf{M}}_2\rangle \geq 0. \tag{23}$$

So finally

$$\begin{aligned}\left\|\hat{\mathbf{M}}_1 - \hat{\mathbf{M}}_2\right\|_F^2 =&\langle \hat{\mathbf{M}}_1 - \hat{\mathbf{M}}_2, \hat{\mathbf{M}}_1 - \hat{\mathbf{M}}_2\rangle\\ \leq&\langle \mathbf{M}_1 - \mathbf{M}_2, \hat{\mathbf{M}}_1 - \hat{\mathbf{M}}_2\rangle && \text{Using (23)}\\ \leq&\|\mathbf{M}_1 - \mathbf{M}_2\|_F \left\|\hat{\mathbf{M}}_1 - \hat{\mathbf{M}}_2\right\|_F && \text{(Cauchy-Schwarz)}\\ =&\|\mathbf{M}_1 - \mathbf{M}_2\|_F \left\|\hat{\mathbf{M}}_1 - \hat{\mathbf{M}}_2\right\|_F.\end{aligned}$$

The result follows. □

Now we can show a similar property for S_Λ, in particular we will show that it S_Λ is a continous map.

Lemma 7.2. *For any two tensors* $\boldsymbol{R}_1, \boldsymbol{R}_2 \in \mathbb{R}^{m_1 \times m_2 \times m_3}$
$\|\boldsymbol{S}_\Lambda(\boldsymbol{R}_1) - \boldsymbol{S}_\Lambda(\boldsymbol{R}_2)\|_F \leq \|\boldsymbol{R}_1 - \boldsymbol{R}_2\|_F$.

Proof.

$$\begin{aligned}
&\|\mathbf{S}_\Lambda(\mathbf{R}_1) - \mathbf{S}_\Lambda(\mathbf{R}_2)\|_F \\
=&\Bigg\| \sum_{k_1,k_2,k_3=1}^{K_1,K_2,K_3} S_{\lambda_{k_1k_2k_3}}(\mathbf{R}_1 \times_1 (X^{k_1})^T \times_2 (Y^{k_2})^T \times_3 (Z^{k_3})^T) \times_1 X^{k_1} \times_2 Y^{k_2} \times_3 Z^{k_3} \\
&\quad - \sum_{k_1,k_2,k_3=1}^{K_1,K_2,K_3} S_{\lambda_{k_1k_2k_3}}(\mathbf{R}_2 \times_1 (X^{k_1})^T \times_2 (Y^{k_2})^T \times_3 (Z^{k_3})^T) \times_1 X^{k_1} \times_2 Y^{k_2} \times_3 Z^{k_3} \Bigg\| \\
=&\Bigg\| \sum_{k_1,k_2,k_3=1}^{K_1,K_2,K_3} S_{\lambda_{k_1k_2k_3}}((\mathbf{R}_1 - \mathbf{R}_2) \times_1 (X^{k_1})^T \times_2 (Y^{k_2})^T \times_3 (Z^{k_3})^T) \times_1 X^{k_1} \times_2 Y^{k_2} \times_3 Z^{k_3} \Bigg\| \\
=& \sum_{k_1,k_2,k_3=1}^{K_1,K_2,K_3} \left\| S_{\lambda_{k_1k_2k_3}}((\mathbf{R}_1 - \mathbf{R}_2) \times_1 (X^{k_1})^T \times_2 (Y^{k_2})^T \times_3 (Z^{k_3})^T) \right\| \\
\leq& \sum_{k_1,k_2,k_3=1}^{K_1,K_2,K_3} \left\| (\mathbf{R}_1 - \mathbf{R}_2) \times_1 (X^{k_1})^T \times_2 (Y^{k_2})^T \times_3 (Z^{k_3})^T) \right\| \qquad \text{by Lemma 7.1} \\
=&\|\mathbf{R}_1 - \mathbf{R}_2\|_F
\end{aligned}$$

□

As an extension to the loss

$$\mathcal{L}(\mathbf{M}) = \frac{1}{2}\|\mathbf{R}_\Omega - \mathbf{P}_\Omega(\mathbf{M})\|_F^2 + \sum_{k_1,k_2,k_3=1}^{K_1,K_2,K_3} \Lambda_{k_1k_2k_3}\mathcal{R}(\mathbf{P}^{k_1,k_2,k_3}(\mathbf{M}))$$

we define

$$Q(\mathbf{A}|\mathbf{B}) = \frac{1}{2}\|\mathbf{R}_\Omega + \mathbf{P}_{\Omega^\perp}(\mathbf{B}) - \mathbf{A}\|_F^2 + \sum_{k_1,k_2,k_3=1}^{K_1,K_2,K_3} \Lambda_{k_1k_2k_3}\mathcal{R}(\mathbf{P}^{k_1,k_2,k_3}(\mathbf{A})).$$

Note that $\mathcal{L}(\mathbf{M}) = Q(\mathbf{M}|\mathbf{M})$ and $\mathbf{M}^{i+1} = \operatorname{argmin}_{\mathbf{M}} Q(\mathbf{M}|\mathbf{M}^i)$.

In the following let $\mathbf{M}^0 \in \mathbb{R}^{m_1 \times m_2 \times m_3}$ and $\mathbf{M}^{i+1} = \operatorname{argmin}_{\mathbf{M}} Q(\mathbf{M}|\mathbf{M}^i)$.
We will show that the loss decreases monotonically.

Lemma 7.3.

$$\mathcal{L}(\boldsymbol{M}^{i+1}) \leq Q(\boldsymbol{M}^{i+1}|\boldsymbol{M}^i) \leq \mathcal{L}(\boldsymbol{M}^i)$$

Proof.

$$\begin{aligned}
\mathcal{L}(\mathbf{M}^i) &= Q(\mathbf{M}^i|\mathbf{M}^i) \\
&= \frac{1}{2}\left\|\mathbf{R}_\Omega + \mathbf{P}_{\Omega^\perp}(\mathbf{M}^i) - \mathbf{M}^i\right\|_F^2 + \sum_{k_1,k_2,k_3=1}^{K_1,K_2,K_3} \lambda_{k_1k_2k_3}\mathcal{R}(\mathbf{P}^{k_1k_2k_3}(\mathbf{M}^i)) \\
&\geq \min_{\mathbf{M}} \frac{1}{2}\left\|\mathbf{R}_\Omega + \mathbf{P}_{\Omega^\perp}(\mathbf{M}^i) - \mathbf{M}\right\|_F^2 + \sum_{k_1,k_2,k_3=1}^{K_1,K_2,K_3} \lambda_{k_1k_2k_3}\mathcal{R}(\mathbf{P}^{k_1k_2k_3}(\mathbf{M})) \\
&= Q(\mathbf{M}^{i+1}|\mathbf{M}^i) \\
&= \frac{1}{2}\left\|(\mathbf{R}_\Omega - \mathbf{P}_\Omega(\mathbf{M}^{i+1})) + (\mathbf{P}_{\Omega^\perp}(\mathbf{M}^i) - \mathbf{P}_{\Omega^\perp}(\mathbf{M}^{i+1})\right\|_F^2 + \sum_{k_1,k_2,k_3=1}^{K_1,K_2,K_3} \lambda_{k_1k_2k_3}\mathcal{R}(\mathbf{P}^{k_1k_2k_3}(\mathbf{M}^{i+1})) \\
&\geq \frac{1}{2}\left\|\mathbf{R}_\Omega - \mathbf{P}_\Omega(\mathbf{M}^{i+1})\right\|_F^2 + \sum_{k_1,k_2,k_3=1}^{K_1,K_2,K_3} \lambda_{k_1k_2k_3}\mathcal{R}(\mathbf{P}^{k_1k_2k_3}(\mathbf{M}^{i+1})) \\
&= \mathcal{L}(\mathbf{M}^{i+1})
\end{aligned}$$

□

Now we show that the difference between the iterates decreases monotonically to zero,

Lemma 7.4.

$$\left\|\boldsymbol{M}^i - \boldsymbol{M}^{i+1}\right\|_F \leq \left\|\boldsymbol{M}^{i-1} - \boldsymbol{M}^i\right\|_F$$

Furthermore,

$$\boldsymbol{M}^i - \boldsymbol{M}^{i-1} \to 0 \text{ as } i \to \infty.$$

Proof. This proof is almost the same as that of Lemma B.3 in **OIMC**.
First we have

$$\begin{aligned}
\left\|\mathbf{M}^i - \mathbf{M}^{i+1}\right\|_F &= \left\|\mathbf{S}_\Lambda(\mathbf{R}_\Omega + \mathbf{P}_{\Omega^\perp}(\mathbf{M}^{i-1})) - \mathbf{S}_\Lambda(\mathbf{R}_\Omega + \mathbf{P}_{\Omega^\perp}(\mathbf{M}^i))\right\| \\
&\leq \left\|(\mathbf{R}_\Omega + \mathbf{P}_{\Omega^\perp}(\mathbf{M}^{i-1})) - (\mathbf{R}_\Omega + \mathbf{P}_{\Omega^\perp}(\mathbf{M}^i))\right\| && \text{by Lemma 7.2} \\
&= \left\|\mathbf{P}_{\Omega^\perp}(\mathbf{M}^{i-1}) - \mathbf{P}_{\Omega^\perp}(\mathbf{M}^i)\right\| \\
&\leq \left\|\mathbf{M}^{i-1} - \mathbf{M}^i\right\|.
\end{aligned}$$

Therefore $\left\|\mathbf{M}^i - \mathbf{M}^{i+1}\right\|_F$ is a monotone and bounded sequence and must convergence as $i \to \infty$. In particular, $\left\|\mathbf{M}^i - \mathbf{M}^{i+1}\right\|_F - \left\|\mathbf{M}^{i-1} - \mathbf{M}^i\right\|_F \to 0$ and by the last inequalty from above

$$\left\|\mathbf{P}_{\Omega^\perp}(\mathbf{M}^{i-1}) - \mathbf{P}_{\Omega^\perp}(\mathbf{M}^i)\right\| - \left\|\mathbf{M}^{i-1} - \mathbf{M}^i\right\|_F \to 0.$$

which shows the following:

$$\left\|\mathbf{P}_\Omega(\mathbf{M}^{i-1}) - \mathbf{P}_\Omega(\mathbf{M}^i)\right\| \to 0. \tag{24}$$

Similarly Lemma 7.3 shows that $\mathcal{L}(\mathbf{M}^i) - \mathcal{L}(\mathbf{M}^{i+1}) \to 0$ and thus also

$$Q(\mathbf{M}^{i+1}|\mathbf{M}^i) - Q(\mathbf{M}^i|\mathbf{M}^i) \to 0.$$

Furthermore since,

$$\begin{aligned}
&Q(\mathbf{M}^{i+1}|\mathbf{M}^i) - Q(\mathbf{M}^i|\mathbf{M}^i) \\
=&\frac{1}{2}\left\|\mathbf{R}_\Omega + \mathbf{P}_{\Omega^\perp}(\mathbf{M}^i) - \mathbf{M}^{i+1}\right\|_F^2 + \sum_{k_1,k_2,k_3=1}^{K_1,K_2,K_3} \lambda_{k_1k_2k_3}\mathcal{R}(\mathbf{P}^{k_1k_2k_3}(\mathbf{M}^{i+1})) \\
&-\frac{1}{2}\left\|\mathbf{R}_\Omega + \mathbf{P}_{\Omega^\perp}(\mathbf{M}^{i+1}) - \mathbf{M}^{i+1}\right\|_F^2 - \sum_{k_1,k_2,k_3=1}^{K_1,K_2,K_3} \lambda_{k_1k_2k_3}\mathcal{R}(\mathbf{P}^{k_1k_2k_3}(\mathbf{M}^{i+1})) \\
=&\left\|\mathbf{P}_{\Omega^\perp}(\mathbf{M}^{i-1}) - \mathbf{P}_{\Omega^\perp}(\mathbf{M}^i)\right\|
\end{aligned}$$

we can conclude:

$$\left\|\mathbf{P}_{\Omega^\perp}(\mathbf{M}^{i-1}) - \mathbf{P}_{\Omega^\perp}(\mathbf{M}^i)\right\| \to 0. \tag{25}$$

Finally combining (24) and (25) we get

$$\left\|\mathbf{M}^{i-1} - \mathbf{M}^i\right\|_F \to 0$$

and thus

$$\mathbf{M}^{i-1} - \mathbf{M}^i \to 0.$$

□

By compactness there exists at least one subsequence $\mathbf{M}^{n_i}$ such that
$\mathbf{M}^{n_i} \to \mathbf{M}^\infty$ as $i \to \infty$. Next, we will show that $\mathbf{M}^\infty$ is a solution to our problem. We need the following technical result.

Proposition 7.5.
Let $p_i \in \partial \sum_{k_1,k_2,k_3=1}^{K_1,K_2,K_3} \Lambda_{k_1k_2k_3} \sum_{k=1}^{d} \gamma_k \left\|\boldsymbol{P}_k(\boldsymbol{P}^{k_1k_2k_3}(\boldsymbol{M}^{n_i}))\right\|_*$ be a sequence of subgradients of the regularizer $\sum_{k_1,k_2,k_3=1}^{K_1,K_2,K_3} \Lambda_{k_1k_2k_3}\mathcal{R}(\boldsymbol{P}^{k_1k_2k_3}(\boldsymbol{M}^{n_i}))$ evaluated at $\boldsymbol{M}^{n_i}$. There exists a subsequence $p_{\hat{n}_i}$ which converges to some

$$p \in \partial \sum_{k_1,k_2,k_3=1}^{K_1,K_2,K_3} \lambda_{k_1k_2k_3} \sum_{k=1}^{d} \gamma_k \left\|\boldsymbol{P}_k(\boldsymbol{P}^{k_1k_2k_3}(\boldsymbol{M}^\infty))\right\|_*,$$

a subgradient of the regularizer at $\boldsymbol{M}^\infty$.

Proof. Follows the same arguments as in the proof of Lemma B.4 in OMIC [1]. □

Lemma 7.6.
$\boldsymbol{M}^\infty = \lim_{i\to\infty} \boldsymbol{M}^{n_i}$ *is a solution of* (13) *and thus:*

$$\boldsymbol{M}^\infty = \boldsymbol{S}_\Lambda(\boldsymbol{R}_\Omega + \boldsymbol{P}_{\Omega^\perp}(\boldsymbol{M}^\infty))$$

Proof.
First note that $\mathbf{M}^{n_i} - \mathbf{M}^{n_i-1} \to 0$ by Lemma 7.4. We can conclude

$$\mathbf{R}_\Omega + \mathbf{P}_{\Omega^\perp}(\mathbf{M}^{n_i-1}) - \mathbf{M}^{n_i} \to \mathbf{R}_\Omega - \mathbf{P}_\Omega(\mathbf{M}^\infty).$$

Furthermore by the definition of $\mathbf{M}^{n_i}$

$$0 \in \partial Q(\mathbf{M}|\mathbf{M}^{n_i-1}) = -(\mathbf{R}_\Omega + \mathbf{P}_{\Omega^\perp}(\mathbf{M}^{n_i-1}) - \mathbf{M}^{n_i}) + \partial \sum_{k_1,k_2,k_3=1}^{K_1,K_2,K_3} \lambda_{k_1k_2k_3} \mathcal{R}(\mathbf{P}^{k_1k_2k_3}(\mathbf{M}^{n_i})).$$

Hence, we can choose $p_i \in \partial \sum_{k_1,k_2,k_3=1}^{K_1,K_2,K_3} \lambda_{k_1k_2k_3} \mathcal{R}(\mathbf{P}^{k_1k_2k_3}(\mathbf{M}^{n_i}))$ such that

$$p_i - (\mathbf{R}_\Omega + \mathbf{P}_{\Omega^\perp}(\mathbf{M}^{n_i-1}) - \mathbf{M}^{n_i}) = 0.$$

Now, by Lemma 7.5 there exists a subsequence $p_{\hat{n}_i}$ such that $p_{\hat{n}_i} \to p$ for some

$$p \in \partial \sum_{k_1,k_2,k_3=1}^{K_1,K_2,K_3} \lambda_{k_1k_2k_3} \mathcal{R}(\mathbf{P}^{k_1k_2k_3}(\mathbf{M}^\infty)).$$

We obtain

$$0 = p_{\hat{n}_i} - (\mathbf{R}_\Omega + \mathbf{P}_{\Omega^\perp}(\mathbf{M}^{n_i-1}) - \mathbf{M}^{n_i}) \to p - (\mathbf{R}_\Omega - \mathbf{P}_\Omega(\mathbf{M}^\infty)),$$

so finally

$$0 \in \partial \mathcal{L}(\mathbf{M}^\infty)$$

which together with Lemma 2.16 shows the first part of the Lemma.
To see the second part note that since $\mathbf{M}^{n_i} - \mathbf{M}^{n_i-1} \to 0$ also $\mathbf{M}^{n_i-1} \to \mathbf{M}^\infty$. Furthermore, using the continouity of $\mathbf{S}_\Lambda$ we get

$$\mathbf{M}^\infty = \lim_{i\to\infty} \mathbf{M}^{n_i} = \lim_{i\to\infty} \mathbf{S}_\Lambda(\mathbf{R}_\Omega + \mathbf{P}_{\Omega^\perp}(\mathbf{M}^{n_i-1})) = \mathbf{S}_\Lambda(\mathbf{R}_\Omega + \mathbf{P}_{\Omega^\perp}(\mathbf{M}^\infty)).$$

□

Proof of Proposition 4.1. We have already shown that any limit point of $\mathbf{M}^i$ is a solution to (13). Thus it suffices to show that $\mathbf{M}^i$ converges. We have for any i:

$$\begin{aligned}\left\|\mathbf{M}^\infty - \mathbf{M}^i\right\|_F^2 &= \left\|\mathbf{S}_\Lambda(\mathbf{R}_\Omega + \mathbf{P}_{\Omega^\perp}(\mathbf{M}^\infty)) - \mathbf{S}_\Lambda(\mathbf{R}_\Omega + \mathbf{P}_{\Omega^\perp}(\mathbf{M}^{i-1}))\right\|_F^2 \\ &\leq \left\|(\mathbf{R}_\Omega + \mathbf{P}_{\Omega^\perp}(\mathbf{M}^\infty)) - (\mathbf{R}_\Omega + \mathbf{P}_{\Omega^\perp}(\mathbf{M}^{i-1}))\right\|_F^2 \\ &= \left\|\mathbf{P}_{\Omega^\perp}(\mathbf{M}^\infty) - \mathbf{M}^{i-1}\right\|_F^2 \leq \left\|\mathbf{M}^\infty - \mathbf{M}^{i-1}\right\|_F^2,\end{aligned}$$

where at the second line we have used Lemma 7.2. Because $\|\mathbf{M}^\infty - \mathbf{M}^{n_i}\|_F^2 \to 0$ and $\left\|\mathbf{M}^\infty - \mathbf{M}^i\right\|_F^2$ is monotonically decreasing we get $\left\|\mathbf{M}^\infty - \mathbf{M}^i\right\|_F^2 \to 0$ and thus:

$$\lim_{i\to\infty} \mathbf{M}^i = \lim_{i\to\infty} \mathbf{M}^{n_i} = \mathbf{M}^\infty.$$

This concludes the proof of convergence.

□

References

[1] A. Ledent, R. Alves, and M. Kloft. “Orthogonal Inductive Matrix Completion”. arXiv:2004.01653. 2020.

[2] Q. Song, H. Ge, J. Caverlee, and X. Hu. “Tensor completion algorithms in big data analytics”. In: *ACM Transactions on Knowledge Discovery from Data, January 2019, Article No.: 6* (2019).

[3] T. Kolda and B. Bader. “Tensor decompositions and applications”. In: *SIAM Review* 51 (2009), pp. 455–500.

[4] S. Brunton and J. Kutz. *Data Driven Science & Engineering.* Cambridge University Press, 2017. Chap. 1.

[5] C. Eckart and G. Young. “The approximation of one matrix by another of lower rank”. In: *Psychometrika* (1936), pp. 211–218.

[6] M. Kilmera and C. Martin. “Factorization strategies for third-order tensors”. In: *Linear Algebra and its Applications* 435/3 (2015), pp. 641–658.

[7] I. Oseledets. “Tensor-Train Decomposition”. In: *Journal on Scientific Computing* 33/5 (2011), pp. 2295–2317.

[8] L. Wang, M. Chu, and B. Yu. “Orthogonal low rank tensor approximation: Alternating Least Squares method and its global convergence”. In: *SIAM Journal on Matrix Analysis and Applications* 36/1 (2014), pp. 1–19.

[9] L. de Lathauwer, B. de Moor, and J. Vandewalle. “A multilinear Singular Value Decomposition”. In: *SIAM Journal on Matrix Analysis and Applications* 21 (2000), pp. 1253–1278.

[10] L. de Lathauwer, B. de Moor, and J. Vandewalle. “On the best rank-1 and rank-(R1, R2,..., RN) approximation of higher order tensors”. In: *SIAM Journal on Matrix Analysis and Applications* 21 (2000), pp. 1324–1342.

[11] J. Cai, E. Candès, and Z. Shen. “A singular value thresholding algorithm for matrix completion”. In: *SIAM J. Optim.* 20 (2010), pp. 1956–1982.

[12] R. Mazumder, T. Hastiem, and R. Tibshirani. “Spectral regularization algorithms for learning large incomplete matrices”. In: *Journal of Machine Learning Research* (2010), pp. 2287–2322.

[13] E. Acara, D. Dunlavy, T. Kolda, and M. Mørupd. “Scalable tensor factorizations for incomplete data”. In: *Chemometrics and Intelligent Laboratory Systems* 106/1 (2011), pp. 41–56.

[14] Q. Zhao, L. Zhang, and A. Cichocki. “Bayesian CP factorization of incomplete tensors with automatic rank determination”. In: *IEEE transactions on pattern analysis and machine intelligence* 37/9 (2015), pp. 1751–1763.

[15] M. Filipović and A. Jukić. “Tucker factorization with missing data with application to low-n-rank tensor completion”. In: *Multidimensional systems and signal processing* 26/3 (2015).

[16] K. Wimalawarne, M. Sugiyama, and R. Tomioka. "Multitask learning meets tensor factorization: task imputation via convex optimization". In: *Advances in Neural Information Processing Systems* 27 (2014), pp. 2825–2833.

[17] J. Liu, P. Musialski, P. Wonka, and J. Ye. "Tensor completion for estimating missing values in visual data". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35/1 (2013), pp. 208–220.

[18] R. Tomioka, K. Hayashi, and H. Kashima. "Estimation of Low-Rank Tensors via Convex Optimization". arXiv:1010.0789. 2011.

[19] X. Guo, Q. Yao, and J. Kwok. "Efficient Sparse Low-Rank Tensor Completion Using the Frank-Wolfe Algorithm". In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence* (2017), pp. 1948–1954.

[20] L. Yuana, Q. Zhaob, L. Guia, and J. Cao. "High-order tensor completion via gradient-based optimization under tensor train format". In: *Signal Processing: Image Communication* (2018), pp. 53–61.

[21] C. Lu, X. Peng, and Y. Wei. "Low-rank tensor completion with a new tensor nuclear norm induced by invertible linear transforms". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019).

[22] R. Tomioka, T. Suzuki, and M. Sugiyama. "Super-linear convergence of dual augmented Lagrangian algorithm for sparsity regularized estimation". In: *Journal of Machine Learning Research* 12 (2011), pp. 1537–1586.

[23] P. Lions and B. Mercier. "Splitting algorithms for the sum of two nonlinear operators". In: *SIAM Journal on Numerical Analysis* (1969), pp. 964–979.

[24] R. Lehoucq, D. Sorensen, and C. Yang. *ARPACK Users Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*. SIAM, 1998.

[25] R. Bro. "Multi-way analysis in the food industry: models, algorithms, and applications". In: *Proc ICSLP* (1998).

Declaration

I, Justus Will, avouch that I have created this thesis on my own and without help or sources but those explicitly mentioned and that I have marked any and all citations as such.

Kaiserslautern, November 10, 2020

Justus Will