GRADIENT-BASED APPROACHES TO LEARN TENSOR PRODUCTS

Markus Rupp, Stefan Schwarz[*]

Technical University of Vienna, Austria
Institute of Telecommunications
Email: {mrupp,sschwarz}@nt.tuwien.ac.at

ABSTRACT

Tensor algebra has become of high interest recently due to its application in the field of so-called Big Data. For signal processing a first important step is to compress a vast amount of data into a small enough set so that particular issues of interest can be investigated with todays computer methods. We propose various gradient-based methods to decompose tensors of matrix products as they appear in structured multiple-input multiple-output systems. While some methods work directly on the observed tensor, others use input-output observations to conclude to the desired decomposition. Although the algorithms are nonlinear in nature, they are being treated as linear estimators; numerical examples validate our results.

Index Terms— Tensors, Decomposition, BigData

1. INTRODUCTION

Big Data is the keyword for future innovations [1], changing not only economies but also our daily life. Once huge data amounts are available, many questions about our society can be answered now in a short time. Even the structure of the Internet will change in future to support effective data access [2]. In the signal processing domain, typical questions treat on how to compute such huge data amounts efficiently and, related to this, how data compression can work best in order to reduce complexity. Linearly separable operators are well known in the context of fast algorithmic implementations such as Fast Hadamard or Fast Fourier Transformations. Once a multidimensional data set can be separated into a set of smaller vectors, matrices or tensors, huge data compressions can be achieved. We provide here a copy of the definition of separability for the convenience of the reader, see, e.g., Definition 9.5 in [3].

Definition 1.1 *A linear operator* $\mathbf{C}$ *is said to be separable if* $\mathbf{C} = \mathbf{B} \otimes \mathbf{A}$ *for some* $\mathbf{B}$ *and* $\mathbf{A}$.

Obviously due to the product term, there is not a unique solution, if it exists; every $\{\gamma\mathbf{A}, \gamma^{-1}\mathbf{B}\}$ is also a solution for $\gamma \neq 0$. While it is straightforward to show how to save complexity (typically savings from M^2 to $M\log_2(M)$ are significant), to show if a vector or matrix is separable or not is not an easy step.

1.1. Relation to Prior Work

Quite recently there were entire IEEE Signal Processing Magazines devoted to tensor algebra (May2014) and the topic Big Data (Sept. 2014). While tensor based methods were originally introduced more than a hundred years ago in differential geometry, such methods have been in use for several decades only by a few researchers. They recently have received a lot of attention in the context of screening vast amount of data, typically from the Internet but also from other sources such as video cameras in security contexts. The original problems date back into the 60s with pioneering work of Tucker [4] and [5–7] in phonetics; they are nowadays mostly referred to as CANDECOMP/PARAFAC (CP) [8]. Rediscoveries by Sidiropoulos, Bro, and Giannakis [9] pushed the field forward as now improved iterative methods were available that offer to decompose a given data set into much smaller but information preserving units. A common problem is to decompose a tensor into its constituents, e.g.,

$$\mathbf{C} = \mathbf{A}_1 \otimes \mathbf{A}_2 \otimes \mathbf{A}_3 \qquad (1)$$

in which the matrices are of smaller dimension. Typically multidimensional data streams are described this way, one index often refers to time or events, e.g., video streams. Tensor applications are in compressing information [10], improvement of incomplete and inaccurate observations [11], matrix completion methods [12], and data mining [13] to name a few. Good overviews are available in [14–19]. The particular problem of this contribution has been tackled by van Loan and Pitsianis [20], however under the condition that the matrix in question is perfectly decomposable.

1.2. Our Contribution and Paper Structure

In the following Section 2 we show several fundamental properties in the context of an Least Squares (LS) approach to uniquely decompose a single tensor product of a two-way tensor. Based on the well-known LS orthogonality property we reveal several desirable properties that can be utilized in the context of data mining, data compression and matrix completion. We extend the approach in [20] towards arbitrary matrices and present the problem rigorously in an LS context including low-complex iterative approaches. In Section 3 we present a low complexity iterative method to solve the problem. In Section 4 we briefly summarize Least Mean Squares (LMS) algorithms to learn structured multiple-input multiple-output (MIMO) systems under the condition of input and output observations and we extent such description in Section 5 where we propose gradient type solutions. While the simple rank one tensor solution ($\mathbf{A}_i$ in (1) are vectors) was proposed in [21], we will here present the result for matrices $\{\mathbf{A}_i\}$ rather than vectors $\{\mathbf{a}_i\}$, describing an MIMO system. In Section 6 we validate our result by simulations and in Section 7 we conclude the paper with some remarks.

Notation: We describe the Hermitian of a vector or matrix by upperscript H and denote the Kronecker product by $\otimes$. All signals are considered complex-valued. The operator vec() realigns a matrix

[*]This work was supported by the Austrian Science Fund (FWF) under Awards S10611-N13 within the National Research Network SISE.

column by column into a vector. Further we denote the set of right side eigenvectors of a matrix by evec().

2. LINEARLY SEPARABLE OPERATORS

Theorem 2.1 *Consider a matrix* $\mathbf{C} \in \mathbb{C}^{M_1 \times M_2}$ *with* $M_1 = N_1 P_1, M_2 = N_2 P_2$, *with* $N_1 N_2 > 1$ *and* $P_1 P_2 > 1$, *i.e.,*

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_{11} & \mathbf{C}_{12} & ... & \mathbf{C}_{1P_2} \\ \mathbf{C}_{21} & \mathbf{C}_{22} & ... & \mathbf{C}_{2P_2} \\ \vdots & & & \\ \mathbf{C}_{P_1 1} & \mathbf{C}_{P_1 2} & ... & \mathbf{C}_{P_1 P_2} \end{bmatrix}$$

with $\mathbf{C}_{kl} \in \mathbb{C}^{N_1 \times N_2}; k = 1, 2, ..., P_1; l = 1, 2, ..., P_2$. *There uniquely exist (up to a phase*[1]*) two matrices* $\mathbf{A} \in \mathbb{C}^{N_1 \times N_2}, \mathbf{B} \in \mathbb{C}^{P_1 \times P_2}$ *with* $\|\mathbf{A}\|_F = 1$ *such that*

$$(\mathbf{A}, \mathbf{B}) = \arg \min_{\mathbf{A},\mathbf{B}} \|\mathbf{C} - \mathbf{B} \otimes \mathbf{A}\|_F^2$$

in the LS sense given by the eigenvector associated to the largest eigenvalue:

$$\mathbf{a} = \mathrm{vec}(\mathbf{A}) = \arg\max \mathrm{evec} \sum_{k=1}^{P_1} \sum_{l=1}^{P_2} \mathrm{vec}(\mathbf{C}_{kl}) \mathrm{vec}(\mathbf{C}_{kl})^H.$$

With $\mathbf{c}_{kl} = \mathrm{vec}(\mathbf{C}_{kl})$ *we obtain for the coefficients* b_{mn} *of* $\mathbf{B}$*:*

$$b_{mn} = \mathbf{a}^H \mathbf{c}_{mn} \,; m = 1, 2, ..., P_1; n = 1, 2, ..., P_2.$$

Proof: We first compute the elements of matrix $\mathbf{B}$ by LS and obtain:

$$\frac{\partial}{\partial b_{mn}} \sum_{k=1}^{P_1} \sum_{l=1}^{P_2} \|\mathbf{C}_{kl} - b_{kl} \mathbf{A}\|_F^2 = 0 \qquad (2)$$

that is

$$b_{mn} = \frac{\mathrm{tr}(\mathbf{A}^H \mathbf{C}_{mn})}{\mathrm{tr}(\mathbf{A}^H \mathbf{A})} = \mathbf{a}^H \mathbf{c}_{mn}, \qquad (3)$$

due to the norm constraint on $\mathbf{A}$. We now have to minimize

$$\min_{\mathbf{A}} \sum_{k=1}^{P_1} \sum_{l=1}^{P_2} \left\| \mathbf{C}_{kl} - \mathrm{tr}(\mathbf{A}^H \mathbf{C}_{kl}) \mathbf{A} \right\|_F^2 \qquad (4)$$

which is equivalent to minimizing

$$\min_{\mathbf{A}} \sum_{k=1}^{P_1} \sum_{l=1}^{P_2} \mathrm{tr}(\mathbf{C}_{kl}^H \mathbf{C}_{kl}) - \mathrm{tr}(\mathbf{C}_{kl}^H \mathbf{A}) \mathrm{tr}(\mathbf{A}^H \mathbf{C}_{kl}). \qquad (5)$$

As $\mathrm{tr}(\mathbf{A}^H \mathbf{C}_{kl}) = \mathbf{a}^H \mathbf{c}_{kl}$, we find equivalently

$$\min_{\mathbf{A}} \sum_{k=1}^{P_1} \sum_{l=1}^{P_2} \mathbf{c}_{kl}^H \mathbf{c}_{kl} - \mathbf{a}^H \mathbf{c}_{kl} \mathbf{c}_{kl}^H \mathbf{a}, \qquad (6)$$

the solution of which is given by the eigenvector associated to the largest eigenvalue:

$$\mathbf{a} = \arg\max \mathrm{evec} \sum_{k=1}^{P_1} \sum_{l=1}^{P_2} \mathbf{c}_{kl} \mathbf{c}_{kl}^H. \qquad (7)$$

While this first theorem provides simple construction of smaller matrices out of a large matrix, it does not state anything about the quality of the separation. Even if a large matrix $\mathbf{C}$ is separable, it may encounter additional corruption due to the observation process. Along the same lines a similar proof has been presented in [20] but under the condition of perfectly separable matrices. There are also some further interesting properties deduced from (3), i.e., if $\mathbf{C}$ is {non-negative, positive definite, banded, symmetric, orthogonal, diagonal, triangular, stochastic}, so is $\mathbf{A}$ and $\mathbf{B}$.

Theorem 2.2 (Orthogonality) *Given a set of matrices* $\{\mathbf{A}, \mathbf{B}\}$ *according to Theorem 2.1 to separate a matrix* $\mathbf{C}$ *in an LS sense,* $\mathbf{C}$ *can exactly be represented by*

$$\mathbf{C} = \mathbf{B} \otimes \mathbf{A} + \mathbf{N}$$

with error matrix $\mathbf{N}$ *being orthogonal onto the tensor product*

$$\mathrm{tr}\left(\mathbf{N}^H (\mathbf{B} \otimes \mathbf{A}) \right) = 0.$$

Proof: We write

$$\mathbf{B} \otimes \mathbf{A} = \qquad (8)$$
$$\begin{bmatrix} \mathbf{a}^H \mathbf{c}_{11} \mathbf{A} & \mathbf{a}^H \mathbf{c}_{12} \mathbf{A} & ... & \mathbf{a}^H \mathbf{c}_{1P_2} \mathbf{A} \\ \mathbf{a}^H \mathbf{c}_{21} \mathbf{A} & \mathbf{a}^H \mathbf{c}_{22} \mathbf{A} & ... & \mathbf{a}^H \mathbf{c}_{2P_2} \mathbf{A} \\ \vdots & & & \\ \mathbf{a}^H \mathbf{c}_{P_1 1} \mathbf{A} & \mathbf{a}^H \mathbf{c}_{P_1 2} \mathbf{A} & ... & \mathbf{a}^H \mathbf{c}_{P_1 P_2} \mathbf{A} \end{bmatrix}.$$

We consider for each sub-block $\mathbf{C}_{kl}$ of $\mathbf{C}$:

$$\begin{aligned} &\mathrm{tr}\left[\left(\mathbf{C}_{kl} - \mathbf{A}\mathbf{a}^H \mathbf{c}_{kl} \right)^H \mathbf{A}\mathbf{a}^H \mathbf{c}_{kl} \right] \qquad (9) \\ &= \left(\mathbf{c}_{kl} - \mathbf{a}\mathbf{a}^H \mathbf{c}_{kl} \right)^H \mathbf{a}\mathbf{a}^H \mathbf{c}_{kl} = 0. \end{aligned}$$

The following theorem eventually provides a quantitative measure that further describes the quality of the separation process based on the previously shown orthogonality principle.

Theorem 2.3 (MMSE) *Separating a matrix* $\mathbf{C}$ *in the LS sense, the corresponding minimum MSE (MMSE) is given by*

$$\begin{aligned} \mathrm{MMSE} &= \mathrm{tr}(\mathbf{N}^H \mathbf{N}) \\ &= \sum_{k=1}^{P_1} \sum_{l=1}^{P_2} \mathbf{c}_{kl}^H \left(\mathbf{I}_{M_1 M_2} - \frac{\mathbf{a}\mathbf{a}^H}{\mathbf{a}^H \mathbf{a}} \right) \mathbf{c}_{kl}. \end{aligned}$$

Proof: The proof follows from the orthogonality property of LS.

Matrix $\mathbf{N}$ allows for various extensions:

- If several partitions of $M_1 = N_1 P_1$ and $M_2 = N_2 P_2$ exist, the various MMSE values can be checked and based on those, it can be decided which partitioning is most suitable to linear separation. The normalized MMSE value, i.e.,

$$\gamma(N_1, N_2, P_1, P_2) = \frac{\mathrm{tr}(\mathbf{N}^H \mathbf{N})}{\mathrm{tr}(\mathbf{C}^H \mathbf{C})} \leq 1 \qquad (10)$$

  provides a convenient measure for the quality of the separation. Due to this measure we can also find optimal separation sizes $\{N_1, N_2, P_1, P_2\}$ if they are not given a-priori:

$$\min_{N_1 P_1 = M_1, N_2 P_2 = M_2} \gamma(N_1, N_2, P_1, P_2).$$

[1] If $\mathbf{A}$ is a solution, $e^{j\phi}\mathbf{A}$ is also a solution for arbitrary phases ϕ.

- In general the elements of matrix $\mathbf{N}$ tell us how ”noisy” our observation of $\mathbf{C}$ is. If the observation noise is equally distributed (stationary), then the noise power is constant over all areas of $\mathbf{N}$. If particular parts of $\mathbf{C}$ are more corrupted than others, they can be detected in $\mathbf{N}$. Such procedures are common in matrix completion methods [12].
- In the context of Big Data often only the correlations are important and this noisy part $\mathbf{N}$ does not contribute to it. However, if it is anomalies and outliers that are of interest then it is exactly this part $\mathbf{N}$ that needs to be investigated while the repetitive behavior in the decomposed tensor is unimportant.

Note that we explicitly stressed here the projection property although $\mathbf{a}^H\mathbf{a} = 1$. In case the uniqueness constraint on $\mathbf{A}$ is not satisfied, this more general formulation of the theorem still holds.

3. ITERATIVE SOLUTIONS

So far all required operations need to apply eigenvalue decompositions of supposedly large matrices which is of high computational burden. We are thus interested in alternative low complex forms which are being presented in the following.

Theorem 3.1 (Alternative Formulation) *Constructing the following matrix out of vectorized forms $\mathbf{c}_{kl}$ of the sub-matrices $\mathrm{vec}(\mathbf{C}_{kl})$: $\tilde{\mathbf{C}} = [\mathbf{c}_{11}, \mathbf{c}_{21}, ..., \mathbf{c}_{P_1 1}, \mathbf{c}_{12}, \mathbf{c}_{22}, ..., \mathbf{c}_{P_1 2}, ...\mathbf{c}_{1P_2}, \mathbf{c}_{2P_2}, ..., \mathbf{c}_{P_1P_2}]$, we can alternatively write*

$$\mathbf{a} = \arg\max \frac{\mathbf{a}^H\tilde{\mathbf{C}}\tilde{\mathbf{C}}^H\mathbf{a}}{\mathbf{a}^H\mathbf{a}}$$

and

$$\mathbf{b} = \tilde{\mathbf{C}}^H\mathbf{a},$$

where $\mathbf{a} = \mathrm{vec}(\mathbf{A})$ and $\mathbf{b} = \mathrm{vec}(\mathbf{B})$. Thus if we decompose $\tilde{\mathbf{C}}^H = \mathbf{U}\Sigma\mathbf{V}^H$, we find that $\mathbf{a} = \mathbf{v}_{\max}$ and $\mathbf{b} = \sigma_{\max}\mathbf{u}^*_{\max}$, i.e., the right and left singular vectors corresponding to the largest singular value.

Note that if matrix $\mathbf{C}$ is separable without observation errors, then all vectors $\mathbf{c}_{kl}$ are linearly dependent and thus the only nonzero singular value of $\tilde{\mathbf{C}}$ is $\sigma_{\max}$.

Theorem 3.2 (Iterative Formulation) *Consider matrix $\tilde{\mathbf{C}}$ from previous Theorem 3.1 and iterate the following steps, starting with $\mathbf{a}_1$:*

$$\bar{\mathbf{b}}_k = \tilde{\mathbf{C}}^H\mathbf{a}_k \quad (11)$$

$$\mathbf{a}_* = \tilde{\mathbf{C}}\bar{\mathbf{b}}_k \quad (12)$$

$$\mathbf{a}_{k+1} = \frac{\mathbf{a}_*}{\|\mathbf{a}_*\|_2}. \quad (13)$$

Starting with a randomly selected vector $\mathbf{a}_1$, the iterative scheme ends with probability one at the desired value $\mathbf{a}$ and $\bar{\mathbf{b}}$ ends at $\mathbf{b}^*$.

Proof: Consider a random starting value $\mathbf{a}_1 = \mathbf{V}\mathbf{x}$. We then obtain

$$\bar{\mathbf{b}}_1 = \mathbf{U}\Sigma\mathbf{x}$$

and

$$\mathbf{a}_2 = \mathbf{V}\Sigma^2\mathbf{x}.$$

Thus in general $\mathbf{a}_k = \mathbf{V}\Sigma^{2k}\mathbf{x}$. The largest term in Σ, i.e., $\sigma_{\max}$ will grow fastest. Since $\mathbf{a}_k$ is normalized after every step, eventually only $\lim_{k\to\infty}\mathbf{a}_k = \mathbf{v}_{\max}$ remains. As a consequence, $\lim_{k\to\infty}\mathbf{b}_k = \sigma_{\max}\mathbf{u}^*_{\max}$. If $\mathbf{x}$ is by chance selecting a single right singular vector of $\tilde{\mathbf{C}}$, that is not $\mathbf{v}_{\max}$, the iterative algorithm stalls at this value. As we randomly select starting vectors and there are only $N_1N_2 - 1$ of such non-favorable singular vectors, we pick them with probability zero and thus with probability one, the scheme converges.

Note that if observation matrix $\mathbf{C}$ is without noise, the iterative scheme converges in two steps due to its single singular value $\sigma_{\max}$. As long as the observation noise is sufficiently small, only few iterations are required, making the iterative scheme very attractive from a complexity and from a (fixed-point) implementation perspective.

4. ROBUST LMS ALGORITHM FOR MIMO SYSTEMS

We first start with the not so common LMS update algorithm in case the system to identify is an MIMO system. Given the sequence pairs of input-output vectors $\{\mathbf{x}_k, \mathbf{y}_k\}$ of a system $\mathbf{C} \in \mathbb{C}^{M_1\times M_2}$

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \mathbf{v}_k,$$

including observation noise $\mathbf{v}_k$, we find the error to be minimized

$$\mathbf{C}_o = \arg\min_{\mathbf{C}} E[\tilde{\mathbf{e}}_k^H\tilde{\mathbf{e}}_k], \quad (14)$$

$$\tilde{\mathbf{e}}_k = \mathbf{y}_k - \hat{\mathbf{C}}_{k-1}\mathbf{x}_k. \quad (15)$$

The LMS updates for such error read

$$\hat{\mathbf{C}}_k = \hat{\mathbf{C}}_{k-1} + \mu\tilde{\mathbf{e}}_k\mathbf{x}_k^H. \quad (16)$$

Following the lines of robustness analysis [22–26], we find the following relation for the system error matrix $\tilde{\mathbf{C}}_k = \mathbf{C}_o - \hat{\mathbf{C}}_k$:

$$\tilde{\mathbf{C}}_k\tilde{\mathbf{C}}_k^H + \bar{\mu}_k\mathbf{e}_k\mathbf{e}_k^H = \tilde{\mathbf{C}}_{k-1}\tilde{\mathbf{C}}_{k-1}^H + \bar{\mu}_k\bar{\mathbf{v}}_k\bar{\mathbf{v}}_k^H, \quad (17)$$

with the undistorted error vector $\mathbf{e}_k = \tilde{\mathbf{C}}_{k-1}\mathbf{x}_k$ and the abbreviations

$$\bar{\mu}_k = \frac{1}{\mathbf{x}_k^H\mathbf{x}_k}, \quad (18)$$

$$\bar{\mathbf{v}}_k = \frac{\mu_k}{\bar{\mu}_k}\mathbf{v}_k - \left(1 - \frac{\mu_k}{\bar{\mu}_k}\right)\mathbf{e}_k. \quad (19)$$

By computing the trace on both ends, and recalling that $\mathrm{tr}(\mathbf{A}^H\mathbf{A}) = \|\mathbf{A}\|_F^2$, we can find the following global robustness property.

Theorem 4.1 *The LMS algorithm with error term (15) and update (16) is globally robust:*

$$\frac{\left\|\tilde{\mathbf{C}}_N\right\|_F^2 + \sum_{k=1}^N \mu_k\mathbf{e}_k^H\mathbf{e}_k}{\left\|\tilde{\mathbf{C}}_0\right\|_F^2 + \sum_{k=1}^N \mu_k\mathbf{v}_k^H\mathbf{v}_k} \le \gamma^2 \quad (20)$$

with $\gamma = 1$ iff $0 \le \mu_k \le \bar{\mu}_k$. Further properties are:

- *The algorithm behaves robust for $\bar{\mu}_k \le \mu_k \le 2\bar{\mu}_k$ with $1 < \gamma < \infty$.*
- *If $\sum_{k=1}^N \|\mathbf{v}_k\|_2^2 < V_o < \infty$, the undistorted error vectors $\mathbf{e}_k \to \mathbf{0}$.*
- *If furthermore the sequence $\{\mathbf{x}_k\}$ is persistently exciting, the system error matrix $\tilde{\mathbf{C}}_k \to \mathbf{O}$.*

5. LMS FOR MIMO SYSTEMS WITH KRONECKER PRODUCT STRUCTURE

We now derive an LMS algorithm with substantially less complexity, taking advantage of the tensor product of $\mathbf{C}$. In particular we take advantage of the identity

$$\mathbf{C} = \mathbf{B} \otimes \mathbf{A} = (\mathbf{I}_{P_1} \otimes \mathbf{A})(\mathbf{B} \otimes \mathbf{I}_{N_2}) = (\mathbf{B} \otimes \mathbf{I}_{N_1})(\mathbf{I}_{P_2} \otimes \mathbf{A}) \quad (21)$$

which allows to partition the algorithm. Here, $(\mathbf{I}_{P_1} \otimes \mathbf{A}) \in \mathbb{C}^{P_1 N_1 \times P_1 N_2}$ and $(\mathbf{B} \otimes \mathbf{I}_{N_2}) \in \mathbb{C}^{P_1 N_2 \times N_2 P_2}$ whereas $(\mathbf{B} \otimes \mathbf{I}_{N_1}) \in \mathbb{C}^{P_1 N_1 \times P_2 N_1}$ and $(\mathbf{I}_{P_2} \otimes \mathbf{A}) \in \mathbb{C}^{N_1 P_2 \times P_2 N_2}$. We then reformulate the matrix vector product into equivalent forms

$$\begin{aligned} \mathbf{C}\mathbf{x}_k &= (\mathbf{I}_{P_1} \otimes \mathbf{A})(\mathbf{B} \otimes \mathbf{I}_{N_2})\mathbf{x}_k = (\mathbf{I}_{P_1} \otimes \mathbf{A})\mathbf{z}_k & (22) \\ &= (\mathbf{B} \otimes \mathbf{I}_{N_1})(\mathbf{I}_{P_2} \otimes \mathbf{A})\mathbf{x}_k = (\mathbf{B} \otimes \mathbf{I}_{N_1})\mathbf{u}_k. & (23) \end{aligned}$$

The so obtained vectors $\mathbf{z}_k \in \mathbb{C}^{P_1 N_2 \times 1}$ and $\mathbf{u}_k \in \mathbb{C}^{P_2 N_1 \times 1}$.

With these useful identities, we can now derive a gradient type algorithm in terms of the regression vector $\mathbf{z}_k$ to update the coefficients in $\mathbf{A}$ and in terms of $\mathbf{u}_k$ to update the coefficients in $\mathbf{B}$. Corresponding to LMS update (16), we now find the following

$$\mathbf{I}_{P_1} \otimes \hat{\mathbf{A}}_k = \mathbf{I}_{P_1} \otimes \hat{\mathbf{A}}_{k-1} + \mu_{A,k}[\mathbf{y}_k - (\mathbf{I}_{P_1} \otimes \hat{\mathbf{A}}_{k-1})\hat{\mathbf{z}}_k]\hat{\mathbf{z}}_k^H \quad (24)$$

$$\hat{\mathbf{B}}_k \otimes \mathbf{I}_{N_1} = \hat{\mathbf{B}}_{k-1} \otimes \mathbf{I}_{N_1} + \mu_{B,k}[\mathbf{y}_k - (\hat{\mathbf{B}}_{k-1} \otimes \mathbf{I}_{N_1})\hat{\mathbf{u}}_k]\hat{\mathbf{u}}_k^H \quad (25)$$

with the estimates $\hat{\mathbf{u}}_k = (\mathbf{I}_{P_2} \otimes \hat{\mathbf{A}}_{k-1})\mathbf{x}_k$ and $\hat{\mathbf{z}}_k = (\hat{\mathbf{B}}_{k-1} \otimes \mathbf{I}_{N_2})\mathbf{x}_k$. Note that the two update partitions are not equivalent to the original LMS algorithm as they minimize different cost functions

$$\begin{aligned} \mathbf{A}_o &= \arg\min_{\mathbf{A}} E[\tilde{\mathbf{e}}_{A,k}^H \tilde{\mathbf{e}}_{A,k}], & (26) \\ \tilde{\mathbf{e}}_{A,k} &= \mathbf{y}_k - (\mathbf{I}_{P_1} \otimes \hat{\mathbf{A}}_{k-1})\hat{\mathbf{z}}_k, & (27) \\ \mathbf{B}_o &= \arg\min_{\mathbf{B}} E[\tilde{\mathbf{e}}_{B,k}^H \tilde{\mathbf{e}}_{B,k}], & (28) \\ \tilde{\mathbf{e}}_{B,k} &= \mathbf{y}_k - (\hat{\mathbf{B}}_{k-1} \otimes \mathbf{I}_{N_1})\hat{\mathbf{u}}_k. & (29) \end{aligned}$$

The error vectors $\tilde{\mathbf{e}}_{A,k}$ and $\tilde{\mathbf{e}}_{B,k}$ are conditioned on $\hat{\mathbf{B}}_{k-1}$ and $\hat{\mathbf{A}}_{k-1}$, respectively. They are only equivalent if the corresponding other term $\hat{\mathbf{B}}_{k-1} = \mathbf{B}$ and $\hat{\mathbf{A}}_{k-1} = \mathbf{A}$.

Note further that the LMS updates in the form of (24) and (25) do not exhibit the full potential of the algorithm. As the terms $\mathbf{A}$ as well as $\mathbf{B}$ appear multiple times in the updates, the estimates can be further improved by averaging them and at the same time complexity can be saved. For this we need to partition block-wise the error vector $\tilde{\mathbf{e}}_k \in \mathbb{C}^{P_1 N_1 \times 1}$ into P_1 pieces of length N_1 each: $\bar{\mathbf{e}}_{l,k} = \tilde{\mathbf{e}}_{(l-1)N_1+1 \ldots lN_1,k}$ for $l = 1, 2, ..., P_1$. Correspondingly we partition also $\bar{\mathbf{z}}_{lk} = \hat{\mathbf{z}}_{(l-1)N_2+1 \ldots lN_2,k}$ for $l = 1, 2, ..., P_1$ and $\bar{\mathbf{u}}_{lk} = \hat{\mathbf{u}}_{(l-1)N_1+1 \ldots lN_1,k}$ for $l = 1, 2, ..., P_2$. We then find:

$$\begin{aligned} \hat{\mathbf{A}}_k &= \hat{\mathbf{A}}_{k-1} + \frac{\mu_{A,k}}{P_1}\mathbf{E}_k \mathbf{Z}_k^H & (30) \\ \hat{\mathbf{B}}_k &= \hat{\mathbf{B}}_{k-1} + \frac{\mu_{B,k}}{N_1}\mathbf{E}_k^T \mathbf{U}_k^* & (31) \end{aligned}$$

where we introduced the following matrices

$$\begin{aligned} \mathbf{E}_k &= [\bar{\mathbf{e}}_{1,k}, \bar{\mathbf{e}}_{2,k}, ..., \bar{\mathbf{e}}_{P_1,k}] \quad \in \mathbb{C}^{N_1 \times P_1} \\ \mathbf{U}_k &= [\bar{\mathbf{u}}_{1,k}, \bar{\mathbf{u}}_{2,k}, ..., \bar{\mathbf{u}}_{P_2,k}]; \quad \mathbf{Z}_k = [\bar{\mathbf{z}}_{1,k}, \bar{\mathbf{z}}_{2,k}, ..., \bar{\mathbf{z}}_{P_1,k}] \end{aligned}$$

Updates (30) and (31) are just a first straightforward form that simply averages all estimates. More sophisticated forms can be considered in which for example the quality of the a-posteriori errors is also taken into account to decide which terms were more hampered by noise than others. Nevertheless, for our further investigations we will remain with this simple update rule obtained by averaging.

Complexity: We first access the complexity of the standard LMS algorithm with update (16) to have a reference. For the computation of the error vector $\tilde{\mathbf{e}}_k$ we require $M_1 M_2$ MAC operations and the same complexity is required for the updates, thus the overall complexity is $2M_1 M_2$ per update step.

For the Kronecker based updates we first have to compute vectors $\hat{\mathbf{u}}_k$ and $\hat{\mathbf{z}}_k$ requiring $N_1 N_2 P_1$ and $P_1 P_2 N_1$, respectively, thus $M_1(N_2 + P_2)$ together. To compute the error costs additional $N_1 P_1 \min(N_2, P_2)$ are required depending on whether we compute the error based on $\hat{\mathbf{u}}_k$ or $\hat{\mathbf{z}}_k$. Finally for the updates of $\hat{\mathbf{A}}_k$ we need $N_1 N_2 P_1$ operations and for $\hat{\mathbf{B}}_k$ we need $N_1 P_1 P_2$ operations, all together $M_1[\min(N_2, P_2) + 2(N_2 + P_2)]$ which becomes substantially less compared to the standard LMS algorithm when the matrices become large.

6. PERFORMANCE

In this section we run a Monte Carlo (MC) experiment with a structured MIMO system $\mathbf{C}$ of dimension $M_1 = 50 = 10 \times 5$ and $M_2 = 21 = 3 \times 7$, whose entries are complex-valued Gaussian distributed and $\mathbf{C}$ is normalized by its Frobenius norm. We employ Gaussian input symbols of unit variance and additive noise of variance $\sigma_v^2 = 0.001$. We compare the standard LMS for the 50×21 matrix with the proposed tensor LMS algorithm by computing the relative system mismatch $\|\mathbf{C} - \hat{\mathbf{C}}_k\|_F^2 / \|\mathbf{C}\|_F^2$. Fig. 1 depicts the results for the standard matrix LMS algorithm after 20 MC runs. As expected for a normalized step-size $\mu_k = \alpha / \|\mathbf{x}_k\|_2^2$ we find fastest convergence at $\alpha = 1$ and stability bound at $\alpha = 2$.

Fig. 2 exhibits the results for the tensor LMS algorithm. Here, we applied the normalization $\mu_{A,k} = \alpha P_1 / \|\hat{\mathbf{z}}_k\|_2^2, \mu_{B,k} = \alpha N_1 / \|\hat{\mathbf{u}}_k\|_2^2$. As we have now substantially less parameters to estimate, we can learn faster and with higher precision as shown by the smaller steady-state values. Note that we have not imposed the uniqueness constraint on $\mathbf{A}$ as we were only interested in the resulting matrix $\mathbf{C}$. A straightforward normalization would substantially add on the complexity. However, there are alternative solutions possible, see e.g. [27, 28]. Due to space constraints the algorithm is not presented here in detail but the procedure can be downloaded from our web page `https://www.nt.tuwien.ac.at/downloads/featured-downloads`

7. CONCLUSIONS

The presented concept is just a first step into the identification of Kronecker structured MIMO systems. While the concept shows large potential to identify the system with less complexity and higher accuracy, many questions about its robustness and the prediction of learning rate and achieved steady-state values remain open.

8. REFERENCES

[1] Viktor Mayer-Schoenberger and Kenneth Cukier, *Big Data: A Revolution That Will Transform How We Live, Work and Think*, John Murray, 2013.

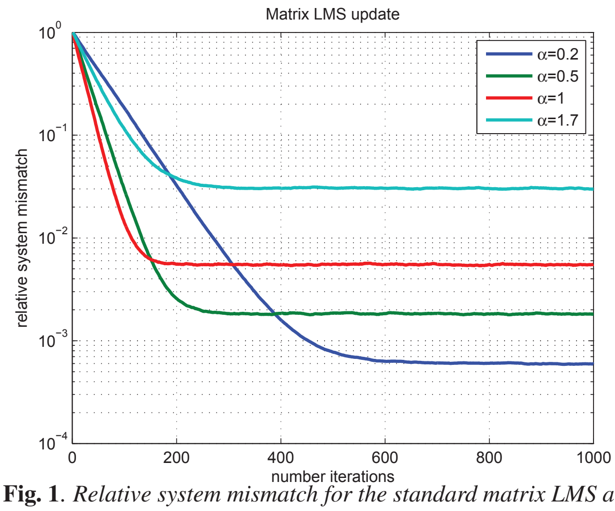

Fig. 1. *Relative system mismatch for the standard matrix LMS algorithm.*

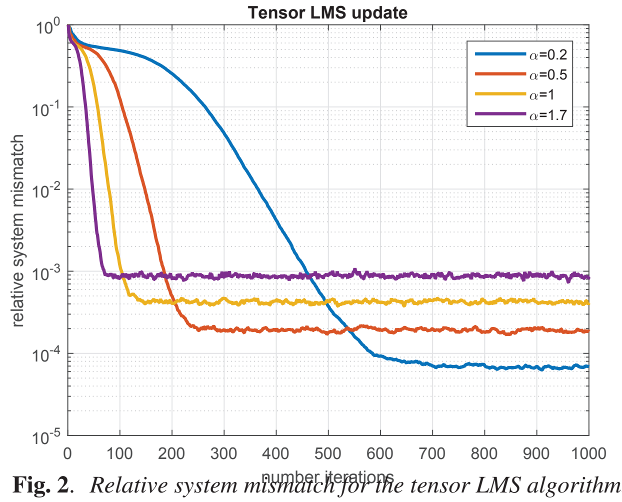

Fig. 2. *Relative system mismatch for the tensor LMS algorithm.*

[2] Hao Yin, Yong Jiang, Chuang Lin, Yan Luo, and Yunjie Liu, "Big Data: Transforming the design philosophy of future Internet," *IEEE Network*, pp. 14–18, July/Aug. 2014.

[3] Todd K. Moon and Wynn C. Stirling, *Mathematical Methods and Algorithms for Signal Processing*, Prentice Hall, 2000.

[4] LR Tucker, "Some mathematical notes on three-mode factor analysis," *Psychometrika*, vol. 31, pp. 279–311, 1966.

[5] J. Carroll and J. Chang, "Analysis of individual differences in multidimensional scaling via an n-way generalization of Eckart-Young decomposition," *Psychometrika*, vol. 35, no. 3, pp. 283–319, 1970.

[6] R. Harshman, "Foundations of the PARAFAC procedure: Models and conditions for an explanatory multimodal factor analysis," *UCLA Working Papers in Phonetics*, vol. 16, pp. 1–84, 1970.

[7] R. Harshman, "Determination and proof of minimum uniqueness conditions for PARAFAC-1," *UCLA Working Papers in Phonetics*, vol. 22, pp. 111–117, 1972.

[8] H.A.L. Kiers, "Towards a standardized notation and terminology in multiway analysis," *Chemometrics*, vol. 14, no. 2, pp. 105–122, 2000.

[9] N. D. Sidiropoulos, R. Bro, and G. B. Giannakis, "Parallel factor analysis in sensor array processing," *IEEE Transactions on Signal Processing*, vol. 48, no. 8, pp. 2377–2388, Aug. 2000.

[10] N.D. Sidiropoulos and A.Kyrillidis, "Multi-way compressed sensing for sparse low-rank tensors," *IEEE Signal Processing Letters*, vol. 19, no. 11, pp. 757–760, Nov. 2012.

[11] E. Candes, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Commun. Pure Appl. Math.*, vol. 59, no. 8, pp. 1207–1223, 2006.

[12] E. Candès and B. Recht, "Exact matrix completion via convex optimization," *Found. Comput. Math.*, vol. 9, no. 6, pp. 717–772, 2009.

[13] T. Kolda and J. Sun, "Scalable tensor decompositions for multi-aspect data mining," in *ICDM 2008: Proc. 8th IEEE Int. Conf. Data Mining*, 2008, p. 363–372.

[14] Pierre Comon, "Tensor decompositions: state of the art and applications," in *Mathematics in Signal Processing V*. 2002, p. 1–24, Clarendon Press, Oxford.

[15] T.G. Kolda and B.W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, 2008.

[16] Pierre Comon, "Tensors – a brief introduction," *IEEE Signal Processing Magazine*, pp. 44–53, May 2014.

[17] Guoxu Zhou et al., "Nonnegative matrix and tensor factorizations–an algorithmic perspective," *IEEE Signal Processing Magazine*, pp. 54–65, May 2014.

[18] Gerard Favier and Andre de Almeida, "Overview of constrained PARAFAC models," *EURASIP JASP*, vol. 142, 2014.

[19] A. et al. Cichocki, "Tensor decompositions for signal processing applications: From two-way to multiway component analysis," *IEEE Signal Processing Magazine*, vol. 32, no. 2, pp. 145–163, March 2015.

[20] C. van Loan and N. Pitsianis, "Approximation with Kronecker products," in *Linear Algebra for Large Scale and Real Time Applications*. 1993, p. 293–314, Norwell, MA: Kluwer.

[21] Markus Rupp and Stefan Schwarz, "A tensor LMS algorithm," in *ICASSP*, Brisbane, Australia, Apr. 2015.

[22] Ali H. Sayed and Markus Rupp, "Error-energy bounds for adaptive gradient algorithms," *IEEE Transactions on Signal Processing*, vol. 44, no. 8, pp. 1982–1989, Aug. 1996.

[23] Markus Rupp and Ali H. Sayed, "A time-domain feedback analysis of filtered-error adaptive gradient algorithms," *IEEE TSP*, vol. 44, no. 6, pp. 1428–1439, June 1996.

[24] Ali H. Sayed and Markus Rupp, "Robustness issues in adaptive filtering," in *The Digital Signal Processing Handbook*, chapter 20. CRC Press, Boca Raton, FL, USA, Jan. 1998.

[25] Ali H. Sayed, *Fundamentals of Adaptive Filtering*, John Wiley & Sons, Inc., Hoboken, NJ, USA, 2003.

[26] Robert Dallinger and Markus Rupp, "A strict stability limit for adaptive gradient type algorithms," in *Record 43rd ACSSC*, Pacific Grove, CA, USA, Nov. 2009, pp. 1370–1374.

[27] S.C. Douglas and M.Rupp, "On bias removal and unit norm constraints in equation error adaptive IIR filters," in *30th. Asilomar Conference*, Monterey, Nov. 1996, pp. 1093–1097.

[28] M.Rupp and S.C. Douglas, "Deterministic stability analyses of unit-norm constraint algorithms for unbiased adaptive IIR filtering," in *ICASSP*, Germany, Apr. 1997, pp. 1937–1940.