An Analysis of Tensor Models for Learning on Structured Data

Maximilian Nickel[1] and Volker Tresp[2]

[1] Ludwig Maximilian University, Oettingenstr. 67, Munich, Germany
nickel@dbs.ifi.lmu.de
[2] Siemens AG, Corporate Technology, Otto-Hahn-Ring 6, Munich, Germany
volker.tresp@siemens.com

Abstract. While tensor factorizations have become increasingly popular for learning on various forms of structured data, only very few theoretical results exist on the generalization abilities of these methods. Here, we discuss the tensor product as a principled way to represent structured data in vector spaces for machine learning tasks. By extending known bounds for matrix factorizations, we are able to derive generalization error bounds for the tensor case. Furthermore, we analyze analytically and experimentally how tensor factorization behaves when applied to over- and understructured representations, for instance, when two-way tensor factorization, i.e. matrix factorization, is applied to three-way tensor data.

Keywords: Tensor Factorization, Structured Data, Generalization Error Bounds.

1 Introduction

Learning from structured data is a very active line of research in a variety of fields, including social network analysis, natural language processing, bioinformatics, and artificial intelligence. While tensor factorizations have a long tradition in psycho- and chemometrics, only more recently have they been applied to various tasks on structured data in machine learning. Examples include link prediction and entity resolution on multi-relational data [18,13] and large knowledge bases [3,19], item recommendation on sequential data [20,21], or the analysis of time varying social networks [2]; only to name a few examples. A reason for the success of tensor methods in these tasks is their very appealing property to efficiently impose structure on the vector space representation of data. Moreover, tensor factorizations can be related to multilinear models, which overcome some limitations of linear models, such as their limited expressiveness, but at the same time remain more scalable and easier to handle than non-linear approaches. However, despite their increasing popularity and their appealing properties, only very few theoretical results exist on the generalization abilities of tensor factorizations. Furthermore, an important open question is what kind of generalization improvements over simpler, less structured models can be expected. For instance,

H. Blockeel et al. (Eds.): ECML PKDD 2013, Part II, LNAI 8189, pp. 272–287, 2013.

propositionalization, which transforms relational data into feature-based representations, has been considered as a mean for relational learning [15,12]. In terms of tensor factorization, propositionalization would be equivalent to transforming a tensor into a matrix representation prior to computing the factorization. While it has been shown empirically that tensor methods usually scale better with the amount of missing data than their matrix counterparts [26,16,25,22] and that they can yield significantly improved results over "flat" methods which ignore a large part of the data structure [18], no theoretical justification of this behavior is known in terms of generalization bounds.

In this paper, we approach several of these open questions. First, we will briefly discuss the tensor product as a principled way to derive vector space representations of structured data. Subsequently, we will present the first generalization error bounds of tensor factorizations for classification tasks. We will analyze experimentally the effect of imposing structure on vector space representations via the tensor product as well as the effect of constraints that are applied to popular tensor decompositions. Based on the newly derived bounds we discuss how these results can be interpreted analytically.

2 Theory and Methods

In this section we will briefly review concepts related to tensor factorization, as far as they are important for the course of this paper. Furthermore, we will discuss how structured data can be modeled as weighted sets of n-tuples, which enables a closer analysis of the relations between tensor factorizations and structured data.

In the following, scalars will be denoted by lowercase letters x; vectors will be denoted by bold lowercase letters $\boldsymbol{x}, \boldsymbol{y}$ with elements x_i, y_j. Vectors are assumed to be column vectors. Matrices will be denoted by uppercase letters X, Y with elements x_{ij}. Tensors will be indicated by upright bold uppercase letters $\mathbf{X}, \mathbf{Y}$ with elements $x_{i_1,\ldots,i_n}$. For notational convenience, we will often group tensor indices into a vector $\boldsymbol{i} = [i_1, \cdots, i_n]^T$ and write $x_{\boldsymbol{i}}$ instead of $x_{i_1,\ldots,i_n}$. Sets will be denoted by calligraphic letters $\mathcal{S}$ and their cardinality will be denoted by $|\mathcal{S}|$.

2.1 Tensor Product

First, we will review basic properties of the tensor product. The review closely follows the discussions in [4] and [14].

Definition 1 (Tensor Product of Vectors). *The tensor product of vectors $\boldsymbol{x} \in \mathbb{R}^{n_1}$ and $\boldsymbol{y} \in \mathbb{R}^{n_2}$, denoted by $\boldsymbol{x} \otimes \boldsymbol{y}$, is an array with $n_1 n_2$ entries, where*

$$(\boldsymbol{x} \otimes \boldsymbol{y})_{ij} = x_i y_j$$

The defining property of the tensor product of vectors is that $(\boldsymbol{x} \otimes \boldsymbol{y})_{ij} = x_i y_j$. However, since the "shape" of $\boldsymbol{x} \otimes \boldsymbol{y}$ is not defined, there exists a deliberate

ambiguity in how to compute the tensor product of vectors. In particular, for two vectors $\boldsymbol{x}$, $\boldsymbol{y}$, we might obtain one- or two-dimensional arrays with

$$\boldsymbol{x} \otimes \boldsymbol{y} = \begin{bmatrix} x_1\boldsymbol{y}^T & x_2\boldsymbol{y}^T & \dots & x_n\boldsymbol{y}^T \end{bmatrix}^T \in \mathbb{R}^{mn} \tag{1}$$

$$\boldsymbol{x} \otimes \boldsymbol{y} = \boldsymbol{x}\boldsymbol{y}^T \in \mathbb{R}^{m \times n} \tag{2}$$

We will refer to eq. (1) as a vectorized representation of the tensor product, as its results is again a vector, while eq. (2) will be called a *structured* representation. Usually, it will be clear from context which representation is used. The tensor product of vectors is easily extended to more than two vectors, e.g. $(\boldsymbol{x} \otimes \boldsymbol{y} \otimes \boldsymbol{z})_{ijk} = x_i y_j z_k$. In the following, we will denote the tensor product of n vectors also by $\bigotimes_n \boldsymbol{v}_n$. In the structured representation, the tensor product of n vectors corresponds to an n-dimensional array. Furthermore, the tensor product of vectors preserves their linear independence: if the vectors $\{\boldsymbol{x}_1, \dots, \boldsymbol{x}_n\}$ and $\{\boldsymbol{y}_1, \dots, \boldsymbol{y}_m\}$ are, respectively, linearly independent, then the vectors $\{\boldsymbol{x}_i \otimes \boldsymbol{y}_j \mid i = 1 \dots n,\ j = 1 \dots m\}$ are also linearly independent.

Definition 2 (Tensor Product of Vector Spaces). *The tensor product of vector spaces V and W, denoted by $V \otimes W$, is the vector space consisting of all linear combinations $\sum_i a_i \boldsymbol{v}_i \otimes \boldsymbol{w}_i$, where $\boldsymbol{v}_i \in V$ and $\boldsymbol{w}_i \in W$.*

Similarly to the tensor product of vectors, the tensor product of vector spaces is easily extended to more than two vector spaces. In the following, $\bigotimes_n V_n$ will denote the tensor product of n vector spaces. We will refer to a vector space that is the result of tensor products of vector spaces also as a *tensor product space*.

Definition 3 (Tensor). *Let $V = \bigotimes_n W_n$ be a tensor product space with $n \geq 1$. The elements of V are called n-th order tensors.*

Following definition 1 and definition 3, tensors can be interpreted in different ways. One way is as a *vector in a structured vector space*, what corresponds to the vectorized representation in eq. (1). However, according to the structured representation in eq. (2), tensors can also be viewed as *multidimensional arrays*, which is the more commonly used interpretation. Here, we will use both interpretations interchangeably. It also follows immediately that any vector is a first-order tensor and each matrix is a second-order tensor. In the following, $\text{ord}(\mathbf{X})$ will denote the order of a tensor $\mathbf{X}$. For notational convenience, we will also write $\mathbf{X} \in \mathbb{R}^{\prod_i n_i}$ instead of $\mathbf{X} \in \mathbb{R}^{n_1 \times \dots \times n_k}$.

2.2 Structured Data, the Cartesian, and the Tensor Product

To analyze the relation between the order of a tensor and the "structuredness" of data representation we introduce the concept of the *order of structured data*. The general framework in which we will describe structured data is in form of sets of weighted m-tuples, which are defined as follows:

Definition 4 (Set of Weighted m-Tuples). *Let $\mathcal{V} = \mathcal{V}^{(1)} \times \cdots \times \mathcal{V}^{(m)}$ be the Cartesian product over m sets $\mathcal{V}^{(1)}, \ldots, \mathcal{V}^{(m)}$ and let $\phi : \mathcal{E} \mapsto \mathbb{R}$ be a real-valued function that assigns a weight to each m-tuple in $\mathcal{E} \subseteq \mathcal{V}$. A set of weighted m-tuples $\mathcal{T}$ is then defined as a 4-tuple $(\mathcal{V}, \mathcal{E}, \phi, m)$. The* order *of $\mathcal{T}$ is defined as the length of its tuples m. For conciseness, we will refer to sets of weighted m-tuples also as* weighted tuple-sets*.*

Weighted tuple-sets can be interpreted in the following way: The elements of the sets $\mathcal{V}^{(1)}, \ldots, \mathcal{V}^{(m)}$ correspond to the constituents of the structured data. The set $\mathcal{E}$ corresponds to the observed m-tuples, while $\mathcal{V}$ corresponds to all possible m-tuples. For a tuple $t \in \mathcal{E}$, the pair $(t, \phi(t))$ corresponds to an observed data point. This is a very general form of data representation that allows us to consider many forms of structured data. For instance, *dyadic multi-relational* data – as it arises in the Semantic Web or Linked Data – has a natural representation as a weighted tuple-set, where $\mathcal{V}^{(e)}$ is the set of all entities in the data, $\mathcal{V}^{(p)}$ is the set of all predicates, and the weight function $\phi : \mathcal{V}^{(p)} \times \mathcal{V}^{(e)} \times \mathcal{V}^{(e)} \mapsto \{\pm 1\}$ is defined as

$$\phi(p_i, e_j, e_k) = \begin{cases} +1, & \text{if the relationship } p_i(e_j, e_k) \text{ exits} \\ -1, & \text{otherwise} \end{cases}.$$

Similarly, *sequential* or *time-varying* data can be modeled via m-tuples such as (`user, item, last item`) triples for item recommendation [20] or (`person, person, month`) triples in time-varying social networks [2]. In these cases, the function ϕ could model the rating of a product or the interaction of persons. Furthermore, traditional *attribute-value data*, as it is common in many machine learning applications, can be modeled via (`object, attribute`) pairs, which are weighted by the respective attribute values, e.g. $\phi($`Anne,age`$) = 36$.

Tuple-sets can be modeled very naturally using tensors in the following way: Let $\mathcal{T} = (\mathcal{V}, \mathcal{E}, \phi, m)$ be a weighted tuple-set and let $I^{(i)}$ be the standard basis of dimension $|\mathcal{V}^{(i)}|$, such that it indexes all elements of $\mathcal{V}^{(i)}$. $\mathcal{T}$ can then be modeled as a tensor $\mathbf{Y} \in \bigotimes_{i=1}^{m} I^{(i)}$ with entries $y_{i_1,\ldots,i_m} = \phi(v_{i_1}, \ldots, v_{i_m})$ for all observed tuples $(v_{i_1}, \ldots, v_{i_m}) \in \mathcal{E}$. For unobserved tuples $(v_{i_1}, \ldots, v_{i_m}) \in \mathcal{V} \setminus \mathcal{E}$, the corresponding entries in $\mathbf{Y}$ are modeled as missing. Using this construction, each set of objects $\mathcal{V}^{(i)}$ is indexed separately by a mode of the tensor $\mathbf{Y}$. Therefore, it holds that the order of the tensor $\mathbf{Y}$ is identical to the order of the weighted tuple-set $\mathcal{T}$. This enables us to rephrase the question how the structuring of a vector space representation affects the generalization ability of a factorization in terms of the order of weighted tuple-sets and the order of tensors. In particular we are interested in how the generalization ability changes for a tensor representation that has *not* the same order as the underlying weighted tuple-set; compared to a tensor representation that has the identical order.

In this work, we will only consider the problem of learning from sets of binary-weighted tuples, i.e. tuple-sets with weight functions of the form $\phi : \mathcal{E} \mapsto \{\pm 1\}$. This corresponds to a classification setting on binary tensors where $y_{\boldsymbol{i}} \in \{\pm 1\}$ indicates the presence or absence of an m-tuple.

2.3 Tensor Factorizations

Learning via tensor factorizations is based on the idea of explaining an observed tensor $\mathbf{Y}$ through a set of latent factors. The Tucker decomposition is a very general form of factorizing a tensor and allows us to consider different factorization methods within this framework through additional constraints. The Tucker decomposition is defined as

Definition 5 (Tucker Decomposition). *Let* $\mathbf{Y} \in \mathbb{R}^{\prod_i n_i}$ *be an observed tensor with* $\text{ord}(\mathbf{Y}) = m$*. The Tucker decomposition with n-rank* $(r_1, \ldots, r_m)$ *factorizes* $\mathbf{Y}$ *such that each entry of* $\mathbf{Y}$ *is described by the multilinear polynomial*

$$y_{i_1,\ldots,i_m} \approx \sum_{j_1=1}^{r_1} \sum_{j_2=1}^{r_2} \cdots \sum_{j_m=1}^{r_m} w_{j_1,\ldots,j_m} \prod_{k=1}^{m} u^{(k)}_{i_k,j_k} \tag{3}$$

We can now make the connection between the Tucker decomposition of a tensor and weighted tuple-sets as defined in definition 4: the factorization eq. (3) can be interpreted as learning a multilinear function $\gamma : \mathcal{V}^{(1)} \times \cdots \times \mathcal{V}^{(m)} \mapsto \mathbb{R}$ which maps m-tuples to the entries of $\mathbf{Y}$. In contrast to the weight function ϕ of a tuple set, γ is defined over the whole Cartesian product $\mathcal{V}^{(1)} \times \cdots \times \mathcal{V}^{(m)}$.

In the following, it will prove convenient to state eq. (3) in different notations. In tensor notation, eq. (3) is equivalent to

$$\mathbf{Y} \approx \mathbf{X} = \mathbf{W} \times_1 U^{(1)} \times_2 \cdots \times_m U^{(m)} \tag{4}$$

where $\times_k$ denotes the n-mode product of a tensor and a matrix in mode k, while $U^{(k)} \in \mathbb{R}^{n_k \times r_k}$ is the latent factor matrix for mode k and $\mathbf{W} \in \mathbb{R}^{r_1 \times \ldots \times r_m}$ is the core tensor of the factorization. Furthermore, via the *unfolding* operation on tensors and the Kronecker product, eq. (4) can be stated in matrix notation as

$$Y_{(k)} \approx U^{(k)} W_{(k)} \left(U^{(m)} \otimes \cdots \otimes U^{(k+1)} \otimes U^{(k-1)} \otimes \cdots \otimes U^{(1)} \right)^T \tag{5}$$

We will also shorten $\text{n-rank}(\mathbf{Y}) = (r_1, \ldots, r_m)$ to $\text{n-rank}(\mathbf{Y}) = \boldsymbol{r}$. Furthermore, we define some quantities associated with the Tucker decomposition that will prove convenient for the rest of this paper.

Definition 6. *Let* $\mathbf{X} = \mathbf{W} \times_1 U^{(1)} \times_2 \cdots \times_m U^{(m)}$ *with* $\text{n-rank}(\mathbf{X}) = \boldsymbol{r}$*,* $m = \text{ord}(\mathbf{X})$ *and* $\mathbf{X} \in \mathbb{R}^{\prod_i n_i}$*. The number of variables of a Tucker decomposition, i.e. the number of entries in the latent factors, is then given by*

$$\text{var}(\mathbf{X}) = \prod_{i=1}^{m} r_i + \sum_{i=1}^{m} n_i r_i$$

The number of polynomials associated with $\mathbf{X}$*, i.e. the number of entries in* $\mathbf{X}$*, is denoted by*

$$\text{pol}(\mathbf{X}) = \prod_{i=1}^{m} n_i$$

By applying specific constraints on the core tensor or the latent factors, various important factorization methods can be expressed as special cases within the Tucker decomposition framework. One focus of this work is to analyze how these constraints affect the generalization ability of a factorization. In the following, we will briefly discuss some important models to illustrate these constraints: Most matrix factorization methods, can be considered a Tucker decomposition of a second-order tensor. For instance, the *singular value decomposition* can be expressed as a Tucker decomposition of a second order tensor with orthogonal factor matrices. Furthermore, *Candecomp / Parafac* (CP) [10,7] can be described as a Tucker decomposition with the additional constraints that the core tensor $\mathbf{W}$ is superdiagonal and $r_1 = r_2 = \cdots = r_m$. Similarly, the *Block-Term decomposition* (BTD) [8] can be viewed as imposing the constraint that the core tensor $\mathbf{W}$ is block-diagonal. While CP and BTD are decompositions that put special constraints on the core tensor, RESCAL [18] is a factorization that constrains the number of different vector spaces under consideration and is particularly useful for modeling knowledge representations [19]. Specifically, it requires that some of the latent factors are identical, which corresponds to the fact that for some sets $\mathcal{V}^{(i)}$, $\mathcal{V}^{(j)}$ of the underlying tuple-set, it holds that $\mathcal{V}^{(i)} = \mathcal{V}^{(j)}$. Due to space constraints we refer the interested reader to [14] for further details on tensor factorization and the Tucker decomposition on particular.

3 Generalization Bounds for Low-Rank Factorizations

To get deeper theoretical insight into the generalization ability of tensor factorizations, we will now present generalization error bounds. In section 3.1 and section 3.2 we will derive generalization error bounds for the zero-one loss and real-valued loss functions, based on the number of sign patterns that a factorization can express. In these sections, we will closely follow the theory developed in [24,23] and extend it to the general multilinear setting. The actual upper and lower bounds on the number of sign patterns that a tensor factorization can express are then given in section 3.3. To derive these bounds, we will employ properties of the tensor product as discussed in section 2.

Consider the following setting: Let $\mathbf{Y}$ be the tensor representation of structured data $\mathcal{T}$, where a subset of entries $y_{\boldsymbol{i}}$ has been observed and let the set $\Omega = \{\boldsymbol{i} \mid y_{\boldsymbol{i}} \text{ observed}\}$ hold the indices of these observed entries. Then, we seek to predict the missing entries in $\mathbf{Y}$, by computing a factorization such that

$$\mathbf{Y} \approx \mathbf{X} = \mathbf{W} \times_1 U^{(1)} \times_2 \cdots \times_m U^{(m)}.$$

Similar to the matrix case [23], we now seek to bound the true discrepancy between the predicted tensor $\mathbf{X}$ and the target tensor $\mathbf{Y}$ as a function of the discrepancy of the observed entries Ω of $\mathbf{Y}$. The discrepancy of tensors is defined relative to a specific loss function $\Delta(\cdot\,,\cdot)$. The *true discrepancy* of a predicted tensor $\mathbf{X}$ and a target tensor $\mathbf{Y}$ with $\operatorname{ord}(\mathbf{X}) = \operatorname{ord}(\mathbf{Y}) = m$ is defined as

$$\mathcal{D}(\mathbf{X}, \mathbf{Y}) = \frac{1}{\prod_{i=1}^{m} n_i} \sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \cdots \sum_{i_m=1}^{n_m} \Delta(x_{i_1,\ldots i_m}, y_{i_1,\ldots,i_m})$$

while the *empirical discrepancy* is given as

$$\mathcal{D}_\Omega(\mathbf{X}, \mathbf{Y}) = \frac{1}{|\Omega|} \sum_{\boldsymbol{i} \in \Omega} \Delta(x_{\boldsymbol{i}}, y_{\boldsymbol{i}})$$

We restrict the latent tensor $\mathbf{X}$ to the class of *fixed* n-rank tensors of a given order, which will be denoted by

$$\mathcal{X}_{\boldsymbol{r}} \coloneqq \{\mathbf{X} \mid \text{n-rank}(\mathbf{X}) \leq \boldsymbol{r}\}$$

Please note that by restricting the factorization to a Tucker-type decomposition and by fixing n-rank$(\mathbf{X}) = \boldsymbol{r}$, we also fix the quantity var$(\mathbf{X})$, while ord$(\mathbf{X})$ and pol$(\mathbf{X})$ are already determined by the target tensor $\mathbf{Y}$. We now seek to derive PAC-type error bounds of the form

$$\forall \mathbf{Y} \in \mathbb{R}^{\prod \boldsymbol{n}} : \Pr_{\Omega} \left(\forall \mathbf{X} \in \mathcal{X}_{\boldsymbol{r}} : \mathcal{D}(\mathbf{X}, \mathbf{Y}) \leq \mathcal{D}_\Omega(\mathbf{X}, \mathbf{Y}) + \varepsilon \right) > 1 - \delta \tag{6}$$

such that the true discrepancy for all tensors in $\mathcal{X}_{\boldsymbol{r}}$ is bounded by their discrepancy on the observed entries Ω plus a second term ε. An important assumption that will be made is that the set of observed entries Ω is chosen uniformly at random.

3.1 Bounds for Zero-One Sign Agreement Loss

A reasonable choice for $\Delta(\cdot, \cdot)$ in a classification setting is the zero-one loss, i.e.

$$\Delta(a, b) = \begin{cases} 0, & \text{if } \operatorname{sgn}(a) = \operatorname{sgn}(b) \\ 1, & \text{otherwise.} \end{cases}$$

For target entries $y_{\boldsymbol{i}} \in \{\pm 1\}$, the zero-one loss $\Delta(x_{\boldsymbol{i}}, y_{\boldsymbol{i}})$ is independent of the magnitude of the predictions $x_{\boldsymbol{i}}$ and only depends on their sign. A central concept in the following discussion will therefore be the equivalence classes of tensors with identical sign patterns, i.e. the elements of the set

$$\mathcal{S}_{\boldsymbol{n},\boldsymbol{r}} = \left\{ \operatorname{sgn}(\mathbf{X}) \in \{-1, 0, +1\}^{\prod \boldsymbol{n}} \;\middle|\; \mathbf{X} \in \mathbb{R}^{\prod \boldsymbol{n}}, \text{n-rank}(\mathbf{X}) \leq \boldsymbol{r} \right\}.$$

The cardinality $|\mathcal{S}_{\boldsymbol{n},\boldsymbol{r}}|$ specifies therefore, how many different sign patterns can be expressed by factorizations with n-rank$(\mathbf{X}) \leq \boldsymbol{r}$ and pol$(\mathbf{X}) = \prod \boldsymbol{n}$.

Lemma 1. *Let $\mathbf{Y} \in \{\pm 1\}^{\prod \boldsymbol{n}}$ be any binary tensor with $n_i > 2$. Furthermore, let Ω be a set of $|\Omega|$ uniformly chosen entries of $\mathbf{Y}$, let $\delta > 0$, and let $\boldsymbol{r} \in \mathbb{N}_+^{\text{ord}(\mathbf{Y})}$. Then, it holds with probability at least $1 - \delta$ that*

$$\forall \mathbf{X} \in \mathcal{X}_{\boldsymbol{r}} : \mathcal{D}(\mathbf{X}, \mathbf{Y}) < \mathcal{D}_\Omega(\mathbf{X}, \mathbf{Y}) + \sqrt{\frac{\log |\mathcal{S}_{\boldsymbol{n},\boldsymbol{r}}| - \log \delta}{2|\Omega|}}$$

where $|\mathcal{S}_{\boldsymbol{n},\boldsymbol{r}}| \leq \left(\frac{4e \;(\text{ord}(\mathbf{X})+1)\;\; \text{pol}(\mathbf{X})}{\text{var}(\mathbf{X})} \right)^{\text{var}(\mathbf{X})}$

Proof. The following proof is analogue to the matrix case [24], hence we will only provide a brief outline. First, we fix $\mathbf{Y}$ and $\mathbf{X}$. For an index $\boldsymbol{i}$, chosen uniformly at random, it holds that $\Delta(x_{\boldsymbol{i}}, y_{\boldsymbol{i}}) \sim Bernoulli(\mathcal{D}(\mathbf{X}, \mathbf{Y}))$. Consequently, for independently and uniformly chosen observed entries, the sum of Bernoulli distributed random variables $|\Omega|\mathcal{D}_\Omega(\mathbf{X}, \mathbf{Y})$ follows a binomial distribution with mean $|\Omega|\mathcal{D}(\mathbf{X}, \mathbf{Y})$. It follows from Chernoff's inequality that

$$\Pr_{\Omega}\left(\mathcal{D}(\mathbf{X}, \mathbf{Y}) \geq \mathcal{D}_\Omega(\mathbf{X}, \mathbf{Y}) + \varepsilon\right) \leq \exp\left(-2|\Omega|\varepsilon^2\right)$$

Furthermore, since $\Delta(x_{\boldsymbol{i}}, y_{\boldsymbol{i}})$ depends only on the sign of $x_{\boldsymbol{i}}$, the random variable $\mathcal{D}_\Omega(\mathbf{X}, \mathbf{Y})$ is identical for all tensors $\mathbf{X}$ in the same equivalence class of sign patterns. Since there exist $|\mathcal{S}_{\boldsymbol{n},\boldsymbol{r}}|$ different equivalence classes, lemma 1 follows by taking a union bound of the events $\mathcal{D}(\mathbf{X}, \mathbf{Y}) \geq \mathcal{D}_\Omega(\mathbf{X}, \mathbf{Y}) + \varepsilon$ for these random variables. The actual bound on $|\mathcal{S}_{\boldsymbol{n},\boldsymbol{r}}|$ is deferred until section 3.3. □

3.2 Bounds for Real-Valued Loss Functions

Before deriving upper and lower bounds for the number of sign patterns, we also provide a bound for real-valued loss functions, which is the more commonly used setting for tensor factorizations. However, these loss functions, and therefore also their associated discrepancies, are not only determined by the sign of an entry $x_{\boldsymbol{i}}$ but are also determined by the value of this entry. We will therefore derive bounds for the pseudodimension of low-rank tensors.

Lemma 2. *Let $\mathbf{Y} \in \{\pm 1\}^{\boldsymbol{n}}$ be any binary tensor with $n_i > 2$. Furthermore, let $|\Delta(\cdot, \cdot)| \leq b$ be a bounded monotone loss function, let Ω be a set of $|\Omega|$ uniformly chosen entries of $\mathbf{Y}$, let $\delta > 0$, and let $\boldsymbol{r} \in \mathbb{N}_+^{\mathrm{ord}(\mathbf{Y})}$. Then, it holds with probability at least $1 - \delta$*

$$\forall \mathbf{X} \in \mathcal{X}_{\boldsymbol{r}} : \mathcal{D}(\mathbf{X}, \mathbf{Y}) < \mathcal{D}_\Omega(\mathbf{X}, \mathbf{Y}) + \sqrt{32 \frac{\log|\mathcal{S}_{\boldsymbol{n},\boldsymbol{r},\mathbf{T}}| \log \frac{b|\Omega|}{\mathrm{var}(\mathbf{X})} - \log \delta}{|\Omega|}}$$

Proof. Again, the following proof is analogue to the matrix case [24], hence we will outline it only briefly. As mentioned in section 2.3, tensor factorizations can be interpreted as real-valued functions, which map from tuples of indices to entries of the tensor, i.e. a multilinear function $\gamma : \mathcal{I}^{(1)} \times \cdots \times \mathcal{I}^{(n)} \mapsto \mathbb{R}$, where $\mathcal{I}^{(i)}$ indexes the i-th mode. This allows to use the pseudodimension of classes of real-valued functions to obtain similar generalization error bounds as for matrices. The difference to the matrix case is that for tensors the domain of the function ϕ ranges of tuples of fixed length n, while for matrices it ranges over ordered pairs. Therefore, we first bound the pseudodimension of n-rank tensors via the number of sign patterns *relative to a threshold tensor* $\mathbf{T} \in \mathbb{R}^{\prod \boldsymbol{n}}$. The equivalence classes for these relative sign patterns are given by the set

$$\mathcal{S}_{\boldsymbol{n},\boldsymbol{r},\mathbf{T}} = \left\{ \mathrm{sgn}(\mathbf{X} - \mathbf{T}) \in \{-1, 0, +1\}^{\prod \boldsymbol{n}} \;\middle|\; \mathbf{X} \in \mathbb{R}^{\prod \boldsymbol{n}}, \text{n-rank}(\mathbf{X}) \leq \boldsymbol{r} \right\}.$$

The concrete bound for $|S_{\boldsymbol{n},\boldsymbol{r},\mathbf{T}}|$ will be given in section 3.3. Using [23, Theorem 44] we can then obtain the desired bound. □

3.3 Bounds on the Number of Sign Patterns

Following the discussion in section 3.1 and section 3.2, we now seek to bound the number of possible sign patterns $|\mathcal{S}_{n,r}|$ and the number of relative sign patterns $|\mathcal{S}_{\boldsymbol{n},\boldsymbol{r},\mathbf{T}}|$ for tensors $\mathbf{X} \in \mathcal{X}_{\boldsymbol{r}}$. For this purpose, consider the polynomial form of the Tucker decompositions as given in eq. (3). Due to the multilinearity of tensor factorizations, the degree of the polynomial in eq. (3) is equal to $\text{ord}(\mathbf{X}) + 1$. Furthermore, for tensors of fixed size and n-rank, the quantities $\text{pol}(\mathbf{X})$ and $\text{var}(\mathbf{X})$ are also fixed. Using this property of multilinear factorizations, we can bound the number of possible sign patterns of tensors with $\text{n-rank}(\mathbf{X}) = \boldsymbol{r}$ by using their polynomial representation. Following [27] it has been shown, that the number of possible sign patterns for polynomials are bounded by

Theorem 1 ([23, Theorem 34, 35]). *The number of sign patterns of m polynomials, each of degree at most d, over q variables is at most*

$$\left(\frac{4edm}{q}\right)^q$$

for all $m > q > 2$.

By combining the polynomial form of tensor factorizations eq. (3) and theorem 1, we can immediately derive the following lemma which bounds the number of possible sign patterns for n-rank tensors.

Lemma 3 (Upper Bound for Sign Patterns). *The number of possible sign patterns of a m-th order tensor $\mathbf{X} \in \mathbb{R}^{\prod \boldsymbol{n}} = \mathbf{W} \times_1 U^{(1)} \times_2 \cdots \times_m U^{(m)}$ with $\text{n-rank}(\mathbf{X}) = \boldsymbol{r}$ is at most*

$$|\mathcal{S}_{\boldsymbol{n},\boldsymbol{r}}| \leq \left(\frac{4e\ (\text{ord}(\mathbf{X}) + 1)\ \ \text{pol}(\mathbf{X})}{\text{var}(\mathbf{X})}\right)^{\text{var}(\mathbf{X})}$$

for $\text{pol}(\mathbf{X}) > \text{var}(\mathbf{X}) > 2$.

Furthermore, the number of relative sign patterns, i.e. $|\mathcal{S}_{\boldsymbol{n},\boldsymbol{r},\mathbf{T}}|$, can be bounded in the same way, since for

$$y_{i_1,\ldots,i_m} - t_{i_1\ldots,i_m} = \sum_{j_1=1}^{r_1} \sum_{j_2=1}^{r_2} \cdots \sum_{j_m=1}^{r_m} w_{j_1,\ldots,j_m} \prod_{k=1}^{m} u^{(k)}_{i_k j_k} - t_{i_1\ldots,i_m}$$

we have again $\text{pol}(\mathbf{X})$ polynomials of degree $\text{ord}(\mathbf{X}) + 1$ over $\text{var}(\mathbf{X})$ variables.

Next, we provide a lower bound on the number of sign patterns, by interpreting tensor factorization as multiple simultaneous linear classifications.

Lemma 4 (Lower Bound for Sign Patterns). *The number of possible sign patterns of a m-th order tensor $\mathbf{X} \in \mathbb{R}^{\prod \boldsymbol{n}} = \mathbf{W} \times_1 U^{(1)} \times_2 \cdots \times_m U^{(m)}$ with $\text{n-rank}(\mathbf{X}) = \boldsymbol{r}$ is at least*

$$|\mathcal{S}_{\boldsymbol{n},\boldsymbol{r}}| \geq \left(\frac{n_i}{r_i - 1}\right)^{\frac{1}{n_i}(r_i-1)\,\text{pol}(\mathbf{X})}$$

Proof. First, consider the Tucker decomposition in its unfolded variant, i.e.

$$X_{(i)} = U^{(i)} W_{(i)} \left(U^{(m)} \otimes \cdots \otimes U^{(i+1)} \otimes U^{(i-1)} \otimes \cdots \otimes U^{(1)} \right)^T$$

Let $B = U^{(m)} \otimes \cdots \otimes U^{(i+1)} \otimes U^{(i-1)} \otimes \cdots \otimes U^{(1)} \in \mathbb{R}^{\prod \boldsymbol{n}/n_i \times \prod \boldsymbol{r}/r_i}$, and fix $U^{(k)} \in \mathbb{R}^{n_k \times r_k}$ with rows in general position for all $k = 1 \dots m$. We now consider the number of possible sign patterns of matrices $U^{(i)} W_{(i)} B^T$. It follows from the rows being in general position that $\operatorname{rank}\left(U^{(k)}\right) = r_k$ for all $k = 1 \dots m$ [11, Sec. 1.3.2]. Furthermore, since the tensor product preserves the linear independence of vectors, it follows that $\operatorname{span}(B) = \mathbb{R}^{\prod \boldsymbol{r}/r_i}$ [1, Sec. 6.1.4]. Although B is highly structured, it follows that the matrix product $W_{(i)} B^T$ varies over all possible $r_i \times \prod \boldsymbol{n}/n_i$ matrices. Therefore, each column of $\operatorname{sgn}(U^{(i)} W_{(i)} B^T)$ can be considered an independent homogeneous linear classification of n_i vectors in $\mathbb{R}^{r_i}$, for which exactly

$$2 \sum_{k=0}^{r_i - 1} \binom{n_i}{k} > \left(\frac{n_i}{r_i - 1} \right)^{r_i - 1}$$

such classifications exists. Consequently, this many sign patterns exist for each of the $\prod \boldsymbol{n}/n_i = \operatorname{pol}(\mathbf{X})/n_i$ columns of $U^{(i)} W_{(i)} B^T$. □

Next we analyze the tightness of bounds in lemma 3 and lemma 4. Let $m = \operatorname{ord}(\mathbf{X})$, let $\alpha = 4e(m+1)$, let $\forall i : r_{min} \leq r_i$, and similarly let $\forall i : n_{max} \geq n_i$. Then, for $r_{min} \geq \sqrt[m]{\alpha}$ it follows from lemma 3 that

$$|\mathcal{S}_{\boldsymbol{n},\boldsymbol{r}}| \leq \left(\frac{\alpha n_{max}^m}{r_{min}^m} \right)^{\operatorname{var}(\mathbf{X})} \leq \left(\frac{\sqrt[m]{\alpha} n_{max}}{r_{min}} \right)^{m \operatorname{var}(\mathbf{X})} \leq n_{max}^{m \operatorname{var}(\mathbf{X})}$$

Furthermore, for low-rank factorizations with $n_i > r_i^2$ and $\operatorname{pol}(\mathbf{X}) > \frac{m}{r_i - 1} \operatorname{var}(\mathbf{X})$ it follows from lemma 4 that

$$|\mathcal{S}_{\boldsymbol{n},\boldsymbol{r}}| \geq \left(\frac{n_i}{r_i - 1} \right)^{\frac{1}{n_i}(r_i - 1) \operatorname{pol}(\mathbf{X})} \geq \sqrt{n_i}^{\frac{1}{n_i}(r_i - 1) \operatorname{pol}(\mathbf{X})} \geq n_i^{\frac{1}{2n_i} m \operatorname{var}(\mathbf{X})}$$

Hence, the bound is tight up to a multiplicative factor in the exponent.

4 The Effect of Structure and Constraints

In section 3 we derived bounds on the generalization error of tensor factorizations. In this section we discuss what conclusions can be drawn from the derived bounds. In particular, we are interested in how additional structure or constraints affect the generalization ability of tensor factorizations. For this purpose, we will first present a setting in which it is reasonable to compare tensor factorizations of different order. Furthermore, we will evaluate experimentally how the generalization ability of tensor factorizations behaves with the change of structure and constraints. At last, we will discuss how these results can be interpreted with respect to the derived generalization bounds.

4.1 Comparable Tensors

Since it is not reasonable to compare arbitrary tensor factorizations, consider the following setting: Let $\mathcal{T} = (\mathcal{V}, \mathcal{E}, \phi, m)$ be a weighted tuple-set of order m and let $\mathbf{Y}$ be the tensor representation of $\mathcal{T}$. Furthermore, let $\mathbf{Y}^-$ be a tensor representation of $\mathcal{T}$ such that the k-th mode of $\mathbf{Y}^-$ is indexed by the set

$$\mathcal{U}^{(k)} = \begin{cases} \mathcal{V}^{(k)} & , k \neq i \neq j \\ \mathcal{V}^{(i)} \times \mathcal{V}^{(j)} & , k = i. \end{cases}$$

This means that for two index sets $\mathcal{V}^{(i)}$, $\mathcal{V}^{(j)}$ of $\mathcal{T}$ only a single vector space representation is used in $\mathbf{Y}^-$. Consequently, it holds that $\text{ord}(\mathbf{Y}^-) = \text{ord}(\mathbf{Y}) - 1$. This setting corresponds, for example, to propositionalization in multi-relational learning. We will refer to $\mathbf{Y}^-$ as an *understructured representation* of $\mathcal{T}$. The opposite setting would be an *overstructured representation* where the tensor $\mathbf{Y}^-$ is the correct representation of $\mathcal{T}$, while $\mathbf{Y}$ represents one index set $\mathcal{V}^{(i)}$ of $\mathcal{T}$ by two modes, i.e.

$$\mathcal{V}^{(k)} = \begin{cases} \mathcal{U}^{(k)} & , k \neq i \neq j \\ \mathcal{U}^{(i)} \times \mathcal{U}^{(j)} & , k = i \end{cases}$$

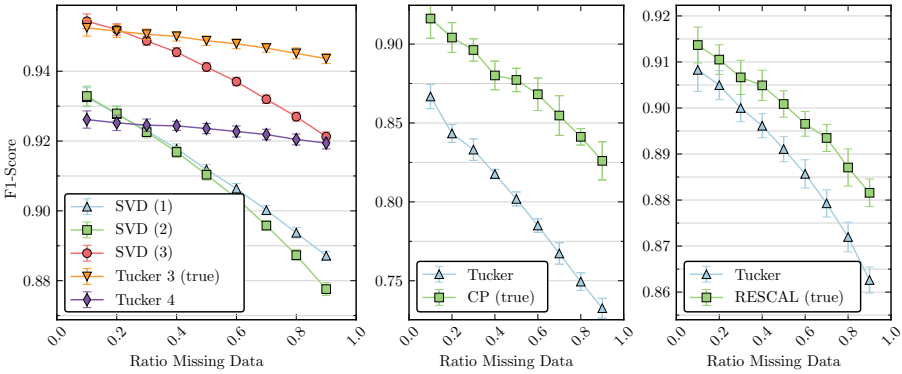
For both, the under- and the overstructured case, we are interested to see how the generalization ability of a tensor factorization changes by factorizing $\mathbf{Y}$ compared to $\mathbf{Y}^-$. Without loss of generalization, let $i = m - 1$, $j = m$ where $m = \text{ord}(\mathbf{Y})$ and $\ell = \text{ord}(\mathbf{Y}^-) = m - 1$. Furthermore, let $\mathbf{X} = \mathbf{W} \times_1 U^{(1)} \times_2 \cdots \times_m U^{(m)} \in \mathbb{R}^{\boldsymbol{n}}$ and $\mathbf{X}^- = \mathbf{W}^- \times_1 U^{(1)-} \times_2 \cdots \times_\ell U^{(\ell)-} \in \mathbb{R}^{\boldsymbol{n}^-}$ be factorizations of $\mathbf{Y}$ and $\mathbf{Y}^-$. Since we are only interested in the effect that the order of data representation has on the generalization ability, we want to exclude the effect of different ranks. Analogously to section 3, we restrict therefore $\mathbf{X}$ and $\mathbf{X}^-$ to be of similar n-rank, in order to get comparable models. Since it holds for the Kronecker product that $\text{rank}(V \otimes W) = \text{rank}(V)\,\text{rank}(W)$, we require that

$$r_k^- = \begin{cases} r_k & , k \neq m \neq \ell \\ r_m r_\ell & , k = \ell \end{cases}$$

It also follows immediately from the construction of $\mathbf{Y}$ and $\mathbf{Y}^-$ and the properties of the Cartesian product that

$$n_k^- = \begin{cases} n_k & , k \neq m \neq \ell \\ n_m n_\ell & , k = \ell \end{cases}$$

In the following, we will refer to tensors $\mathbf{X}$, $\mathbf{X}^-$ who have these properties as *comparable tensors*. Please note that for comparable tensors, it holds that $\text{var}(\mathbf{X}^-) > \text{var}(\mathbf{X})$, since $n_m n_\ell r_m r_\ell > n_m r_m + n_\ell r_\ell$. Furthermore, it holds that $\text{ord}(\mathbf{X}^-) + 1 = \text{ord}(\mathbf{X})$ and $\text{pol}(\mathbf{X}^-) = \text{pol}(\mathbf{X})$.



(a) Third-Order Model (b) CP Model (c) RESCAL Model

Fig. 1. Mean and standard error of the F1-Score over 100 iterations per percentage of missing data. SVD (i) denotes the singular value decomposition of $Y_{(i)}$, i.e. the unfolding of the i-th mode of $\mathbf{Y}$.

4.2 Experimental Results

Given comparable tensors, we evaluated experimentally how tensor factorization behaves under the change of structure and constraints. The experiments were carried out on synthetic data with different amounts of missing data. To evaluate the *effects of structure*, we created a third-order tensor $\mathbf{X} = \mathbf{W} \times_1 A \times_2 B \times_3 C$, where $\mathbf{W} \in \mathbb{R}^{5\times10\times2}$, $A \in \mathbb{R}^{50\times5}$, $B \in \mathbb{R}^{100\times10}$, $C \in \mathbb{R}^{20\times2}$ and where all entries of the core tensor and the factor matrices had been drawn from the standard normal distribution $\mathcal{N}(0, 1)$. From $\mathbf{X}$ we created the target tensor $\mathbf{Y}$ by setting $y_{ijk} = \text{sgn}(x_{ijk})$. Furthermore, the set of observed entries Ω has been drawn uniformly at random, where we increased the ratio of missing entries from $[0.1, 0.9]$. To evaluate the effects of under- and overstructuring, we compared three models: a Tucker-3 decomposition, which is the correct model, the SVD which is an understructured model and a Tucker-4 decomposition, which is an overstructured model. Moreover, the SVD has been computed on all possible unfoldings $Y_{(i)}$, where $i \in \{1, 2, 3\}$. For the Tucker-4 decomposition, we split the second mode of $\mathbf{Y}$ into two size-10 modes, such that $\mathbf{Y}_4 \in \mathbb{R}^{50\times10\times10\times20}$. For each model and each ratio of missing entries we computed 100 factorizations and recorded the F1-score for the classification of the missing entries compared to the ground truth. fig. 1(a) shows the results of these experiments. As expected, the true model provides the best overall performance. One understructured model, i.e. SVD (3), shows comparable results to the true model for low amounts of missing entries but scales significantly worse as the missing data increases. The overstructured model displays the opposite behaviour; it shows reduced overall generalization ability compared to the true model but is more stable with the amount of missing data.

In similar experiments we also evaluated the *effects of constraints*. For this purpose, we created synthetic CP and RESCAL models under similar conditions

as in the previous experiment. However, in this experiment we evaluated how the correct model compared to an unconstrained Tucker model. Figures fig. 1(b) and fig. 1(c) show the results of these experiments. Again, the true models show the best overall performance in both experiments. Furthermore, in both settings, the constrained models scale better with the amount of missing data than the unconstrained tucker model.

4.3 Discussion

The previously derived generalization bounds can provide insight in how to interpret these experimental results. First, note that both terms in eq. (6), i.e. $\mathcal{D}_\Omega(\mathbf{X}, \mathbf{Y})$ and ε, are influenced by the number of sign patterns that a factorization can express. For $\mathcal{D}_\Omega(\mathbf{X}, \mathbf{Y})$ this is the case because the discrepancy will increase when a model $\mathbf{X}$ is not expressive enough to model the sign patterns of a target tensor $\mathbf{Y}$. Furthermore, it has been shown in section 3 that the term ε grows with the number of sign patterns. Since it has also been shown that the upper bound on the number of sign configurations in lemma 3 is tight at least up to a multiplicative factor in the exponent, we consider how this bound changes with the order of the data representation; to see what possible effects the change of structure can have in terms of the generalization ability.

Corollary 1. *For comparable tensors* $\mathbf{X} \in \mathbb{R}^{\boldsymbol{n}}$, $\mathbf{X}^- \in \mathbb{R}^{\boldsymbol{n}^-}$ *with* $\text{ord}(\mathbf{X}) = \text{ord}(\mathbf{X})^- + 1$, $\text{n-rank}(\mathbf{X}) = \boldsymbol{r}$ *and* $\text{n-rank}(\mathbf{X})^- = \boldsymbol{r}^-$, *the ratio of upper bounds on then number of possible sign patterns is at most*

$$1 < \frac{\mathcal{O}(|\mathcal{S}^-_{\boldsymbol{n},\boldsymbol{r}}|)}{\mathcal{O}(|\mathcal{S}_{\boldsymbol{n},\boldsymbol{r}}|)} < \left(\frac{4e\ (\text{ord}(\mathbf{X}^-) + 1)\ \ \text{pol}(\mathbf{X})}{\text{var}(\mathbf{X}^-)} \right)^v$$

where $v = n_m n_\ell r_m r_\ell - (n_\ell r_\ell + n_m r_m) > 0$

Proof. It follows straight from the definition of comparable tensors that $\text{var}(\mathbf{X}^-)$ can be rewritten as $\text{var}(\mathbf{X}^-) = \text{var}(\mathbf{X}) + v$. Furthermore, let

$$\alpha = 4e\ (\text{ord}(\mathbf{X}^-) + 1)\ \ \text{pol}(\mathbf{X})$$
$$\beta = 4e\ (\text{ord}(\mathbf{X}) + 1)\ \ \text{pol}(\mathbf{X}) = \alpha + 4e\ \ \text{pol}(\mathbf{X})$$

Then, it holds that

$$\begin{aligned} \frac{\mathcal{O}(|\mathcal{S}^-_{\boldsymbol{n},\boldsymbol{r}}|)}{\mathcal{O}(|\mathcal{S}_{\boldsymbol{n},\boldsymbol{r}}|)} &= \frac{\alpha^{\text{var}(\mathbf{X})+v}}{\text{var}(\mathbf{X}^-)^{\text{var}(\mathbf{X})+v}} \frac{\text{var}(\mathbf{X})^{\text{var}(\mathbf{X})}}{\beta^{\text{var}(\mathbf{X})}} \\ &= \left(\frac{\alpha}{\text{var}(\mathbf{X}^-)} \right)^v \frac{\alpha^{\text{var}(\mathbf{X})}}{\beta^{\text{var}(\mathbf{X})}} \frac{\text{var}(\mathbf{X})^{\text{var}(\mathbf{X})}}{(\text{var}(\mathbf{X}) + v)^{\text{var}(\mathbf{X})}} \leq \left(\frac{\alpha}{\text{var}(\mathbf{X}^-)} \right)^v \end{aligned}$$

□

The main result of corollary 1 for this discussion is that the bound increases as we decrease the order of the tensor. This suggests that as we increase the

order of the data representation, we will reduce the term ε in eq. (6). As the amount of missing data increases, it is therefore likely to see increasingly severe overfitting for $\mathbf{X}^-$ compared to $\mathbf{X}$. However, when $\mathbf{X}^-$ is the correct and $\mathbf{X}$ is an understructured representation, $\mathcal{O}(|\mathcal{S}^-_{\boldsymbol{n},\boldsymbol{r}}|) > \mathcal{O}(|\mathcal{S}_{\boldsymbol{n}}, \boldsymbol{r}|)$ also suggests that the model $\mathbf{X}$ might not be expressive enough to model the sign patterns of $\mathbf{Y}^-$. This corresponds nicely to the experimental results shown in fig. 1(a). The understructured models are expressive enough to model the sign patterns of $\mathbf{Y}$, as seen in the case of SVD (3). However, they also scale significantly worse than the correct model with the amount of missing data. The overstructured Tucker-4 model scales even better with missing data than the true model, but at the same time gives significantly worse overall results, what suggests that it might not be expressive enough. A possible interpretation is therefore, that the ratio between expressiveness and overfitting is superior for a correct model specification. Since the correct model $\mathbf{X}$ has a much smaller number of variables, it should also be noted that the memory complexity of $\mathbf{X}$ is significantly reduced compared to $\mathbf{X}^-$.

Similar arguments apply for the effect of constraints. Here, the key insight is that both CP-type and RESCAL-type constraints decrease the number of variables in a model. Models like CP or the Block-Term Decompostion, require that $\mathbf{W}$ is superdiagonal or block-superdiagonal and therefore set most entries in the core tensor to $w_{\boldsymbol{i}} = 0$. Models like RESCAL on the other hand, decrease the number of variables through the constraint that some factor matrices $U^{(i)}$, $U^{(j)}$ have to be identical. Since $\mathcal{O}(|\mathcal{S}_{\boldsymbol{n},\boldsymbol{r}}|)$ depends exponentially on $\text{var}(\mathbf{X})$, conclusions similar to the effects of structure can be drawn with regard to the effects of constraints. It suggests that a model with a larger number of variables, i.e. fewer constraints, has more capacity to model sign patterns, but at the same time is more likely to overfit as the amount of missing data increases. Again, this corresponds nicely to the experimental results in fig. 1(b) and fig. 1(c).

5 Related Work

We are not aware of any previous generalization error bounds for tensor factorizations or of any theoretical results that relate the order of a tensor and the order of structured data to the generalization ability of factorizations. Our derivation of error bounds for the tensor case builds strongly on the work of [24,23], which provided error bounds for matrix factorizations with zero-one loss and general loss functions. [28] derived similar bounds in the context of rank-k SVMs. For general matrices, [6,5] show that under suitable conditions a low-rank matrix can be recovered from a minimal set of entries via convex optimization and also provide theoretical bounds. [9,26] extends these methods to tensor completion, although without providing error bounds. It has also been shown experimentally that by adding structure to the vector space representations via the tensor product, the amount of data needed for exact recovery can be greatly reduced [26,25].

6 Conclusion

To obtain a deeper understanding of the generalization ability of tensor factorizations, we derived generalization error bounds based on the number of sign patterns that a tensor factorization can model. Using a general framework to describe structured data based on weighted tuple-sets, we analyzed how tensor factorizations behave when their order does not match the true order of the data. We showed experimentally that structuring vector space representations via the tensor product, up to the true order of the data, adds important information such that tensor models often scale better with sparsity or missing data than their understructured counterparts. We also discussed analytically how this behaviour can be explained in the light of the newly derived generalization bounds. In this work, we only considered binary values for the target tensor $\mathbf{Y}$, which corresponds to a classification setting. For future work, it would prove very valuable to also derive error bounds for the more general case of real-valued weight functions. Since the current error bounds are based on the assumption that the observed entries are independently and identically distributed what – especially on structured data – might not hold, it might also be useful to consider techniques as in [17], to overcome this limitation.

References

1. Anthony, M., Harvey, M.: Linear Algebra: Concepts and Methods. Cambridge University Press (2012)
2. Bader, B.W., Harshman, R.A., Kolda, T.G.: Temporal analysis of semantic graphs using ASALSAN. In: Seventh IEEE International Conference on Data Mining (ICDM 2007), Omaha, NE, USA, pp. 33–42 (2007)
3. Bordes, A., Weston, J., Collobert, R., Bengio, Y.: Learning structured embeddings of knowledge bases. In: Proceedings of the 25th Conference on Artificial Intelligence, San Francisco, USA (2011)
4. Burdick, D.S.: An introduction to tensor products with applications to multiway data analysis. Chemometrics and Intelligent Laboratory Systems 28(2), 229–237 (1995)
5. Candes, E.J., Plan, Y.: Matrix completion with noise. Proceedings of the IEEE 98(6), 925–936 (2010)
6. Candès, E.J., Recht, B.: Exact matrix completion via convex optimization. Foundations of Computational Mathematics 9(6), 717–772 (2009)
7. Carroll, J.D., Chang, J.J.: Analysis of individual differences in multidimensional scaling via an n-way generalization of Eckart-Young decomposition. Psychometrika 35(3), 283–319 (1970)
8. De Lathauwer, L.: Decompositions of a higher-order tensor in block termsPart II: definitions and uniqueness. SIAM Journal on Matrix Analysis and Applications 30(3), 1033–1066 (2008)
9. Gandy, S., Recht, B., Yamada, I.: Tensor completion and low-n-rank tensor recovery via convex optimization. Inverse Problems 27(2), 025010 (2011)
10. Harshman, R.A., Lundy, M.E.: PARAFAC: parallel factor analysis. Computational Statistics & Data Analysis 18(1), 39–72 (1994)

11. Hassoun, M.H.: Fundamentals of Artificial Neural Networks. MIT Press (1995)
12. Huang, Y., Tresp, V., Bundschus, M., Rettinger, A., Kriegel, H.-P.: Multivariate prediction for learning on the semantic web. In: Frasconi, P., Lisi, F.A. (eds.) ILP 2010. LNCS, vol. 6489, pp. 92–104. Springer, Heidelberg (2011)
13. Jenatton, R., Le Roux, N., Bordes, A., Obozinski, G.: A latent factor model for highly multi-relational data. In: Advances in Neural Information Processing Systems, vol. 25, pp. 3176–3184. MIT Press, Lake Tahoe (2012)
14. Kolda, T.G., Bader, B.W.: Tensor decompositions and applications. SIAM Review 51(3), 455–500 (2009)
15. Kramer, S., Lavrac, N., Flach, P.: Propositionalization approaches to relational data mining. Springer-Verlag New York, Inc. (2001)
16. Liu, J., Musialski, P., Wonka, P., Ye, J.: Tensor completion for estimating missing values in visual data. In: 2009 IEEE 12th International Conference on Computer Vision, pp. 2114–2121 (2009)
17. Mohri, M., Rostamizadeh, A.: Rademacher complexity bounds for non-iid processes. In: Advances in Neural Information Processing Systems, vol. 21, pp. 1097–1104. MIT Press, Cambridge (2009)
18. Nickel, M., Tresp, V., Kriegel, H.P.: A three-way model for collective learning on multi-relational data. In: Proceedings of the 28th International Conference on Machine Learning, pp. 809–816. ACM, Bellevue (2011)
19. Nickel, M., Tresp, V., Kriegel, H.P.: Factorizing YAGO: scalable machine learning for linked data. In: Proceedings of the 21st International Conference on World Wide Web, pp. 271–280. ACM, New York (2012)
20. Rendle, S., Freudenthaler, C., Schmidt-Thieme, L.: Factorizing personalized markov chains for next-basket recommendation. In: Proceedings of the 19th International Conference on World Wide Web, pp. 811–820. ACM (2010)
21. Rettinger, A., Wermser, H., Huang, Y., Tresp, V.: Context-aware tensor decomposition for relation prediction in social networks. Social Network Analysis and Mining 2(4), 373–385 (2012)
22. Signoretto, M., Van de Plas, R., De Moor, B., Suykens, J.A.: Tensor versus matrix completion: a comparison with application to spectral data. IEEE Signal Processing Letters 18(7), 403–406 (2011)
23. Srebro, N.: Learning with matrix factorizations. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, USA (2004)
24. Srebro, N., Alon, N., Jaakkola, T.S.: Generalization error bounds for collaborative prediction with low-rank matrices. In: Advances in Neural Information Processing Systems, vol. 17, pp. 1321–1328. MIT Press, Cambridge (2005)
25. Tomioka, R., Hayashi, K., Kashima, H.: Estimation of low-rank tensors via convex optimization. arXiv preprint arXiv:1010.0789 (2010)
26. Tomioka, R., Suzuki, T., Hayashi, K., Kashima, H.: Statistical performance of convex tensor decomposition. In: Advances in Neural Information Processing Systems, vol. 24, pp. 972–980 (2012)
27. Warren, H.E.: Lower bounds for approximation by nonlinear manifolds. Transactions of the American Mathematical Society 133(1), 167–178 (1968)
28. Wolf, L., Jhuang, H., Hazan, T.: Modeling appearances with low-rank SVM. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–6 (2007)