Nonlinear System Identification via Tensor Completion

Nikos Kargas,[1] **Nicholas D. Sidiropoulos**[2]
[1]Department of ECE, University of Minnesota, MN, USA
[2]Department of ECE, University of Virginia, VA, USA
karga005@umn.edu, nikos@virginia.edu

Abstract

Function approximation from input and output data pairs constitutes a fundamental problem in supervised learning. Deep neural networks are currently the most popular method for learning to mimic the input-output relationship of a general nonlinear system, as they have proven to be very effective in approximating complex highly nonlinear functions. In this work, we show that identifying a general nonlinear function $y = f(x_1, \ldots, x_N)$ from input-output examples can be formulated as a tensor completion problem and under certain conditions provably correct nonlinear system identification is possible. Specifically, we model the interactions between the N input variables and the scalar output of a system by a single N-way tensor, and setup a weighted low-rank tensor completion problem with smoothness regularization which we tackle using a block coordinate descent algorithm. We extend our method to the multi-output setting and the case of partially observed data, which cannot be readily handled by neural networks. Finally, we demonstrate the effectiveness of the approach using several regression tasks including some standard benchmarks and a challenging student grade prediction task.

The problem of identifying a nonlinear function $y = f(x_1, \ldots, x_N)$ from input-output examples is of paramount importance in machine learning, dynamical system identification and control, communications, and many other disciplines. In machine learning in particular, most of the supervised learning tasks are nonlinear system identification problems. For example, binary/multiclass classification, where the goal is to predict a discrete variable denoting the class label of each realization, and regression/prediction, where the goal is to predict real or complex valued variables. Algorithmic advancements, availability of vast amounts of data and increasing computational power have led to the development of state-of-the-art prediction models with unprecedented success in various domains such as image classification, speech recognition, and language processing. Kernel methods, random forests, neural networks and deep learning are powerful classes of machine learning models that can learn highly nonlinear functions and have been successfully applied in many supervised machine learning tasks (Hastie, Tibshirani, and Friedman 2009). Each of the aforementioned methods can be well suited for a particular problem, but may perform badly for another. In general it is seldom known in advance which method will perform best for any given problem.

This paper presents a simple and elegant alternative for nonlinear system identification based on low-rank tensor decomposition. Tensor decomposition is a powerful tool for analyzing multi-way data and has had major successes in applications spanning machine learning, statistics, signal processing and data mining (Sidiropoulos et al. 2017). The Canonical Polyadic Decomposition (CPD) model is one of the most popular tensor models mainly due to its simplicity and its uniqueness properties. The CPD model has been applied in various machine learning applications, including recommender systems to model time-evolving relational data (Xiong et al. 2010), community detection and clustering to model user interactions across different networks (Papalexakis, Akoglu, and Ience 2013), knowledge base completion and link prediction for discovering unobserved subject-object interactions (Lacroix, Usunier, and Obozinski 2018) and in latent variable models for parameter identification (Anandkumar et al. 2014). These works deal with relatively low-order tensors; however, high-order tensors also arise in practical scenarios – e.g., a joint probability mass function of N categorical random variables can be naturally regarded as an N-th order tensor and modeled using a CPD model (Kargas, Sidiropoulos, and Fu 2018).

In this work, we show that the CPD model offers an appealing solution for modeling and learning a *general* nonlinear system using a single high-order tensor. Tensors have been used to model *low-order multivariate polynomial* systems: a multivariate polynomial of order d is represented by a tensor of order d – e.g., a second-order polynomial is represented by a quadratic form involving a single matrix (Rendle 2010). However, such an approach requires prior knowledge of polynomial order, and assuming that one deals with a polynomial of a given degree can be highly restrictive in practice. Instead, what we advocate here is a simple and general approach: a nonlinear system having N discrete inputs and a single output can be naturally repre-

Dedicated to John S. Baras.

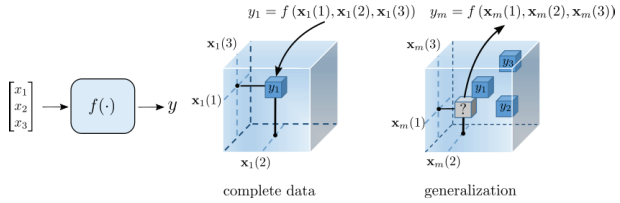

Figure 1: Nonlinear system identification as tensor completion.

sented as an N-way tensor where the tuple of input variables $[\mathbf{x}_m(1), \ldots, \mathbf{x}_m(N)]$ can be viewed as a cell multi-index and the cell content is the response of the system y_m. Given a new data point, the corresponding tensor cell is queried and the output is used as the predictor. Note that only a small fraction of the tensor entries are observed during training, and we are ultimately interested in answering queries for unobserved data points (Figure 1). This motivates the use of low-rank tensor models as a tool for capturing interactions between the predictors and imputing the missing data.

Both experimental (Tomasi and Bro 2005) and theoretical studies (Krishnamurthy and Singh 2013; Jain and Oh 2014; Sorensen and De Lathauwer 2019) have shown that exact tensor completion from limited samples is possible under certain conditions. The implication of our simple but profound modeling idea is very compelling, since:

- The CPD can model any nonlinearity (even of ∞ order) for high-enough rank – because every tensor admits a CPD of bounded rank; see (Sidiropoulos et al. 2017) and references therein. Even for low ranks, it can model highly nonlinear operators such as products or sums of the signs of the input variables.
- Provably correct nonlinear system identification is possible from limited samples. If the associated tensor describing the nonlinear operator is low rank, then it can be fully identified.
- In practice, tensors corresponding to real-world systems may not be low-rank; nevertheless even if a system is not exactly low rank, our approach will identify the principal components of the unknown nonlinear mapping, in a sense that will be clarified in the sequel.

Even though tensor recovery can be guaranteed under a low-rank assumption, tensor decomposition can often benefit from additional knowledge regarding the application by incorporating constraints such as non-negativity, sparsity or smoothness (Sidiropoulos et al. 2017). In our present context, smoothness is a desirable property for applications where we expect that small perturbations in the input will most probably cause small changes in the output of the system. Therefore, we propose augmenting the CPD tensor completion problem with smoothness regularization on the ordinal latent factors.

Contributions: We model a general nonlinear system using a single high-order tensor admitting a CPD model. Specifically, we formulate the problem as a smooth tensor decomposition problem with missing data. Although our method is naturally suited to handle discrete features, it can also be used for continuous valued features (Kargas and Sidiropoulos 2019) and be enhanced using ensemble techniques. Additionally, leveraging the structure of the CPD model, we propose a simple yet effective approach to handle randomly missing input variables. Finally, we discuss how the approach can be extended to vector valued function prediction. The proposed approach requires little parameter tuning, and can model complex nonlinear functions. We propose an easy to implement Block Coordinate Descent (BCD) algorithm and demonstrate the performance in UCI machine learning datasets against competitive baselines as well as a challenging grade prediction task, using real student grade data.

Notation and Background

We use the symbols x, $\mathbf{x}$, $\mathbf{X}$, $\mathcal{X}$ for scalars, vectors, matrices, and tensors respectively. We use the notation $\mathbf{x}(n)$, $\mathbf{X}(:, n)$, $\mathcal{X}(:, :, n)$ to refer to a particular element of a vector, a column of a matrix and a slab of a tensor. Symbols $\circ$, $\otimes$, $\circledast$, $\odot$ denote the outer, Kronecker, Hadamard and Khatri-Rao (column-wise Kronecker) product respectively.

An N-way tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is a multi-dimensional array whose entries are indexed by N coordinates. A polyadic decomposition expresses $\mathcal{X}$ as a sum of rank-1 components $\mathcal{X} = \sum_{f=1}^{F} \mathbf{a}_f^1 \circ \mathbf{a}_f^2 \circ \cdots \circ \mathbf{a}_f^N$, where $\mathbf{a}_f^n \in \mathbb{R}^{I_n}$. If the number of rank-1 components is minimal then the decomposition is called the CPD of $\mathcal{X}$ and F is called the rank of $\mathcal{X}$ (Sidiropoulos et al. 2017). By defining factor matrices $\mathbf{A}_n = [\mathbf{a}_1^n \cdots \mathbf{a}_F^n] \in \mathbb{R}^{I_n \times F}$, the elements of the tensor $\mathcal{X}$ can be expressed as

$$\mathcal{X}(i_1, \ldots, i_N) = \sum_{f=1}^{F} \prod_{n=1}^{N} \mathbf{A}_n(i_n, f). \tag{1}$$

We adopt the common notation $\mathcal{X} = [\![\mathbf{A}_1, \ldots, \mathbf{A}_N]\!]_F$ to denote the tensor synthesized from the CPD model using these factors. The mode-n fibers of a tensor are the vectors obtained by fixing all the indices except for the n-th index. We can represent tensor $\mathcal{X}$ using a matrix $\mathcal{X}^{(n)} \in \mathbb{R}^{I_1 \cdots I_{n-1} I_{n+1} \cdots I_N \times I_n}$ called mode-n matricization obtained by arranging the mode-n fibers of the tensor as columns of the resulting matrix

$$\mathcal{X}^{(n)} = \big(\odot_{k \neq n} \mathbf{A}_k\big) \mathbf{A}_n^T, \tag{2}$$

where $\bigodot_{k \neq n} \mathbf{A}_k = \mathbf{A}_N \odot \cdots \odot \mathbf{A}_{n+1} \odot \mathbf{A}_{n-1} \odot \cdots \odot \mathbf{A}_1$.

The n-mode product of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \cdots \times I_N}$ with a matrix $\mathbf{U} \in \mathbb{R}^{J \times I_n}$ is denoted by $\mathcal{X} \times_n \mathbf{U}$ and an entry of the resulting tensor is given by

$$\begin{aligned}(\mathcal{X} \times_n \mathbf{U})(i_1, \ldots, i_{n-1}, j, i_{n+1}, \ldots, i_N) \\ = \sum_{i_n} \mathcal{X}(i_1, \ldots, i_N) \mathbf{U}(j, i_n).\end{aligned} \tag{3}$$

Furthermore, assuming that a tensor $\mathcal{X}$ admits a CPD with rank F, the n-mode product can be expressed as

$$\begin{aligned}&[\![\mathbf{A}_1, \ldots, \mathbf{A}_N]\!]_F \times_n \mathbf{U} \\ &= [\![\mathbf{A}_1, \ldots, \mathbf{A}_{n-1}, \mathbf{U}\mathbf{A}_n, \mathbf{A}_{n+1} \ldots, \mathbf{A}_N]\!]_F.\end{aligned} \tag{4}$$

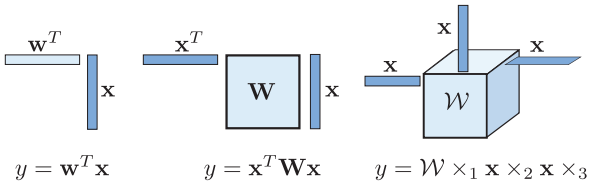

Figure 2: Traditional approaches.

CPD is a powerful tool for data analysis mainly due to its uniqueness properties. For a tensor $\mathcal{X}$ of rank F, we say that a decomposition $\mathcal{X} = [\![\mathbf{A}_1, \ldots, \mathbf{A}_N]\!]_F$ is unique if the factors are unique up to a common permutation and scaling / counter-scaling of columns. Specifically, if there exists another decomposition $\mathcal{X} = [\![\widehat{\mathbf{A}}_1, \ldots, \widehat{\mathbf{A}}_N]\!]_F$, then, there exists a permutation matrix $\mathbf{\Pi}$ and diagonal scaling matrices $\mathbf{\Lambda}_1, \ldots, \mathbf{\Lambda}_N$ such that $\widehat{\mathbf{A}}_n = \mathbf{A}_n\mathbf{\Pi}\mathbf{\Lambda}_n, \forall n \in [N]$ and $\mathbf{\Lambda}_1 \cdots \mathbf{\Lambda}_N = \mathbf{I}$. Tensor decomposition is unique under mild rank conditions; see (Sidiropoulos et al. 2017) and references therein. In our context, uniqueness is a desirable property since it is necessary for model interpretability.

Related Work

Tensors have been mostly used to model low-order multivariate polynomial systems. A multivariate polynomial of degree (order) d can be represented by a tensor of order d. For example, a second-order polynomial is represented by a quadratic form involving a single matrix i.e., $f(\mathbf{x}) = \mathbf{x}^T\mathbf{W}\mathbf{x}$ while a third order polynomial is represented using a 3-way tensor i.e., $f(\mathbf{x}) = \mathcal{W} \times_1 \mathbf{x} \times_2 \mathbf{x} \times \mathbf{x}_3$ (Figure 2). The number of parameters grows exponentially with the order of the approximation making this approach computationally demanding. One way to reduce the number of parameters is to assume that the coefficient tensor is low-rank.

Polynomial Networks (PN) and Factorization Machines (FM) utilize mainly third-order CPD models in order to parameterize the polynomial coefficients with applications in recommender systems and link prediction (Rendle 2010; Blondel et al. 2016a; 2016b). Such approaches require prior knowledge of polynomial order, and assuming that one deals with a polynomial of a given degree can be restrictive. Additionally, even when dealing with the simplest possible approximation model which is rank-1, the number of parameters grows linearly with d meaning that this approach cannot model high-degree polynomial functions. Similarly, another tensor model, known as the Tucker model, has also been used for parametrization of polynomial functions in a chemogenomics data prediction task (Perros et al. 2017). Finally, a tensor train model (Oseledets 2011) has also been used in multivariate polynomial regression (Novikov, Trofimov, and Oseledets 2016). Unlike CPD, the model parameters of Tucker and tensor train models are not identifiable.

Output-only ('blind') identification of *linear* systems has also been considered from a tensor point of view. Specifically, identification of Finite Impulse Response (FIR) systems *using only output examples* has been shown in (Boussé, Debals, and De Lathauwer 2017; Van Eeghem et al. 2018).

Our work is radically different from existing approaches. We model a general nonlinear system using a single tensor of order equal to the number of inputs and propose using a high-order tensor completion approach for system identification. One of the earliest applications of tensor decomposition with smooth latent factors has been fluorescence data analysis (Bro 1998; Fu et al. 2015). Recently, it is has been mostly proposed in the area of image processing. Specifically, CPD and Tucker models with smoothness constraints or regularization have been used for the recovery of incomplete 3- and 4-dimensional image data (Yokota, Zhao, and Cichocki 2016; Imaizumi and Hayashi 2017). To the best of our knowledge, tensor completion (with or without smooth latent factors) has not been considered yet as a tool for general nonlinear system identification.

Proposed Approach

Canonical System Identification (CSID)

We are given a training dataset of M input-output pairs $D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_M, y_M)\}$. Let us assume that all predictors are discrete and take values from a common alphabet $\mathcal{I} = \{1, \ldots, I\}$. The scalar output y_m is a nonlinear function of the input $\mathbf{x}_m$ distorted by some unknown noise $y_m = f(\mathbf{x}_m(1), \ldots, \mathbf{x}_m(N)) + \epsilon_m$. The nonlinear function $f : \{1, \ldots, I\}^N \to \mathbb{R}$ can be modeled as an N-way tensor $\mathcal{X}$ where each input vector $[\mathbf{x}_m(1), \ldots, \mathbf{x}_m(N)]$ can be viewed as a cell multi-index and the cell content is the estimated response of the system $\widehat{y}_m$. We are interested in building a model that minimizes the Mean Square Error (MSE) between the model predictions and the actual response. However, it is evident that it is impossible to infer the response of unobserved data without any assumptions on $\mathcal{X}$. To alleviate this problem we aim for the principal components of the nonlinear operator by minimizing the tensor rank. Assuming a low-rank CPD model, the problem of finding the rank-F approximation which best fits our data can be formulated as

$$\begin{aligned} \min_{\mathcal{X}, \{\mathbf{A}_n\}_{n=1}^N} \; & \frac{1}{M} \sum_{m=1}^{M} \left(y_m - \mathcal{X}(\mathbf{x}_m(1), \ldots, \mathbf{x}_m(N))\right)^2 \\ & + \sum_{n=1}^{N} \rho \|\mathbf{A}_n\|_F^2 \\ \text{s.t.} \quad & \mathcal{X} = \sum_{f=1}^{F} \mathbf{A}_1(:, f) \odot \cdots \odot \mathbf{A}_N(:, f), \end{aligned} \tag{5}$$

where ρ is a regularization parameter. It is convenient to express the problem in the following equivalent form

$$\begin{aligned} \min_{\mathcal{X}, \{\mathbf{A}_n\}_{n=1}^N} \; & \frac{1}{M} \left\| \sqrt{\mathcal{W}} \circledast (\mathcal{Y} - \mathcal{X}) \right\|_F^2 + \sum_{n=1}^{N} \rho \|\mathbf{A}_n\|_F^2 \\ \text{subject to} \quad & \mathcal{X} = \sum_{f=1}^{F} \mathbf{A}_1(:, f) \odot \cdots \odot \mathbf{A}_N(:, f), \end{aligned} \tag{6}$$

where $\mathcal{W}$ is a tensor containing the number of times a particular data point $\mathbf{x} = [i_1, \ldots, i_n]^T$ appears in the dataset

and $\mathcal{Y}$ is a tensor containing the mean response of the corresponding data points. The equivalence between Problems (5), (6) is straightforward

$$\begin{aligned}
&\min \sum_{m=1}^{M} \Big(y_m - \mathcal{X}(x_m(1), \ldots, x_m(N))\Big)^2 \Leftrightarrow \\
&\min \sum_{i_1,\ldots,i_N} \sum_{m' \in S_{i_1,\ldots,i_N}} \big(y_m - \mathcal{X}(i_1, \ldots, i_N)\big)^2 \Leftrightarrow \\
&\min \sum_{i_1,\ldots,i_N} \sum_{m' \in S_{i_1,\ldots,i_N}} \big(y_m - \mathcal{Y}(i_1, \ldots, i_N) \\
&\qquad + \mathcal{Y}(i_1, \ldots, i_N) - \mathcal{X}(i_1, \ldots, i_N)\big)^2 \Leftrightarrow \\
&\min \sum_{i_1,\ldots,i_N} \mathcal{W}(i_1, \ldots, i_N)\big(\mathcal{Y}(i_1, \ldots, i_N) - \mathcal{X}(i_1, \ldots, i_N)\big)^2.
\end{aligned}$$

The set $S_{i_1,\ldots,i_N}$ contains the indices the data point $[i_1, \ldots, i_N]^T$ appears in the dataset. Oftentimes, datasets contain both categorical and ordinal predictors, the later, being either discrete or continuous. In the presence of ordinal predictors a desirable property of a regression model is having smooth prediction surfaces i.e., small variations in the input will cause small changes in the output. As an example consider the task of estimating students' grades in future courses based on their grades in past courses, an important topic in educational data mining as it can facilitate the creation of personalized degree paths which will potentially lead to timely graduation (Polyzou and Karypis 2016). The predictors correspond to the grades in N past courses that a student has received and the the predicted response is the student's grade in a future course. We are interested in building a model that maps an N-dimensional discrete feature vector to the output response. Adding a smoothness constraint or regularization will guarantee that the model will produce similar outputs for two students that differ slightly in their past grades as they are likely to perform similarly in the future. Therefore, we propose augmenting the CPD tensor completion problem with smoothness regularization:

$$\begin{aligned}
\min_{\mathcal{X}, \{\mathbf{A}_n\}_{n=1}^N} \; & \frac{1}{M} \Big\| \sqrt{\mathcal{W}} \circledast (\mathcal{Y} - \mathcal{X}) \Big\|_F^2 + \sum_{n=1}^{N} \rho \|\mathbf{A}_n\|_F^2 \\
& + \sum_{n=1}^{N} \mu_n \|\mathbf{T}_n \mathbf{A}_n\|_F^2 \qquad (7) \\
\text{s.t.} \quad & \mathcal{X} = \sum_{f=1}^{F} \mathbf{A}_1(:, f) \odot \cdots \odot \mathbf{A}_N(:, f),
\end{aligned}$$

where the matrix $\mathbf{T}_n$ is a smoothness promoting matrix typically defined as $\mathbf{T}_n \in \mathbb{R}^{(I_n - 1) \times I_n}$ with $\mathbf{T}_n(i, i) = 1$ and $\mathbf{T}_n(i, i{+}1) = -1$ or $\mathbf{T}_n \in \mathbb{R}^{(I_n - 2) \times I_n}$ with $\mathbf{T}_n(i, i) = -1$, $\mathbf{T}_n(i, i + 1) = 2$ and $\mathbf{T}_n(i, i + 2) = -1$. We set $\mu_n = 0$ for categorical predictors and $\mu_n > 0$ otherwise. Penalizing the difference of consecutive row elements of a factor $\mathbf{A}_n$ guarantees that varying the n-th dimension and keeping the remaining fixed will have a small impact on the predicted response. Another appealing feature of the proposed smoothness regularization is that it can potentially measure feature importance. Note that the effect a variable will have in the prediction is minimized if each column of the corresponding factor is a constant number. Irrelevant features are more likely to have factors that vary slightly. On the contrary, factors associated with predictive features will have more variations and induce a larger penalty cost.

Remark: CPD can model any nonlinear operator for high-enough rank, but even for low ranks, it can model highly nonlinear operators such as

$$\begin{aligned}
f_1(x_1, \ldots, x_N) &= \prod_{n=1}^{N} \operatorname{sign}(x_n), \\
f_2(x_1, \ldots, x_N) &= \sum_{n=1}^{N} \operatorname{sign}(x_n).
\end{aligned}$$

Comparing these equations with Equation (1) we can verify that the former corresponds to a rank-1 CPD model, while the later to rank N.

Tensor Completion: Identifiability

In this section we briefly review existing probabilistic and deterministic theoretical results on tensor recovery from a few samples. This is important because, using our approach of casting system identification as tensor completion, the results below directly yield new results on nonlinear multivariate system identification - even for systems of unbounded nonlinearity degree. Recovering a tensor from samples depends mainly on how the samples $(X_1, \ldots, X_N)$ are generated – randomly or systematically, and if randomly from what distribution – as well as the operational $\mathbf{A}_n$. Practical experience suggests that the generic sample complexity for randomly drawn point samples is proportional to the degrees of freedom $O(FNI)$ in the model. This has been proven for randomly drawn *linear* (generalized, aggregated) samples, but not yet for point samples (Bousse et al. 2018).

An adaptive sampling method with an estimation algorithm has been proposed by (Krishnamurthy and Singh 2013) that provably recovers an N-th order rank-F tensor using $O(IF^{N-0,5}\mu^{N-1}N \log F)$ samples where μ is a coherence bound on the factor matrices. Later, (Jain and Oh 2014) proposed a method that can recover an N-th order rank-F tensor with orthogonal factor matrices and random sampling using $O(I^{N/2}\mu^6 F^5 \log^4 I)$ samples. A necessary condition for these methods is that the rank needs to be less than the maximum outer dimension although a CPD model can be unique even if rank exceeds this bound. Probabilistic results on tensor completion have also been proven for incomplete tensors that have low mode-n ranks under certain incoherence conditions, relying on minimization of the sum of nuclear norms of the tensor unfoldings (Gandy, Recht, and Yamada 2011). For $F < I$, the result in (Yuan and Zhang 2016) can be used to show that for uniform random point samples, the sample complexity for our low-rank model is $O(\sqrt{F} I^{N/2} \log I)$.

Deterministic conditions based on a specific sampling strategy, namely fiber sampling, have been given by (Sorensen and De Lathauwer 2019). Necessary and sufficient conditions are provided which are dependent on the

sampling pattern, assuming that the rank is low enough. The authors propose an eigenvalue decomposition algorithm and demonstrate exact recovery of low-rank incomplete tensors even when, less than one percent of the tensor entries are available. The authors have also extended the results in the case of not fully observed fibers. Finally, several *regular* sampling strategies are investigated and generic identifiability conditions are provided by (Kanatsoulis et al. 2019).

Algorithm

The work-horse of tensor decomposition is the so-called Alternating Least Squares (ALS) algorithm. ALS is a special type of BCD which offers two distinct advantages: monotonic decrease of the cost function, and no need for parameter tuning. In this section, we propose an ALS approach to tackle Problem 7.

Tensors $\mathcal{W}, \mathcal{Y}$ despite being high-dimensional, are in general very sparse and optimized sparse tensor formats can offer huge memory and computational savings (Smith and Karypis 2015). The idea of ALS is that we cyclically update variables $\{\mathbf{A}_n\}_{n=1}^N$ while fixing the remaining variables at their last updated values. Assume that we fix estimates $\mathbf{A}_n$, $\forall n \in [N] \setminus \{k\}$ we need to solve the following optimization problem

$$\begin{aligned}\min_{\mathbf{A}_k} \big\| \widehat{\mathcal{W}}^{(k)} \circledast \mathcal{Y}^{(k)} - \widehat{\mathcal{W}}^{(k)} \circledast (\mathbf{Q}_k \mathbf{A}_k^T) \big\|_F^2 \\ + \rho \|\mathbf{A}_k\|_F^2 + \mu_k \|\mathbf{T}_k \mathbf{A}_k\|_F^2,\end{aligned} \tag{8}$$

where $\mathbf{Q}_k = (\odot_{n \neq k} \mathbf{A}_n)$, $\widehat{\mathcal{W}} = \sqrt{\mathcal{W}}$ with the square root computed element-wise. Equivalently, we have

$$\begin{aligned}\min_{\mathbf{A}_k} \sum_{i_k=1}^{I_k} \Big\| \mathrm{diag}(\widehat{\mathbf{w}}_{i_k}^k) \left(\mathbf{y}_{i_k}^k - \mathbf{Q}_k \mathbf{a}_i^k \right) \Big\|_2^2 \\ + \rho \|\mathbf{A}_n\|_F^2 + \mu_k \|\mathbf{T}_k \mathbf{A}_n\|_F^2,\end{aligned} \tag{9}$$

where $\widehat{\mathbf{w}}_{i_k}^k = \widehat{\mathcal{W}}^{(k)}(:, i_k)$, $\mathbf{y}_{i_k}^k = \mathcal{Y}^{(k)}(:, i_k)$ and $\mathbf{a}_i^k = \mathbf{A}_k(i_k, :)^T$. Note that we do not need to instantiate $\mathbf{Q}_k$ because only the non-zero elements of the sparse vector $\widehat{\mathbf{w}}_{i_k}^k$ contribute to the cost function. The non-zero elements of $\widehat{\mathbf{w}}_{i_k}^k$ correspond to the observed data points for which the k-th variable takes the value i_k and therefore we need to compute the corresponding rows of the Khatri-Rao product. Problem 9 can be optimally solved by finding the solution to a set of linear equations obtained after setting the gradient to zero e.g., using the conjugate Gradient descent algorithm (Bertsekas 1997). Simpler updates can be obtained by fixing all variables except for a single row of the factor $\mathbf{A}_k$. Let us fix every parameter except for the i_k-th row of $\mathbf{A}_k$

$$\begin{aligned}\min_{\mathbf{a}_i^k} \frac{1}{M} \big\| \mathrm{diag}(\widehat{\mathbf{w}}_{i_k}^k)(\mathbf{y}_i^k - \mathbf{Q}_k \mathbf{a}_i^k \big\|_2^2 + \rho \|\mathbf{a}_i^k\|_2^2 \\ + \mu_k \left\| \mathbf{a}_{i-1}^k - \mathbf{a}_i^k \right\|_2^2 + \mu_k \left\| \mathbf{a}_{i+1}^k - \mathbf{a}_i^k \right\|_2^2,\end{aligned} \tag{10}$$

The solution for $\mathbf{a}_i^k$ is given by

$$\begin{aligned}\mathbf{a}_i^k = (\mathbf{Q}_k^T \mathrm{diag}(\mathbf{w}_i)^2 \mathbf{Q}_k + (\rho + 2\mu_k)\mathbf{I})^{-1} \\ (\mathbf{Q}_k^T \mathrm{diag}(\mathbf{w}_i)^2 \mathbf{y}_i^k - \mu_k (\mathbf{a}_{i-1}^k + \mathbf{a}_{i+1}^k))\end{aligned} \tag{11}$$

which results in very lightweight row-wise updates. BCD algorithms usually offer faster convergence in terms of the cost function compared to stochastic algorithms for small or moderate size problems. For large-scale problems on the other hand, Stochastic Gradient Descent (SGD) can attain moderate solution accuracy faster than BCD. The merits of both alternating optimization and stochastic optimization can be combined by considering block-stochastic updates (Xu and Yin 2015). In this work, we propose an easy to implement ALS algorithm as our main goal is to present a fresh perspective on the nonlinear identification problem through low-rank tensor completion. Further algorithmic developments are underway, but beyond the scope of this first submission. Next, we show how the proposed approach can be extended to handle partially observed and multi-output regression tasks.

Missing Data

It is quite common in general to have observations with missing values for one or more predictors. For example, in the grade prediction task described in the introduction, the predictions for a student rely on the student's performance achieved in previously taken courses. Consider a student-grade matrix $\mathbf{D} \in \mathbb{R}^{M \times N}$ where our goal is to predict the N-th course. The matrix will be in general sparse since each student enrolls in only few of the available courses, and the selected courses vary from student to student.

Common approaches for handling missing data include (1) removal of observations with any missing values, (2) imputing the missing values before training e.g., by replacing them with the mean, median, or the mode, and (3) directly handling the imputation by the algorithm. Let $\mathcal{O} = \{o_1, \ldots, o_T\}$ and $\mathcal{M} = \{m_1, \ldots, m_L\}$ denote the indices of the observed and missing entries of a single observation respectively. Instead of ignoring observations with missing entries we aim at computing the expectation of the nonlinear function conditioned on the observed variables i.e., we set

$$\begin{aligned}f(\mathbf{x}_\mathcal{O}) &= \mathbb{E}_{\mathbf{x}_\mathcal{M}|\mathbf{x}_\mathcal{O}}[f(\mathbf{x}_\mathcal{O}, \mathbf{x}_\mathcal{M})] \\ &= \sum_{\mathbf{x}_\mathcal{M}} \mathsf{Pr}(\mathbf{x}_\mathcal{M}|\mathbf{x}_\mathcal{O}) f(\mathbf{x}_\mathcal{O}, \mathbf{x}_\mathcal{M}).\end{aligned} \tag{12}$$

Estimating the conditional probability $\mathsf{Pr}(\mathbf{x}_\mathcal{M}|\mathbf{x}_\mathcal{O})$ is not possible since the number of parameters grows exponentially with the number of missing entries. Given the low-rank structure of the nonlinear function we propose modeling the Probability Mass Function (PMF) using a nonnegative CPD model which is a universal model for PMF estimation (Kargas, Sidiropoulos, and Fu 2018). For the sake of simplicity, we adopt a simple rank-one joint PMF model estimated via the empirical first-order marginals (Huang and Sidiropoulos 2017). Without loss of generality assume that the first T predictors are known and the remaining missing, then, the expectation can be computed very efficiently

$$\begin{aligned}&f(\mathbf{x}_\mathcal{O}) = \mathbb{E}_{\mathbf{x}_\mathcal{M}|\mathbf{x}_\mathcal{O}}[f(\mathbf{x}_\mathcal{O}, \mathbf{x}_\mathcal{M})] \\ &= \mathcal{X}(i_1, \ldots, i_T, :, \ldots, :) \times_{T+1} \mathbf{p}_{T+1} \cdots \times_{T+L} \mathbf{p}_N \\ &= \sum_{f=1}^F \prod_{n=1}^T \mathbf{A}_n(i_n, f) \prod_{n=T+1}^N \mathbf{p}_n^T \mathbf{A}_n(:, f).\end{aligned} \tag{13}$$

Table 1: Comparison of RMSE performance of different models on UCI datasets without missing data.

Dataset	RR	SVR (RBF)	SVR (polynomial)	DT	MLP (5 Layer)	CSID
Energy Eff. (1)	2.91±0.17	2.68±0.17	4.09±0.49	0.56±0.03	**0.48±0.06** [**50**]	**0.39±0.05**
Energy Eff. (2)	3.09±0.19	3.03±0.21	4.14±0.44	1.86±0.19	**0.97±0.14** [**50**]	**0.57±0.09**
C. Comp. Strength	10.47±0.42	9.72±0.38	11.30±0.36	6.57±0.82	**4.92±0.63** [**50**]	**4.67±0.50**
SkillCraft Master Table	1.68±1.61	**0.99±0.03**	1.22±0.05	1.03±0.04	1.00±0.03 [10]	**0.91±0.02**
Abalone	2.25±0.10	**2.19±0.08**	3.90±3.43	2.35±0.08	**2.09±0.09** [**10**]	2.23±0.09
Wine Quality	0.76±0.02	**0.69±0.02**	1.01±0.39	0.75±0.03	0.72±0.02 [10]	**0.70±0.02**
Parkinsons Tel. (1)	7.51±0.11	6.66±0.14	7.89±0.88	**2.40±0.26**	3.60±0.18 [100]	**1.33±0.10**
Parkinsons Tel. (2)	9.75±0.15	9.14±0.17	10.04±0.43	**2.60±0.38**	5.01±0.19 [100]	**1.79±0.17**
C. Cycle Power Plant	5.51±0.09	4.13±0.09	8.00±0.19	**3.98±0.13**	4.06±0.11 [50]	**3.76±0.15**
Bike Sharing (1)	36.45±0.46	32.67±0.81	34.93±0.97	18.89±0.36	**14.81±0.44** [**100**]	**15.17±0.44**
Bike Sharing (2)	122.65±2.87	113.18±1.73	117.25±2.01	42.06±2.06	**38.69±1.24** [**100**]	**36.93±1.19**
Phys. Prop.	5.19±0.03	4.91±1.26	6.49±1.15	4.40±0.04	**4.20±0.05** [**100**]	**4.21±0.04**

Table 2: Comparison of RMSE performance of different models on UCI datasets with 30% missing data.

Dataset	RR	SVR (RBF)	SVR (polynomial)	DT	MLP (5 Layer)	CSID
Energy Eff. (1)	3.01±0.15	3.38±0.27	6.88±0.63	2.57±0.49	**2.49±0.48** [**10**]	**2.17±0.25**
Energy Eff. (2)	3.26±0.16	3.57±0.30	6.65±0.48	**2.64±0.28**	3.02±0.36 [10]	**2.48±0.22**
C. Comp. Strength	10.33±0.61	11.39±0.48	13.16±1.17	**9.90±1.05**	10.01±0.54 [10]	**9.69±0.79**
SkillCraft Master Table	1.79±1.63	**1.05±0.03**	1.61±0.33	1.08±0.03	1.10±0.04 [10]	**1.05±0.01**
Abalone	**2.27±0.07**	2.31±0.08	3.12±0.79	2.42±0.07	**2.28±0.07** [**10**]	2.40±0.13
Wine Quality	0.76±0.02	**0.73±0.02**	0.93±0.21	0.78±0.02	**0.76±0.03** [**10**]	0.78±0.02
Parkinsons Tel. (1)	7.52±0.11	6.91±0.13	8.12±0.11	**3.10±0.22**	5.90±0.28 [10]	**4.98±0.12**
Parkinsons Tel. (2)	9.76±0.18	9.38±0.21	10.68±0.23	**3.59±0.81**	7.67±0.18 [10]	**6.58±0.18**
C. Cycle Power Plant	5.51±0.09	6.16±0.15	10.45±0.31	**5.29±0.36**	5.33±0.07 [50]	**5.04±0.12**
Bike Sharing (1)	37.40±0.52	35.50±0.31	36.85±0.38	25.41±1.5	**21.51±0.83**± [**50**]	**23.89±0.19**
Bike Sharing (2)	123.81±1.26	127.06±1.55	130.20±1.13	**71.93±1.18**	**64.03±1.66** [**50**]	75.65±1.51
Phys. Prop.	5.18±0.02	7.53±0.67	7.87±0.83	5.08±0.03	**4.99±0.09** [**100**]	**4.70±0.03**

Table 3: Comparison of RMSE performance of different models on multi-output regression.

Dataset	RR	MLP (1 Layer)	MLP (3 Layer)	MLP (5 Layer)	DT	CSID
En. Eff. (2)	2.70±0.19	2.82±0.08 [50]	2.73±0.11[100]	2.67±0.11[10]	**2.19±0.19**	**2.01±0.14**
Park. Tel. (2)	12.19±0.09	7.59±0.21[250]	6.54±0.06[250]	6.18±0.42[250]	**3.37±0.39**	**2.85±0.22**
B. Shar. (2)	127.75±3.32	64.12±6.49[250]	43.60±1.95[100]	**42.25±1.22**[**100**]	46.21±1.20	**45.29±1.47**

In this case, we minimize the squared error between the target value and the conditional expectation of the function. The modification can be easily incorporated in the ALS algorithm. Rich dependencies between the variables can also be captured using a higher-order PMF model, but we defer this discussion to follow-up work due to space limitations.

Multi-Output Regression

The proposed framework is quite flexible and can easily be extended to vector-valued functions $f : \{1, \ldots, I\}^N \rightarrow \mathbb{R}^K$. When there is no correlation between the output variables of a system, one can build K independent models, one for each output, and then use those models to independently predict each one of the K outputs. However, it is likely that the output values related to the same input are themselves correlated and often a better way is to build a single model capable of predicting simultaneously all K outputs. We can treat each different model as an N-way tensor and stack them together to build an $(N+1)$-way tensor. The new tensor model can be described by $N + 1$ factors associated with the N predictors and an additional mode of dimension K, $\mathcal{X} = [\![\mathbf{A}_1, \ldots, \mathbf{A}_N, \mathbf{V}]\!]_F$. The vector-valued prediction for $[i_1, \ldots, i_N]^T$ is given by $\mathcal{X}(i_1, \ldots, i_N, j) = \sum_{f=1}^{F} \mathbf{V}(j, f) \prod_{n=1}^{N} \mathbf{A}_n(i_n, f)$. In matrix form we have

$$\mathcal{X}(i_1, \ldots, i_N, :) = (\mathbf{A}_1(i_1, :) \circledast \cdots \circledast \mathbf{A}_N(i_N, :)) \, \mathbf{V}^T$$

No modification is needed for the ALS updates. Depending on the application one may or may not need to apply smoothness regularization on $\mathbf{V}$.

Experiments

We evaluate the proposed approach in single output regression tasks using several datasets obtained from the UCI machine learning repository (Lichman 2013). Our proposed approach is implemented in MATLAB using the Tensor Toolbox (Bader and Kolda 2007) for tensor operations. We then assess the ability of our model to handle missing predictors by hiding 30% of the data as well as its ability to predict vector valued responses. For each experiment we split the dataset into two sets, 80% used for training and 20% for testing, and run 10 Monte-Carlo simulations. Finally, we evaluate the performance of our approach in a challenging student grade prediction task using a real student grade dataset. For each method we tune the hyper-parameters using 5-fold cross-validation. We compare the performance of the different algorithms in terms of the Root Mean Square Error (RMSE).

UCI Datasets

We used four different machine learning algorithms as baselines, Ridge Regresion (RR), Support Vector Regression (SVR), Decision Tree (DT) and Multilayer Perceptrons

Table 4: Comparison of RMSE performance on student grade data.

Dataset	GPA	BMF	CSID
CSCI-1	0.52±0.02	**0.48±0.03**	**0.48±0.03**
CSCI-2	0.56±0.02	**0.55±0.02**	**0.55±0.03**
CSCI-3	**0.48±0.04**	**0.48±0.04**	**0.48±0.05**
CSCI-4	0.53±0.03	0.52±0.04	**0.51±0.03**
CSCI-5	0.43±0.02	0.43±0.02	**0.42±0.02**
CSCI-6	0.63±0.03	0.58±0.03	**0.57±0.03**
CSCI-7	0.57±0.02	0.58±0.01	**0.56±0.02**
CSCI-8	0.52±0.02	0.49±0.03	**0.47±0.02**
CSCI-9	0.61±0.03	0.60±0.05	**0.57±0.03**
CSCI-10	0.58±0.04	**0.56±0.04**	**0.56±0.04**
CSCI-11	0.68±0.06	**0.66±0.04**	0.67±0.03
CSCI-12	0.58±0.04	0.51±0.04	**0.48±0.01**
CSCI-13	0.67±0.03	0.55±0.05	**0.54±0.03**
CSCI-14	0.70±0.06	**0.62±0.03**	0.65±0.07
CSCI-15	0.56±0.03	**0.56±0.06**	0.57±0.03
CSCI-16	0.52±0.03	0.51±0.03	**0.50±0.02**
CSCI-17	0.60±0.02	**0.58±0.05**	0.59±0.05
CSCI-18	0.57±0.03	0.56±0.05	**0.55±0.04**
CSCI-19	0.68±0.04	0.70±0.04	**0.61±0.04**
CSCI-20	0.61±0.06	**0.58±0.02**	0.63±0.04

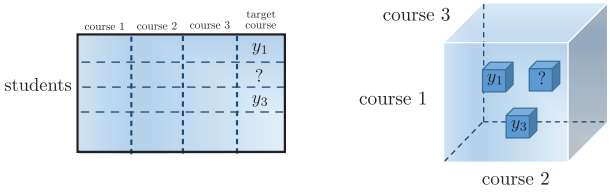

Figure 3: Low-rank matrix completion (left) canonical system identification (right).

(MLPs) using the implementation of scikit-learn (Pedregosa et al. 2011). For RR, SVR and MLP we standardize each ordinal feature such that it has zero mean and unit variance. Categorical features are transformed using one-hot encoding. For DT no preprocessing step is required. For our method, we fix the alphabet size to be $I = 25$ and use Lloyd-Max scalar quantizer for discretization of continuous predictors. For the MLPs, we set the number of hidden layers to 1,3 or 5 and varied the number of nodes per layer $10, 50, 100$ and 250. We observed that in most cases the MLP with 5 hidden layers performed better than the 1 or 3 layer MLP and that further increasing the number of layers did not improve the performance.

Table 1 shows the RMSE performance of the different methods when there are no missing predictors on the datasets. The number inside the square brackets denotes the number of nodes for each layer of MLP. We highlight the two best performing methods for each dataset. Our approach performs similarly or better than best baseline in most of the datasets. Note that both decision trees and our approach rely on discretization of continuous predictors however, adding the smooth regularization plays a significant role in boosting the RMSE performance for our method.

Next, we evaluate our approach on partially observed datasets. We randomly hide 30% of the full dataset and repeat 10 Monte-Carlo simulations. Before fitting the data to the baseline algorithms we replace each missing entry of an ordinal predictor with the mean and for each categorical predictor we use the most frequent value (mode). For our algorithm we use a rank-1 approximation of the joint PMF tensor estimated from the training data. Table 2 shows the performance of the different algorithms in this setting. Again, our approach similarly or better than best baseline.

Finally, we test our approach in predicting multi-output responses against RR, DT tree and MLPs. Table 3 contains the results for three datasets. Similarly to the single output setting our approach performs the same or slightly better compared to the baseline methods.

Grade Prediction Datasets

Finally we evaluate our method in a student grade prediction task on a real dataset obtained from the CS department of a university. The predictors corespond to the course grades the students have received. Specifically, we used the 20 most frequent courses to build 20 independent single output regression tasks each one of them having 34 predictors. Grades take 11 discrete values (A-F) and due to the natural ordering between the different values smoothness regularization was applied on all factors. We used the Grade Point Average (GPA) and Biased Matrix Factorization as our baselines. Low-rank matrix completion is considered a state-of-art method in student grade prediction (Polyzou and Karypis 2016; Almutairi, Sidiropoulos, and Karypis 2017). Note that in the matrix case each course is represented by a column while in the proposed tensor approach, each course is represented by a tensor mode (Figure 3). Table 4 shows the results for the different algorithms. Our approach outperforms BMF in 11 tasks, performs the same in 4 and worse in 5.

Conclusion and Future work

In this paper, we considered the problem of nonlinear system identification. We formulated the problem as a smooth tensor completion problem with missing data and developed a lightweight BCD algorithm to tackle it. We have proposed a simple approach to handle randomly missing data and extended our model to vector valued function approximation. Experiments on several real data regression tasks showcased the effectiveness of the proposed approach.

Acknowledgements

The work of the authors was supported in part by NSFIIS-1447788, IIS-1704074.

References

Almutairi, F. M.; Sidiropoulos, N. D.; and Karypis, G. 2017. Context-aware recommendation-based learning analytics using tensor and coupled matrix factorization. *IEEE Journal of Selected Topics in Signal Processing* 11(5):729–741.

Anandkumar, A.; Ge, R.; Hsu, D.; Kakade, S. M.; and Telgarsky, M. 2014. Tensor decompositions for learning latent variable models. *The Journal of Machine Learning Research* 15(1):2773–2832.

Bader, B. W., and Kolda, T. G. 2007. Efficient MATLAB computations with sparse and factored tensors. *SIAM Journal on Scientific Computing* 30(1):205–231.

Bertsekas, D. P. 1997. Nonlinear programming. *Journal of the Operational Research Society* 48(3):334–334.

Blondel, M.; Fujino, A.; Ueda, N.; and Ishihata, M. 2016a. Higher-order factorization machines. In *Advances in Neural Information Processing Systems*, 3351–3359.

Blondel, M.; Ishihata, M.; Fujino, A.; and Ueda, N. 2016b. Polynomial networks and factorization machines: New insights and efficient training algorithms. In *International Conference on Machine Learning*, 850–858.

Bousse, M.; Vervliet, N.; Domanov, I.; Debals, O.; and De Lathauwer, L. 2018. Linear systems with a canonical polyadic decomposition constrained solution: Algorithms and applications. *Numerical Linear Algebra with Applications* 25(6):e2190.

Boussé, M.; Debals, O.; and De Lathauwer, L. 2017. Tensor-based large-scale blind system identification using segmentation. *IEEE Transactions on Signal Processing* 65(21):5770–5784.

Bro, R. 1998. *Multi-way analysis in the food industry*. Ph.D. Dissertation.

Fu, X.; Huang, K.; Ma, W.; Sidiropoulos, N. D.; and Bro, R. 2015. Joint tensor factorization and outlying slab suppression with applications. *IEEE Transactions on Signal Processing* 63(23):6315–6328.

Gandy, S.; Recht, B.; and Yamada, I. 2011. Tensor completion and low-n-rank tensor recovery via convex optimization. *Inverse Problems* 27(2):025010.

Hastie, T.; Tibshirani, R.; and Friedman, J. 2009. *The Elements of Statistical Learning*. Springer Series in Statistics.

Huang, K., and Sidiropoulos, N. D. 2017. Kullback-Leibler principal component for tensors is not NP-hard. In *Asilomar Conference on Signals, Systems, and Computers*, 693–697.

Imaizumi, M., and Hayashi, K. 2017. Tensor decomposition with smoothness. In *International Conference on Machine Learning*, volume 70, 1597–1606.

Jain, P., and Oh, S. 2014. Provable tensor factorization with missing data. In *Advances in Neural Information Processing Systems 27*, 1431–1439.

Kanatsoulis, C. I.; Fu, X.; Sidiropoulos, N. D.; and Akçakaya, M. 2019. Tensor completion from regular sub-nyquist samples. *arXiv preprint arXiv:1903.00435*.

Kargas, N., and Sidiropoulos, N. D. 2019. Learning mixtures of smooth product distributions: Identifiability and algorithm. In *International Conference on Artificial Intelligence and Statistics*, volume 89, 388–396.

Kargas, N.; Sidiropoulos, N. D.; and Fu, X. 2018. Tensors, learning, and "Kolmogorov extension" for finite-alphabet random vectors. *IEEE Transactions on Signal Processing* 66(18):4854–4868.

Krishnamurthy, A., and Singh, A. 2013. Low-rank matrix and tensor completion via adaptive sampling. In *Advances in Neural Information Processing Systems*, 836–844.

Lacroix, T.; Usunier, N.; and Obozinski, G. 2018. Canonical tensor decomposition for knowledge base completion. In *International Conference on Machine Learning*, volume 80, 2863–2872.

Lichman, M. 2013. UCI machine learning repository.

Novikov, A.; Trofimov, M.; and Oseledets, I. 2016. Exponential machines. In *International Conference on Learning Representations Workshop*.

Oseledets, I. V. 2011. Tensor-train decomposition. *SIAM Journal on Scientific Computing* 33(5):2295–2317.

Papalexakis, E. E.; Akoglu, L.; and Ience, D. 2013. Do more views of a graph help? community detection and clustering in multi-graphs. In *International Conference on Information FUSION*, 899–905.

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

Perros, I.; Wang, F.; Zhang, P.; Walker, P.; Vuduc, R.; Pathak, J.; and Sun, J. 2017. Polyadic regression and its application to chemogenomics. In *SIAM International Conference on Data Mining*, 72–80.

Polyzou, A., and Karypis, G. 2016. Grade prediction with models specific to students and courses. *International Journal of Data Science and Analytics* 2(3):159–171.

Rendle, S. 2010. Factorization machines. In *IEEE International Conference on Data Mining*, 995–1000.

Sidiropoulos, N. D.; De Lathauwer, L.; Fu, X.; Huang, K.; Papalexakis, E. E.; and Faloutsos, C. 2017. Tensor decomposition for signal processing and machine learning. *IEEE Transactions on Signal Processing* 65(13):3551–3582.

Smith, S., and Karypis, G. 2015. Tensor-matrix products with a compressed sparse tensor. In *Workshop on Irregular Applications: Architectures and Algorithms*, 5.

Sorensen, M., and De Lathauwer, L. 2019. Fiber sampling approach to canonical polyadic decomposition and application to tensor completion. *SIAM Journal on Matrix Analysis and Applications* 40(3):888–917.

Tomasi, G., and Bro, R. 2005. PARAFAC and missing values. *Chemometrics and Intelligent Laboratory Systems* 75(2):163–180.

Van Eeghem, F.; Debals, O.; Vervliet, N.; and De Lathauwer, L. 2018. Coupled and incomplete tensors in blind system identification. *IEEE Transactions on Signal Processing* 66(23):6137–6147.

Xiong, L.; Chen, X.; Huang, T.-K.; Schneider, J.; and Carbonell, J. G. 2010. Temporal collaborative filtering with bayesian probabilistic tensor factorization. In *SIAM International Conference on Data Mining*, 211–222.

Xu, Y., and Yin, W. 2015. Block stochastic gradient iteration for convex and nonconvex optimization. *SIAM Journal on Optimization* 25(3):1686–1716.

Yokota, T.; Zhao, Q.; and Cichocki, A. 2016. Smooth PARAFAC decomposition for tensor completion. *IEEE Transactions on Signal Processing* 64(20):5423–5436.

Yuan, M., and Zhang, C.-H. 2016. On tensor completion via nuclear norm minimization. *Foundations of Computational Mathematics* 16(4):1031–1068.