Low-dimensional Subspace Regularization through Structured Tensor Priors

Batselier, Kim

DOI
10.1137/24M1688497

Publication date
2025

Document Version
Final published version

Published in
SIAM-ASA Journal on Uncertainty Quantification

Citation (APA)
Batselier, K. (2025). Low-dimensional Subspace Regularization through Structured Tensor Priors. *SIAM-ASA Journal on Uncertainty Quantification*, *13*(3), 1452-1474. https://doi.org/10.1137/24M1688497

SIAM/ASA J. Uncertainty Quantification
Vol. 13, No. 3, pp. 1452–1474

Low-dimensional Subspace Regularization through Structured Tensor Priors[*]

Kim Batselier[†]

Abstract. Specifying a prior distribution is an essential part of solving Bayesian inverse problems. The prior encodes a belief on the nature of the solution and this regularizes the problem. In this article we completely characterize a Gaussian prior that encodes the belief that the solution is a structured tensor that lies in a low-dimensional subspace. We define the notion of $(\boldsymbol{A},\boldsymbol{b})$-constrained tensors and show that they describe a large variety of different structures such as Hankel, circulant, triangular, symmetric, and so on. We prove that the low-dimensional subspace defined by this prior is the right nullspace of the matrix $\boldsymbol{A}$ that defines the tensor structure. We completely characterize the Gaussian probability distribution of such tensors by specifying its mean vector and covariance matrix in terms of $\boldsymbol{A}$ and $\boldsymbol{b}$. Furthermore, explicit expressions are proved for the covariance matrix of tensors whose entries are invariant under a permutation. These results unlock a whole new class of priors for Bayesian inverse problems. We illustrate how new kernel functions can be designed and efficiently computed and apply our results on two particular Bayesian inverse problems: completing a Hankel matrix from a few noisy measurements and learning an image classifier of handwritten digits. The effectiveness of the proposed priors is demonstrated for both problems. All applications have been implemented as reactive Pluto notebooks in Julia.

Key words. Bayesian inverse problems, structured tensors, tensors, kernel methods

MSC codes. 15A29, 15A69, 62F15

DOI. 10.1137/24M1688497

1. Introduction.

We consider a set of data samples $\{(\boldsymbol{x}_n, y_n)\,|\,\boldsymbol{x}_n \in \mathbb{R}^D,\, y_n \in \mathbb{R}\}_{n=1}^N$ and the following linear forward model:

$$y_n = \langle \boldsymbol{\mathcal{P}}(\boldsymbol{x}_n), \boldsymbol{\mathcal{W}}\rangle + \epsilon_n. \tag{1.1}$$

Each scalar measurement y_n is obtained from an inner product of a data-dependent tensor $\boldsymbol{\mathcal{P}}(\boldsymbol{x}_n) \in \mathbb{R}^{J_1\times\cdots\times J_D}$ with a tensor of unknown latent variables $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{J_1\times\cdots\times J_D}$, corrupted by measurement noise ϵ_n. Tensors in this context are D-dimensional arrays, with vectors $(D=1)$ and matrices $(D=2)$ being the most well-known cases. Vectorizing all tensors and collecting the measurements $y_1,\ldots,y_N$ into a vector $\boldsymbol{y}\in\mathbb{R}^N$ allows (1.1) to be rewritten into the linear system of equations

$$\boldsymbol{y} = \boldsymbol{\Phi}(\boldsymbol{x})\,\boldsymbol{w} + \boldsymbol{\epsilon}. \tag{1.2}$$

Row n of the matrix $\boldsymbol{\Phi}(\boldsymbol{x}) \in \mathbb{R}^{N\times J_1\cdots J_D}$ contains the vectorization of the tensor $\boldsymbol{\mathcal{P}}(\boldsymbol{x}_n)$. For notational convenience the indication that $\boldsymbol{\Phi}$ depends on $\boldsymbol{x}$ is dropped from here on.

[*] Received by the editors August 27, 2024; accepted for publication (in revised form) June 7, 2025; published electronically August 29, 2025.
https://doi.org/10.1137/24M1688497
Funding: This work was supported by the project Sustainable Learning for Artificial Intelligence from noisy large-scale data (with project number VI.Vidi.213.017) which is financed by the Dutch Research Council (NWO).

[†] Delft Center for Systems and Control, Delft University of Technology, The Netherlands (k.batselier@tudelft.nl).

The inverse problem consists of inferring the latent variables $\boldsymbol{w}$ from the noisy measurements $\boldsymbol{y}$. In this article a Bayesian approach [2] is considered by assuming that $\boldsymbol{w}$ and $\boldsymbol{\epsilon}$ are random variables. The goal is then to infer the posterior distribution $p(\boldsymbol{w}|\boldsymbol{y})$ of $\boldsymbol{w}$ conditioned on the measurements $\boldsymbol{y}$ using Bayes' theorem

$$p(\boldsymbol{w}|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{w})\, p(\boldsymbol{w})}{p(\boldsymbol{y})}.$$

The distribution $p(\boldsymbol{w})$ is called the prior and encodes a belief on what $\boldsymbol{w}$ is before the measurements are known. The main contribution of this article is the complete characterization of a prior $p(\boldsymbol{w})$ that encodes the belief that the corresponding tensor $\boldsymbol{\mathcal{W}}$ is structured. A Gaussian distribution is assumed for the noise distribution $p(\boldsymbol{\epsilon}) = \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ with mean vector $\mathbf{0}$ and covariance matrix $\boldsymbol{\Sigma}$ and likewise for the prior $p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}_0, \boldsymbol{P}_0)$. The linear forward model (1.2) combined with the Gaussian assumptions results in a Gaussian posterior $p(\boldsymbol{w}|\boldsymbol{y}) = \mathcal{N}(\boldsymbol{w}_+, \boldsymbol{P}_+)$ with mean vector $\boldsymbol{w}_+$ and covariance matrix $\boldsymbol{P}_+$

$$\boldsymbol{w}_+ = (\boldsymbol{P}_0^{-1} + \boldsymbol{\Phi}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\Phi})^{-1} (\boldsymbol{\Phi}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{y} + \boldsymbol{P}_0^{-1} \boldsymbol{w}_0), \tag{1.3}$$

$$\boldsymbol{P}_+ = (\boldsymbol{P}_0^{-1} + \boldsymbol{\Phi}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\Phi})^{-1}. \tag{1.4}$$

The role of the prior $p(\boldsymbol{w})$ can now be understood from (1.3) and (1.4). In the absence of data ($\boldsymbol{\Phi} = \mathbf{0}$ and $\boldsymbol{y} = \mathbf{0}$) the posterior equals the prior. In other words, the prior encodes a belief on what the solution $\boldsymbol{w}$ of (1.2) should be before any data is known. A natural question to ask is then what kind of prior to use. In this article we consider a prior encoding the belief that the tensor $\boldsymbol{\mathcal{W}}$ has a structure that is completely determined by a matrix $\boldsymbol{A} \in \mathbb{R}^{I \times J_1 \cdots J_D}$ and vector $\boldsymbol{b} \in \mathbb{R}^I$ such that

$$\boldsymbol{A} \operatorname{vec}(\boldsymbol{\mathcal{W}}) = \boldsymbol{b} \text{ and } \operatorname{rank}(\boldsymbol{A}) < \min(I, J_1 \cdots J_D), \tag{1.5}$$

which we will refer to as $(\boldsymbol{A}, \boldsymbol{b})$-constrained tensors. In other words, every prior sample $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{J_1 \times \cdots \times J_D}$ satisfies (1.5). We show in section 2.1 that each sample $\boldsymbol{\mathcal{W}}$ of the prior lies in a low-dimensional subspace. In this way, these priors can regularize the inverse problem by limiting the "effective" number of posterior latent variables $\boldsymbol{\mathcal{W}}$ to the nullity R of $\boldsymbol{A}$. Or in other words, these priors favor samples of the posterior to lie in some R-dimensional subspace of $\mathbb{R}^{J_1 \cdots J_D}$, where typically $R \ll J_1 \cdots J_D$. In this sense, these priors can also be interpreted as imposing a kind of low-rank constraint on $\boldsymbol{\mathcal{W}}$. Inverse problems that can benefit from this kind of low-dimensional regularization appear in many different applications fields such as machine learning [9, 34, 35, 39, 40] control [4, 6, 29, 33] and signal processing [14, 17, 19, 21, 26, 27, 38]. Consider for example the weight-space view of Gaussian processes [40]. In this case the data-dependent tensor $\boldsymbol{\mathcal{P}}(\boldsymbol{x}_n)$ can be thought of as the evaluation of an exponential amount of D-variate basis functions. One can assume that not all exponentially many variables in $\boldsymbol{\mathcal{W}}$ will be equally important but lie in some low-dimensional subspace instead. The effectiveness of this regularization is demonstrated in Application 6.2, completing a noisy Hankel matrix, and Application 6.3, learning an image classifier for handwritten digits. Our proposed prior, which limits the number of effective latent variables, consistently outperforms the max-likelihood estimate or the commonly-used Tikhonov prior, which assumes that $\boldsymbol{\mathcal{W}}$ lives in the whole vector space $\mathbb{R}^{J_1 \times \cdots \times J_D}$. The contributions of this article are threefold.

1. We show how the definition of $(\boldsymbol{A}, \boldsymbol{b})$-constrained tensors is well-motivated since it encompasses a wide variety of application-relevant structured tensors. Examples are given for tensors with fixed entries, tensors with known sums of entries and symmetric, Hankel, Toeplitz, circulant, and triangular tensors.
2. In Lemma 2.1 we completely characterize the mean vector $\boldsymbol{w}_0$ and covariance matrix $\boldsymbol{P}_0$ of the prior $p(\boldsymbol{w})$ for $(\boldsymbol{A}, \boldsymbol{b})$-constrained tensors, which leads to the interpretation of prior samples living in a (typically) low-dimensional subspace.
3. In Theorems 4.5 and 4.13 we provide explicit expressions for $\boldsymbol{P}_0$ for $(\boldsymbol{A}, \boldsymbol{b})$-constrained tensors whose entries remain invariant under a permutation $\boldsymbol{P}$. Such tensors will be called $\boldsymbol{P}$-invariant or skew-$\boldsymbol{P}$-invariant.

These three contributions are important because the prior mean $\boldsymbol{w}_0$ and covariance matrix $\boldsymbol{P}_0$ are necessary to solve the Bayesian inverse problem via equations (1.3) and (1.4). Contrary to most solution strategies for linear least squares problems the matrix inverse of $\boldsymbol{P}_0^{-1} + \boldsymbol{\Phi}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\Phi}$ is explicitly required as it forms the posterior covariance. Also note that the dimension of the matrix to invert is $J_1 J_2 \ldots J_D \times J_1 J_2 \ldots J_D$, which limits the use of direct solvers to cases of small J and D. Hybrid projection methods [10, 11] and randomized solvers [41] are a viable alternative for cases where J and D are prohibitively large. Another alternative is to solve the corresponding dual problem, which is described in terms of the so-called kernel matrix $\boldsymbol{\Phi} \boldsymbol{P}_0 \boldsymbol{\Phi}^T \in \mathbb{R}^{N \times N}$. This approach is commonly used in least-squares support vector machines [35] and Gaussian processes [40] and has a computational complexity of at least $O(N^2)$. When the tensor $\boldsymbol{\mathcal{P}}(\boldsymbol{x}_n)$ exhibits a low-rank structure, then another way to obtain low computational complexity of solving (1.3) is by imposing a low-rank tensor structure to $\boldsymbol{w}_+$ and $\boldsymbol{P}_+$ [6, 28, 34]. Imposing a low-rank tensor structure onto $\boldsymbol{\mathcal{W}}$ is another way of restricting the number of effective latent variables. Developing dedicated solution strategies for equations (1.3) and (1.4) lies outside the scope of this article as they are application-specific. Also note that the results of this article can be trivially extended to vector-valued observations through a matrix $\boldsymbol{W}$ of latent variables in (1.2). Out of notational convenience, we only consider scalar-valued observations throughout the article, but we consider a problem with vector-valued observations in Application 6.3.

1.1. Notation and tensor definitions.

Tensors in this article are multidimensional arrays with real entries. We denote scalars by italic letters $a, b, \ldots$, vectors by boldface italic letters $\boldsymbol{a}, \boldsymbol{b}, \ldots$, matrices by boldface capitalized italic letters $\boldsymbol{A}, \boldsymbol{B}, \ldots$ and higher-order tensors by boldface calligraphic italic letters $\boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}}, \ldots$. The vector $\boldsymbol{e}_j \in \mathbb{R}^J$ denotes a canonical basis vector that has a single nonzero unit entry at position j. The vector $\boldsymbol{1}_J \in \mathbb{R}^J$ denotes a vector of ones, and $\boldsymbol{I}_J \in \mathbb{R}^{J \times J}$ is the unit matrix. The number of indices required to determine an entry of a tensor is called the order of the tensor. A Dth-order or D-way tensor is hence denoted $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_D}$. An index j_d always satisfies $1 \leq j_d \leq J_d$, where J_d is called the dimension of that particular mode. Tensor entries of a tensor $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_D}$ are denoted $w_{j_1, j_2, \ldots, j_D}$. Tensors can be reshaped into tensors of lower order by combining indices. Combining a set of d indices $j_1, j_2, \ldots, j_d$ results in a single index $\overline{j_1 j_2 \ldots j_d}$ that satisfies

$$\overline{j_1 j_2 \ldots j_d} = j_1 + (j_2 - 1)\, J_1 + \cdots + (j_d - 1) J_1 \cdots J_{d-1}.$$

The vectorization of a tensor reshapes all entries of a tensor $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_D}$ into a single vector $\boldsymbol{w} = \text{vec}\,(\boldsymbol{\mathcal{W}})$ such that $w_{\overline{j_1 j_2 \cdots j_d}} = w_{j_1, j_2, \ldots, j_D}$. For a tensor $\boldsymbol{\mathcal{W}}$, we will always assume that the corresponding vector $\boldsymbol{w} = \text{vec}(\boldsymbol{\mathcal{W}})$. A more detailed introduction to tensor basics can be found in [22]. The square root matrix $\sqrt{\boldsymbol{P}_0}$ of a square symmetric positive-semidefinite matrix $\boldsymbol{P}_0$ is defined as any matrix that satisfies $\boldsymbol{P}_0 = \sqrt{\boldsymbol{P}_0}\,(\sqrt{\boldsymbol{P}_0})^T$. A permutation matrix $\boldsymbol{P}$ is a square binary matrix that has exactly one entry of 1 in each row and each column with all other entries 0.

2. Full characterization of the prior distribution.

In this section the Gaussian prior $p(\boldsymbol{w})$ for $(\boldsymbol{A}, \boldsymbol{b})$-constrained tensors is fully characterized. We also discuss how the square root covariance matrix $\sqrt{\boldsymbol{P}_0}$ can be computed without explicitly constructing the matrix $\boldsymbol{A}$ through a block-row partitioning of $\boldsymbol{A}$ and how $\sqrt{\boldsymbol{P}_0}$ can be parameterized in terms of covariances of underlying random variables.

Lemma 2.1. *The Gaussian distribution of $(\boldsymbol{A}, \boldsymbol{b})$-constrained tensors $\mathcal{N}(\boldsymbol{w}_0, \boldsymbol{P}_0)$ is described by a mean vector $\boldsymbol{w}_0$ such that $\boldsymbol{A}\,\boldsymbol{w}_0 = \boldsymbol{b}$ and by a covariance matrix $\boldsymbol{P}_0$ such that the columns of $\sqrt{\boldsymbol{P}_0}$ span the right nullspace of $\boldsymbol{A}$.*

Proof. Let $\boldsymbol{x}$ be a sample of the standard normal distribution $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. A sample $\boldsymbol{w}$ of the desired Gaussian distribution is then

$$\boldsymbol{w} = \boldsymbol{w}_0 + \sqrt{\boldsymbol{P}_0}\,\boldsymbol{x} = \boldsymbol{w} = \begin{pmatrix} \boldsymbol{w}_0 & \sqrt{\boldsymbol{P}_0} \end{pmatrix} \begin{pmatrix} 1 \\ \boldsymbol{x} \end{pmatrix}, \tag{2.1}$$

where $\sqrt{\boldsymbol{P}_0}$ is the matrix square root of the covariance matrix $\boldsymbol{P}_0$. Any sample $\boldsymbol{w}$ being an $(\boldsymbol{A}, \boldsymbol{b})$-constrained tensor implies $\boldsymbol{A}\,\boldsymbol{w} = \boldsymbol{A}\,\boldsymbol{w}_0 + \boldsymbol{A}\,\sqrt{\boldsymbol{P}_0}\,\boldsymbol{x} = \boldsymbol{b}$, which can only be true for all random samples $\boldsymbol{x}$ if and only if $\boldsymbol{A}\,\boldsymbol{w}_0 = \boldsymbol{b}$ and $\boldsymbol{A}\,\sqrt{\boldsymbol{P}_0} = \boldsymbol{0}$. In other words, the mean $\boldsymbol{w}_0$ of the prior also has to satisfy the linear constraints, and the columns of $\sqrt{\boldsymbol{P}_0}$ span the right nullspace of $\boldsymbol{A}$. ■

From Lemma 2.1, we can deduce that the linear constraints $\boldsymbol{A}\boldsymbol{w} = \boldsymbol{b}$ restrict the dimension of the vector space of prior samples $\boldsymbol{w}$.

Corollary 2.2. *Let R be the nullity of $\boldsymbol{A}$ and $\boldsymbol{b} \neq \boldsymbol{0}$, then each sample of the $(\boldsymbol{A}, \boldsymbol{b})$-constrained tensor prior $\mathcal{N}(\boldsymbol{w}_0, \boldsymbol{P}_0)$ lies in a $(R+1)$-dimensional subspace of $\mathbb{R}^{J_1 \cdots J_D}$.*

Each sample $\boldsymbol{w}$ lies in the column space of the matrix $\begin{pmatrix} \boldsymbol{w}_0 & \sqrt{\boldsymbol{P}_0} \end{pmatrix}$ and consists of a constant one-dimensional part $\boldsymbol{w}_0$ and a variable R-dimensional part $\sqrt{\boldsymbol{P}_0}$. When $\boldsymbol{b} = \boldsymbol{0}$ then the prior mean $\boldsymbol{w}_0$ lies in the right nullspace of $\boldsymbol{A}$ and the dimension of the subspace is reduced to R. From (1.4) we see that the posterior covariance $\boldsymbol{P}_+$ depends only on $\boldsymbol{P}_0$ and not on the prior mean $\boldsymbol{w}_0$. The prior therefore regularizes the inverse problem by restricting the posterior samples to lie in an R-dimensional subspace. If we interpret each of the I rows of $\boldsymbol{A}$ as enforcing a linear constraint on the entries of $\boldsymbol{\mathcal{W}}$, then Corollary 2.2 tells us that the more linearly independent constraints we enforce, the lower-dimensional the subspace of prior samples will be. In section 3 a few examples of possible constraints together with related application fields will be discussed.

Lemma 2.1 tells us that the desired prior square root covariance $\sqrt{\boldsymbol{P}_0}$ is found by computing a basis for the right nullspace of $\boldsymbol{A}$. When the matrix $\boldsymbol{A}$ is too large to construct explicitly then it can be beneficial to compute a basis for its right nullspace recursively.

This is possible when considering a partitioning into S block-rows $\boldsymbol{A} = \begin{pmatrix}\boldsymbol{A}_1^T & \boldsymbol{A}_2^T & \dots & \boldsymbol{A}_S^T\end{pmatrix}^T$, where each block-row $\boldsymbol{A}_s\,(s = 1,\dots,S)$ enforces its own set of linear constraints on $\boldsymbol{\mathcal{W}}$. A basic algorithm to compute a basis for the right nullspace recursively one block-row at a time can be found in Theorem 6.4.1 from [16, p. 329] with a computational complexity of at most $O(4J_1\cdots J_D\,(I/S)^2 + 8\,(I/S)^3)$.

2.1. Parameterizing the prior covariance matrix.

The covariance matrix $\boldsymbol{P}_0$ as described in Lemma 2.1 encodes the structure of $(\boldsymbol{A},\boldsymbol{b})$-constrained tensors without having any free parameters to quantify the importance of the prior $p(\boldsymbol{w})$ relative to the likelihood $p(\boldsymbol{y}|\boldsymbol{w})$. Such free parameters are often called hyperparameters. Suppose for example that through Lemma 2.1 an orthogonal basis for the nullspace $\boldsymbol{V}_2 \in \mathbb{R}^{J_1\cdots J_D \times R}$ of $\boldsymbol{A}$ is computed from its singular value decomposition (SVD)

$$\boldsymbol{A} = \begin{pmatrix}\boldsymbol{U}_1 & \boldsymbol{U}_2\end{pmatrix}\begin{pmatrix}\boldsymbol{S} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0}\end{pmatrix}\begin{pmatrix}\boldsymbol{V}_1^T\\ \boldsymbol{V}_2^T\end{pmatrix}.$$

The square-root covariance matrix $\sqrt{\boldsymbol{P}_0}$ is then $\boldsymbol{V}_2\boldsymbol{T}$, where $\boldsymbol{T} \in \mathbb{R}^{R\times R}$ is any invertible matrix. The $\boldsymbol{T}$ matrix can be interpreted as the square-root covariance matrix of R random variables $\tilde{\boldsymbol{x}} = \boldsymbol{T}\boldsymbol{x}$, with $\boldsymbol{x}$ the standard normal random variables of (2.1), since

$$\boldsymbol{P}_0 = \sqrt{\boldsymbol{P}_0}\,(\sqrt{\boldsymbol{P}_0})^T = \boldsymbol{V}_2\,\left(\boldsymbol{T}\boldsymbol{T}^T\right)\,\boldsymbol{V}_2^T.$$

The matrix $\boldsymbol{V}_2$ is then to be understood as projecting the covariance matrix $\boldsymbol{T}\boldsymbol{T}^T$ of the R underlying random variables $\tilde{\boldsymbol{x}}$ to the $J_1\cdots J_D$ entries of the $(\boldsymbol{A},\boldsymbol{b})$-constrained $\boldsymbol{\mathcal{W}}$ tensor. Parameterizing $\boldsymbol{T}$ in terms of a single hyperparameter $\sigma \in \mathbb{R}^+$ as $\boldsymbol{T} = \sigma\,\boldsymbol{I}$ implies that these R variables are independent and have equal variance σ^2. Correlations between the R variables can be modeled by for example parameterizing $\boldsymbol{T}$ as a lower triangular matrix. The values of these hyperparameters can be learned from data through cross-validation, marginal likelihood optimization, or a hierarchical Bayesian approach [35, 40].

3. $(\boldsymbol{A},\boldsymbol{b})$-constrained tensors.

After having completely characterized the Gaussian prior of $(\boldsymbol{A},\boldsymbol{b})$-constrained tensors in Lemma 2.1, the question still remains what are application-relevant choices for $\boldsymbol{A}$ and $\boldsymbol{b}$. In this section we demonstrate the breadth of $(\boldsymbol{A},\boldsymbol{b})$-constrained tensors through three particular examples and show that the definition of $(\boldsymbol{A},\boldsymbol{b})$-constrained tensors is well-motivated in that it captures a wide variety of application-relevant structured tensors.

3.1. Tensors with fixed entries.

A tensor $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{J_1\times J_2\times\cdots\times J_D}$ with I fixed entries can be described as $\boldsymbol{A}\boldsymbol{w} = \boldsymbol{b}$ where row i of the matrix $\boldsymbol{A} \in \mathbb{R}^{I\times J_1\cdots J_D}$ is a canonical basis vector $\boldsymbol{e}_{\overline{j_1\cdots j_D}}$ that selects the ith fixed entry $w_{j_1,\dots,j_D}$. The corresponding fixed numerical value of $w_{j_1,\dots,j_D}$ is then given by b_i.

Example 3.1. Suppose $\boldsymbol{W} \in \mathbb{R}^{2\times 2}$ and we fix the values of the entries $w_{1,1}$ and $w_{1,2} =$ to 1 and -4, respectively. The corresponding matrix equation is then

$$\begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0\end{pmatrix}\boldsymbol{w} = \begin{pmatrix}1\\ -4\end{pmatrix}.$$

The first row of $\boldsymbol{A}$ selects entry $w_{1,1}$ and is therefore the canonical basis vector $\boldsymbol{e}_{\overline{11}}$. Likewise, the second row of $\boldsymbol{A}$ is the canonical basis vector $\boldsymbol{e}_{\overline{12}}$.

Fixed values are commonly zero. In the context of system identification, such a prior can impose stability on the estimated model [37]. Triangular or banded structures are common in matrices, e.g., discrete convolution operators in signal processing. Such structures can also be generalized to the tensor case, where they are for example exploited for efficient simulation of nonlinear dynamic models [5].

Definition 3.2. *A tensor* $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_D}$ *is lower (upper) triangular when* $w_{j_1,j_2,\cdots,j_D} = 0$ *holds for each consecutive index pair* $j_d, j_{d+1}\,(d = 1, \ldots, D-1)$ *such that* $j_d - j_{d+1} < (>)\, 0$.

The characterization of a lower (upper) triangular tensor as an $(\boldsymbol{A}, \boldsymbol{b})$-constrained tensor can be built up by generalizing the matrix case to higher orders. In what follows, we assume that $J_1 = J_2 = \cdots = J_D = J$. Consider a lower (upper) triangular matrix $\boldsymbol{W} \in \mathbb{R}^{J \times J}$. The $\boldsymbol{A}$ matrix needs to select all upper (lower) triangular entries of $\boldsymbol{W}$, and $\boldsymbol{b}$ contains their numerical values, which are zero. The $I := J(J-1)/2$ index pairs (j_1, j_2) of all upper (lower) triangular entries of $\boldsymbol{W}$ satisfy $j_1 - j_2 < (>)\, 0$. Now let $\boldsymbol{L}$ be the $I \times J^2$ matrix such that row i has a single unit entry for the ith particular index pair that satisfies $j_1 - j_2 < (>)\, 0$:

$$l_{i,\overline{j_1,j_2}} = \begin{cases} 1, & \text{for the } i\text{th index pair } (j_1, j_2) \text{ that satisfies } j_1 - j_2 < (>)\, 0, \\ 0, & \text{otherwise }. \end{cases}$$

Then $\boldsymbol{L}\boldsymbol{w} = \boldsymbol{b}$ states that all upper (lower) triangular entries of $\boldsymbol{W}$ are zero. Extending this formulation to tensors of higher orders can be done by considering all consecutive index pairs as stated in the following lemma.

Lemma 3.3. *Let* $J_1 = J_2 = J$*, then there are* $I := J(J-1)/2$ *index pairs* (j_1, j_2) *such that* $j_1 - j_2 < (>)\, 0$. *Now let* $\mathbf{L}$ *be the* $I \times J^2$ *matrix that selects all upper (lower) triangular entries of a matrix* $\boldsymbol{W}$. *Lower (upper) triangular tensors are then described by*

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{L} \otimes \boldsymbol{I}_J \otimes \cdots \otimes \boldsymbol{I}_J \\ \boldsymbol{I}_J \otimes \boldsymbol{L} \otimes \cdots \otimes \boldsymbol{I}_J \\ \vdots \\ \boldsymbol{I}_J \otimes \boldsymbol{I}_J \otimes \cdots \otimes \boldsymbol{L} \end{pmatrix} \in \mathbb{R}^{\frac{(D-1)(J-1)J^{D-1}}{2} \times J^D},$$

where each block-row of $\boldsymbol{A}$ *contains a Kronecker product of* $D-2$ *identity matrices and* $\boldsymbol{b}$ *is a vector of zeros.*

The known fixed values of lower (upper) triangular tensors are zero and hence $\boldsymbol{b}$ is a vector of zeros. Each row of the matrix $\boldsymbol{A}$ has a single unit entry to select a particular tensor entry for which some consecutive indices $j_d, j_{d+1}(d = 1, \ldots, D-1)$ satisfy $j_d - j_{d+1} < (>)\, 0$. A tensor with D indices has $D-1$ consecutive index pairs, and $\boldsymbol{A}$ can be partitioned into $D-1$ block-rows of Kronecker products of $D-2$ identity matrices with $\boldsymbol{L}$. The first block-row selects all entries of $\boldsymbol{\mathcal{W}}$ with $j_1 - j_2 < (>) 0$, the second block-row selects all entries with $j_2 - j_3 < (>)\, 0$ and so on. In each block-row the $\boldsymbol{L}$ matrix factor only selects index pairs for which $j_d - j_{d+1} < (>)\, 0$ while the Kronecker product of the identity matrices generate all possible index combinations of the $D-2$ remaining index values. Each block-row consists of $(J-1)J^{D-1}/2$ rows which means that $D-1$ block rows results in a total amount of $(D-1)(J-1)J^{D-1}/2$ rows for $\boldsymbol{A}$. The $\boldsymbol{A}$ matrix that describes tensors with known fixed entries in Lemma 3.3 is sparse and structured into block-rows, which facilitates computing its nullspace recursively one block-row at a time as discussed in section 2.

Example 3.4. Consider a lower triangular tensor $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{3\times 3\times 3}$ with $D=3$ and $J_1 = J_2 = J_3 = 3$. The condition $j_d - j_{d+1} < 0$ occurs in three cases $(j_d, j_{d+1}) \in \{(1,2),(1,3),(2,3)\}$. Defining the matrix $\boldsymbol{L} \in \mathbb{R}^{3\times 9}$ with three nonzero entries $l_{1,\overline{12}} = l_{2,\overline{13}} = l_{3,\overline{23}} = 1$ allows us to describe the desired $\boldsymbol{A}$ matrix as

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{L} \otimes \boldsymbol{I}_3 \\ \boldsymbol{I}_3 \otimes \boldsymbol{L} \end{pmatrix} \in \mathbb{R}^{18\times 27}, \tag{3.1}$$

with 18 nonzero entries. The first block-row of $\boldsymbol{A}$ selects nine tensor entries $w_{1,2,j_3}, w_{1,3,j_3}, w_{2,3,j_3}$, with $1 \le j_3 \le 3$. The second block-row selects nine entries $w_{j_1,1,2}, w_{j_1,1,3}, w_{j_1,2,3}$, with $1 \le j_1 \le 3$.

3.2. Known sum of entries.

Tensors for which the sum over all or only particular entries add up to a known value are also quite common in applications. Stochastic tensors with applications in hypergraphs and hidden Markov models are a particular example [1, 15, 18, 25]. Knowing a particular sum of entries can be described as follows.

Definition 3.5. *The sum over an index set $\mathcal{J} \subseteq \{j_1, j_2, \ldots, j_D\}$ of a tensor $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{J_1\times\cdots\times J_D}$ is defined as $\boldsymbol{A}\,\boldsymbol{w} = \operatorname{vec}(\boldsymbol{\mathcal{B}})$ with $\boldsymbol{A} = \boldsymbol{A}_D \otimes \cdots \otimes \boldsymbol{A}_1$, where each matrix $\boldsymbol{A}_d\,(d = 1,\ldots,D)$ in the Kronecker product is defined as*

$$\boldsymbol{A}_d = \begin{cases} \boldsymbol{1}_{J_d}^T & \text{if } j_d \in \mathcal{J}, \\ \boldsymbol{I}_{J_d} & \text{if } j_d \notin \mathcal{J}. \end{cases} \tag{3.2}$$

The resulting tensor $\boldsymbol{\mathcal{B}}$ has indices $\{j_1, j_2, \ldots, j_D\} \setminus \mathcal{J}$.

Example 3.6. Let $\boldsymbol{W} \in \mathbb{R}^{2\times 3}$ be a matrix $(D=2, J_1 = 2, J_2 = 3)$ for which each row sum equals to 1. Therefore $\mathcal{J} = \{j_2\}$ and Lemma 3.5 then implies that

$$\boldsymbol{A} = \boldsymbol{1}_3^T \otimes \boldsymbol{I}_2 = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix},$$

and $\boldsymbol{b} = \boldsymbol{1}_2$, since each row sum equals to 1. Note that the resulting $\boldsymbol{b}$ vector is described by the index $\{j_1, j_2\} \setminus \{j_2\} = \{j_1\}$ with corresponding dimension $J_1 = 2$. The first row of $\boldsymbol{A}$ encodes the summation over the entries $w_{\overline{11}}, w_{\overline{12}}, w_{\overline{13}}$, which is the first row of $\boldsymbol{W}$. Likewise, the second row of $\boldsymbol{A}$ encodes the summation over the second row of $\boldsymbol{A}$.

3.3. Eigenvector structure.

Tensors whose vectorization is an eigenvector of a matrix $\boldsymbol{P}$ with eigenvalue λ are described by the constraint $\boldsymbol{A} = \lambda \boldsymbol{I} - \boldsymbol{P}$ and $\boldsymbol{b} = \boldsymbol{0}$. An important structure in this article is obtained when $\boldsymbol{P}$ is a permutation matrix. Indeed, $\boldsymbol{P}\,\boldsymbol{w} = \boldsymbol{w}$ then implies that the entries of $\boldsymbol{\mathcal{W}}$ remain invariant under the permutation $\boldsymbol{P}$. In what follows, we only consider tensors for which $J_1 = J_2 = \cdots = J_D = J$. The distinction between $\lambda = 1$ and $\lambda = -1$ is made explicit through the following two definitions.

Definition 3.7. *Let $\boldsymbol{P} \in \mathbb{R}^{J^D \times J^D}$ be a permutation matrix. A $\boldsymbol{P}$-invariant tensor $\boldsymbol{\mathcal{W}}$ is defined by*

$$(\boldsymbol{I}-\boldsymbol{P})\,\boldsymbol{w}=\boldsymbol{0}\Leftrightarrow\boldsymbol{P}\,\boldsymbol{w}=\boldsymbol{w}.$$

Likewise, a skew-$\boldsymbol{P}$-invariant tensor $\boldsymbol{\mathcal{W}}$ is defined by

$$(-\boldsymbol{I}-\boldsymbol{P})\,\boldsymbol{w}=\boldsymbol{0}\Leftrightarrow\boldsymbol{P}\,\boldsymbol{w}=-\boldsymbol{w}.$$

In this way, any particular permutation matrix $\boldsymbol{P}$ then defines a corresponding structured tensor. Next we discuss some prominent examples of $\boldsymbol{P}$-invariant tensor structures found in different applications [3, 13, 20, 24, 31, 42].

Definition 3.8 (Symmetric tensor). *Let $\boldsymbol{S}$ be the permutation matrix such that all entries of $\tilde{\boldsymbol{w}}:=\boldsymbol{S}\,vec(\boldsymbol{\mathcal{W}})$ satisfy $\tilde{w}_{j_1,\ldots,j_D}=w_{\pi(j_1,\ldots,j_D)}$, where $\pi(j_1,\ldots,j_D)$ is any permutation of the indices. An $\boldsymbol{S}$-invariant tensor $\boldsymbol{\mathcal{W}}$ defines a symmetric tensor.*

Definition 3.9 (Centrosymmetric tensor [7]). *A $\boldsymbol{J}$-invariant tensor $\boldsymbol{\mathcal{W}}$, where $\boldsymbol{J}$ is the column-reversed identity matrix, is called a centrosymmetric tensor.*

A centrosymmetric tensor $\boldsymbol{\mathcal{W}}$ satisfies

$$w_{j_1,\ldots,j_D}=w_{J_1-j_1+1,\ldots,J_D-j_D+1}.$$

Probably the most famous tensor that exhibits centrosymmetry is the matrix-matrix multiplication tensor [12].

Definition 3.10 (Hankel Tensor). *Let $\boldsymbol{H}\in\mathbb{R}^{J^D\times J^D}$ be the permutation matrix that cyclically permutes all D indices $j_1,\ldots,j_D$ with constant index sum $j_1+\cdots+j_D$. A $\boldsymbol{H}$-invariant tensor $\boldsymbol{\mathcal{W}}$ is called a Hankel tensor.*

The minimal index sum is $D=1+1+1+\cdots+1$ and maximal index sum is $JD=J+J+\cdots+J$. This implies that $\boldsymbol{H}$ consists of $JD-D+1$ permutation cycles and $\text{rank}(\boldsymbol{H})=JD-D+1$.

Definition 3.11 (Toeplitz Tensor). *Let $\boldsymbol{T}\in\mathbb{R}^{J^D\times J^D}$ be the permutation matrix that cyclically permutes all indices $j_d\mapsto j_d+1$, where $J_d+1\mapsto 1\ (d=1,\ldots,D)$. A $\boldsymbol{T}$-invariant tensor $\boldsymbol{\mathcal{W}}$ is called a Toeplitz tensor.*

A special case of a Toeplitz tensor is a circulant tensor.

Definition 3.12 (Circulant Tensor). *Let $\boldsymbol{T}\in\mathbb{R}^{J^D\times J^D}$ be the permutation matrix that cyclically permutes all indices $j_d\mapsto \mathrm{mod}\,(j_d+1,J_d)\neq 0$. If $\mathrm{mod}\,(j_d+1,J_d)=0$, then $j_d\mapsto J_d\ (d=1,\ldots,D)$. A $\boldsymbol{T}$-invariant tensor $\boldsymbol{\mathcal{W}}$ is called a circulant tensor.*

4. Explicit covariance matrix construction for permutation-invariant tensors.

Computing the covariance matrix $\boldsymbol{P}_0$ via Lemma 2.1 requires a basis for the nullspace of $\boldsymbol{A}$. For $\boldsymbol{P}$-invariant tensors it is possible to derive an explicit formula for $\boldsymbol{P}_0$ as a function of the permutation matrix $\boldsymbol{P}$, which enables efficient sampling of the prior. Before we can state the main result in Theorem 4.5, we first need to discuss some facts about permutation matrices. An important concept tied to permutation matrices is its order. Any permutation can be written as a product of R disjoint cycles. From Theorem 4.13 we will show that R is exactly the nullity of $\boldsymbol{A}$ and therefore the dimension of the subspace in which the prior samples live. Each cycle has a particular length, also called the order of the cycle. In this article K will denote the least common multiple of all orders of disjoint cycles of a given permutation.

Definition 4.1. *The order $K \in \mathbb{N}$ of a permutation matrix $\boldsymbol{P}$ is defined as the smallest natural number such that $\boldsymbol{P}^K = \boldsymbol{I}$.*

Skew-$\boldsymbol{P}$-invariant structures always have an even order K.

Lemma 4.2. *A skew-$\boldsymbol{P}$-invariant structure has an even order K.*

Proof. From the definition of skew-$\boldsymbol{P}$-invariance follows that $\boldsymbol{P}\boldsymbol{w} = -\boldsymbol{w}$. The definition of the order K tells us that $\boldsymbol{P}^K\boldsymbol{w} = \boldsymbol{w}$. Combining these two definitions results in the statement that $(-1)^K = 1$, which can only be true when K is even. ■

Theorem 4.5 will express the desired covariance matrix $\boldsymbol{P}_0$ as a function of powers of the permutation matrix $\boldsymbol{P}$. The following two lemmas relating powers of permutation matrices are easily proved.

Lemma 4.3. *Let $\boldsymbol{P}$ be a permutation matrix of order K, then for any $1 \leq k \leq K$:*

$$\boldsymbol{P}^k = \boldsymbol{P}^{K+k}. \tag{4.1}$$

Lemma 4.4. *Let $\boldsymbol{P}$ be a permutation matrix of order K, then for any $1 \leq k \leq K$:*

$$\boldsymbol{P}^{K-k} = \left(\boldsymbol{P}^k\right)^T. \tag{4.2}$$

Lemma 4.3 follows from $\boldsymbol{P}^K = \boldsymbol{I}$. Lemma 4.4 follows from the orthogonality of permutation matrices and from the fact that powers of permutation matrices are still permutation matrices. We now have all ingredients to describe the main result that provides an analytic solution for the covariance matrix $\boldsymbol{P}_0$ as an average over powers of the permutation matrix P.

Theorem 4.5. *Let $\boldsymbol{P}$ be a permutation matrix of order K. The Gaussian distribution of $\boldsymbol{P}$-invariant tensors $\mathcal{N}(\boldsymbol{w}_0, \boldsymbol{P}_0)$ is described by a mean vector $\boldsymbol{w}_0$ that is $\boldsymbol{P}$-invariant and covariance matrix*

$$\boldsymbol{P}_0 = \frac{\boldsymbol{P} + \boldsymbol{P}^2 + \cdots + \boldsymbol{P}^K}{K}. \tag{4.3}$$

The $\boldsymbol{P}$-invariance of the mean $\boldsymbol{w}_0$ follows directly from Lemma 2.1. The proof of Theorem 4.5 therefore requires showing that $\boldsymbol{P}_0$ in (4.3) is the desired covariance matrix. A matrix $\boldsymbol{P}_0$ is a covariance matrix if it satisfies the following three sufficient conditions:

1. has positive diagonal entries,
2. is symmetric,
3. is positive (semi-)definite.

Short proofs will now be given for each of these three covariance conditions.

Lemma 4.6. *The matrix $\boldsymbol{P}_0$ has positive diagonal entries.*

Proof. $\boldsymbol{P}_0$ is defined as a sum of permutation matrices, all diagonal entries of $\boldsymbol{P}_0$ are therefore either zero or positive. Since $\boldsymbol{P}^K = \boldsymbol{I}$, we have that the diagonal entries are guaranteed to be positive. ■

Lemma 4.7. *The matrix $\boldsymbol{P}_0$ is symmetric.*

Proof. The symmetry of $\boldsymbol{P}_0$ follows from

$$\begin{aligned}\boldsymbol{P}_0^T &= \frac{\boldsymbol{P}^T + (\boldsymbol{P}^2)^T + \cdots + (\boldsymbol{P}^{K-1})^T + (\boldsymbol{P}^K)^T}{K},\\ &= \frac{\boldsymbol{P}^{K-1} + \boldsymbol{P}^{K-2} + \cdots + \boldsymbol{P} + \boldsymbol{P}^K}{K},\\ &= \boldsymbol{P}_0,\end{aligned}$$

where the second line follows from Lemma 4.4. ■

The semipositive definiteness of $\boldsymbol{P}_0$ follows from its idempotency.

Lemma 4.8. *The matrix $\boldsymbol{P}_0$ is idempotent, that is $\boldsymbol{P}_0^2 = \boldsymbol{P}_0$.*

Proof. Writing out $(K\boldsymbol{P}_0)^2$ in terms of $\boldsymbol{P}$ and applying Lemma 4.3 results in

$$\begin{aligned}&(\boldsymbol{P} + \boldsymbol{P}^2 + \cdots + \boldsymbol{P}^K)^2,\\ &\quad = \boldsymbol{P}^2 + 2\,\boldsymbol{P}^3 + \cdots + (K-1)\,\boldsymbol{P}^K + K\,\boldsymbol{P}^{K+1} + (K-1)\,\boldsymbol{P}^{K+2} + \cdots + 2\,\boldsymbol{P}^{2K-1} + \boldsymbol{P}^{2K},\\ &\quad = K\,\boldsymbol{P} + \underbrace{\boldsymbol{P}^2 + (K-1)\,\boldsymbol{P}^{K+2}}_{K\,\boldsymbol{P}^2} + \cdots + \underbrace{2\,\boldsymbol{P}^{2K-1} + (K-2)\,\boldsymbol{P}^{K-1}}_{K\,\boldsymbol{P}^{K-1}} + \underbrace{(K-1)\,\boldsymbol{P}^K + \boldsymbol{P}^{2K}}_{K\boldsymbol{P}^K},\\ &\quad = K\,(\boldsymbol{P} + \boldsymbol{P}^2 + \boldsymbol{P}^3 + \cdots + \boldsymbol{P}^K),\\ &\quad = K^2\,\boldsymbol{P}_0,\end{aligned}$$

which proves that $\boldsymbol{P}_0$ is idempotent. ■

The first consequence of $\boldsymbol{P}_0$ being idempotent is that it is positive semidefinite.

Lemma 4.9. *The matrix $\boldsymbol{P}_0$ is positive semidefinite.*

Proof. The two eigenvalue equations

$$\boldsymbol{P}_0\,\boldsymbol{v} = \lambda\,\boldsymbol{v}, \quad (\boldsymbol{P}_0)^2\,\boldsymbol{v} = \lambda^2\,\boldsymbol{v}$$

are actually equal due to $\boldsymbol{P}_0$ being idempotent. It therefore follows that $\lambda^2 - \lambda = 0$, which implies that the eigenvalues are either 0 or 1. This proves the positive semidefiniteness of $\boldsymbol{P}_0$. ■

Having proved that $\boldsymbol{P}_0$ is a covariance matrix it remains to show that samples drawn from $\mathcal{N}(\boldsymbol{w}_0, \boldsymbol{P}_0)$ are $\boldsymbol{P}$-invariant. From its symmetry and idempotency it follows that $\boldsymbol{P}_0$ is its own matrix square root $\boldsymbol{P}_0 = \sqrt{\boldsymbol{P}_0} = \boldsymbol{P}_0^T = \sqrt{\boldsymbol{P}_0}^T$.

Lemma 4.10. *Every sample $\boldsymbol{w}$ drawn from $\mathcal{N}(\boldsymbol{w}_0, \boldsymbol{P}_0)$ is $\boldsymbol{P}$-invariant.*

Proof. A sample $\boldsymbol{w}$ from $\mathcal{N}(\boldsymbol{w}_0, \boldsymbol{P}_0)$ can be drawn by computing $\boldsymbol{w} = \boldsymbol{w}_0 + \sqrt{\boldsymbol{P}_0}\,\boldsymbol{x}$, where $\boldsymbol{x}$ is drawn from a standard normal distribution $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. The $\boldsymbol{P}$-invariance of $\boldsymbol{w}$ follows from

$$\begin{aligned}\boldsymbol{P}\,\boldsymbol{w} &= \boldsymbol{P}\,\boldsymbol{w}_0 + \boldsymbol{P}\,\sqrt{\boldsymbol{P}_0}\,\boldsymbol{x},\\ &= \boldsymbol{w}_0 + \boldsymbol{P}\,\sqrt{\boldsymbol{P}_0}\,\boldsymbol{x},\end{aligned}$$

$$
\begin{aligned}
&= \boldsymbol{w}_0 + \boldsymbol{P}\left(\frac{\boldsymbol{P} + \boldsymbol{P}^2 + \cdots + \boldsymbol{P}^{K-1} + \boldsymbol{P}^K}{K}\right)\boldsymbol{x},\\
&= \boldsymbol{w}_0 + \left(\frac{\boldsymbol{P}^2 + \boldsymbol{P}^3 + \cdots + \boldsymbol{P}^K + \boldsymbol{P}}{K}\right)\boldsymbol{x},\\
&= \boldsymbol{w}_0 + \sqrt{\boldsymbol{P}_0}\,\boldsymbol{x} = \boldsymbol{w}.
\end{aligned}
$$

The $\boldsymbol{P}$-invariance of $\boldsymbol{w}_0$ is used to go from line 1 to 2 and Lemma 4.3 is used to go from line 3 to line 4. ■

Lemmas 4.6 up to 4.10 constitute the proof of Theorem 4.5. Another consequence from the idempotency of $\boldsymbol{P}_0$ is that this matrix is its own pseudoinverse.

Lemma 4.11. *The pseudoinverse $\boldsymbol{P}_0^\dagger$ satisfies*

$$\boldsymbol{P}_0^\dagger = \boldsymbol{P}_0.$$

Proof. The pseudoinverse $\boldsymbol{P}_0^\dagger$ needs to satisfy the following four properties:

1. $\boldsymbol{P}_0\boldsymbol{P}_0^\dagger\boldsymbol{P}_0 = \boldsymbol{P}_0$,
2. $\boldsymbol{P}_0^\dagger\boldsymbol{P}_0\boldsymbol{P}_0^\dagger = \boldsymbol{P}_0^\dagger$,
3. $(\boldsymbol{P}_0\boldsymbol{P}_0^\dagger)^T = \boldsymbol{P}_0\boldsymbol{P}_0^\dagger$,
4. $(\boldsymbol{P}_0^\dagger\boldsymbol{P}_0)^T = \boldsymbol{P}_0^\dagger\boldsymbol{P}_0$.

All these properties are satisfied when assuming $\boldsymbol{P}_0^\dagger = \boldsymbol{P}_0$ and they follow from the idempotency of $\boldsymbol{P}_0$. For example, Properties 1 and 2 follow from

$$\boldsymbol{P}_0\boldsymbol{P}_0^\dagger\boldsymbol{P}_0 = \boldsymbol{P}_0^\dagger\boldsymbol{P}_0\boldsymbol{P}_0^\dagger = (\boldsymbol{P}_0)^3 = \boldsymbol{P}_0 = \boldsymbol{P}_0^\dagger.$$

Properties 3 and 4 follow from the symmetry of $\boldsymbol{P}_0$. ■

The fact that $\boldsymbol{P}_0 = \sqrt{\boldsymbol{P}_0} = \boldsymbol{P}_0^\dagger = \sqrt{\boldsymbol{P}_0^\dagger}$ is convenient for several reasons. First, no explicit $\boldsymbol{P}_0^{-1}$ computation is required in equations (1.3) and (1.4). Second, sampling $\mathcal{N}(\boldsymbol{w}_0, \boldsymbol{P}_0)$ can be done without a matrix square-root computation and without any matrix-vector multiplications. Using Theorem 4.5 the product $\sqrt{\boldsymbol{P}_0}\,\boldsymbol{x} = \boldsymbol{P}_0\,\boldsymbol{x}$ can be implemented as a weighted sum of permuted versions of $\boldsymbol{x}$

$$\frac{\boldsymbol{P}\,\boldsymbol{x} + \boldsymbol{P}^2\,\boldsymbol{x} + \cdots + \boldsymbol{P}^K\,\boldsymbol{x}}{K}.$$

All information of the permutation $\boldsymbol{P}$ is contained in a vector $\boldsymbol{p}$ of J^D entries that specifies how each entry gets mapped to the next. Each term $\boldsymbol{P}^k\,\boldsymbol{x}$ of the weighted sum is then computed by successive permutations of $\boldsymbol{x}$ according to $\boldsymbol{p}$ with computational complexity $O(J^D)$. The pseudocode for sampling the distribution is given in Algorithm 4.1.

A similar result as in Theorem 4.5 can be proven for $\boldsymbol{P}$-skew-invariant tensors.

Theorem 4.12. *For a permutation of even order K, the Gaussian distribution of $\boldsymbol{P}$-skew-invariant tensors $\mathcal{N}(\boldsymbol{w}_0, \boldsymbol{P}_0)$ is described by a mean vector $\boldsymbol{w}_0$ that is $\boldsymbol{P}$-skew-invariant and covariance matrix*

$$\boldsymbol{P}_0 := \frac{-\boldsymbol{P} + \boldsymbol{P}^2 - \cdots + \boldsymbol{P}^K}{K} = \frac{\sum_{k=1}^K (-1)^k\,\boldsymbol{P}^k}{K}. \tag{4.4}$$

Algorithm 4.1. Generate $\boldsymbol{P}$-invariant sample from $\mathcal{N}(\boldsymbol{w}_0, \boldsymbol{P}_0)$

```
Require: $\boldsymbol{w}_0$, index permutation vector $\boldsymbol{p}$, $K$
  $\boldsymbol{x} \leftarrow \mathrm{randn}(J^D)$          % sample standard normal $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$
  $\boldsymbol{w} \leftarrow K\,\boldsymbol{w}_0$
  for $k = 1 : K$ do
    $\boldsymbol{w} \leftarrow \boldsymbol{w} + \boldsymbol{x}$
    $\boldsymbol{x} \leftarrow \boldsymbol{x}[\boldsymbol{p}]$          % permute entries of $\boldsymbol{x}$ according to $\boldsymbol{p}$
  end for
  $\boldsymbol{w} \leftarrow \frac{\boldsymbol{w}}{K}$
  return $\boldsymbol{w}$
```

Proof. The proof is very similar to that of Theorem 4.5. The diagonal entries being nonnegative can be derived from the following argument. The permutation matrix $\boldsymbol{P}$ itself consists of cyclic permutations, with either even or odd order. If a cyclic permutation has an even order k, then $\boldsymbol{P}^k$ will have ones on the diagonal for entries of the cycle. This cycle will occur K/k times in (4.4), always with a positive sign. If a cyclic permutation has odd order k, then the diagonal entries of $\boldsymbol{P}^k$ will come in equal amounts of $K/(2k)$ negative and $K/(2k)$ positive contributions, which results in a zero contribution to the diagonal. The total effect of all cyclic permutations then add up to either zero or positive diagonal entries. Symmetry is proven by using Corollary 4.4 and the fact that K is even: an even order k gets mapped to another even order $K-k$ and an odd order k gets mapped to and odd order $K-k$. Hence,

$$\boldsymbol{P}_0^T = \frac{\sum_{k=1}^K (-1)^k\,(\boldsymbol{P}^k)^T}{K} = \frac{\sum_{k=1}^K (-1)^k\,\boldsymbol{P}^{K-k}}{K} = \boldsymbol{P}_0.$$

The idempotency of $\boldsymbol{P}_0$ follows a similar proof as for the case of $\boldsymbol{P}$-invariance. Writing out $(K\boldsymbol{P}_0)^2$ in terms of $\boldsymbol{P}$ and applying Corollary 4.3 results in

$$\begin{aligned}
&(-\boldsymbol{P} + \boldsymbol{P}^2 - \cdots + \boldsymbol{P}^K)^2 \\
&\quad = \boldsymbol{P}^2 - 2\,\boldsymbol{P}^3 + \cdots + (K-1)\,\boldsymbol{P}^K - K\,\boldsymbol{P}^{K+1} + (K-1)\,\boldsymbol{P}^{K+2} - \cdots - 2\,\boldsymbol{P}^{2K-1} + \boldsymbol{P}^{2K} \\
&\quad = -K\,\boldsymbol{P} + \underbrace{\boldsymbol{P}^2 + (K-1)\,\boldsymbol{P}^{K+2}}_{K\,\boldsymbol{P}^2} - \cdots \underbrace{-2\,\boldsymbol{P}^{2K-1} - (K-2)\,\boldsymbol{P}^{K-1}}_{-K\,\boldsymbol{P}^{K-1}} + \underbrace{(K-1)\,\boldsymbol{P}^K + \boldsymbol{P}^{2K}}_{K\boldsymbol{P}^K} \\
&\quad = K\,(-\boldsymbol{P} + \boldsymbol{P}^2 - \boldsymbol{P}^3 + \cdots + \boldsymbol{P}^K) \\
&\quad = K^2\,\boldsymbol{P}_0
\end{aligned}$$

which proves that $\boldsymbol{P}_0$ is idempotent. ■

Theorems 4.5 and 4.12 are practical when the order K of the permutation matrix $\boldsymbol{P}$ stays small compared to J and D. For Hankel structures this is unfortunately not the case. Consider for example a 20×20 Hankel matrix. Its corresponding permutation matrix has permutation cycles ranging from length 1 up to 20, and K is therefore the least common multiple of $1, 2, \ldots, 20 = 232{,}792{,}560$. Fortunately, it is possible to explicitly construct a sparse matrix of orthogonal columns $\boldsymbol{V}$ such that $\sqrt{\boldsymbol{P}_0} = \boldsymbol{V}$.

Every permutation $\boldsymbol{P}$ can be decomposed in terms of R cyclic permutations. These cyclic permutations partition the set of all tensor entries into R disjoint sets and allow for an alternative construction of $\sqrt{\boldsymbol{P}_0}$, where the resulting matrix is sparse and consists of orthogonal columns.

Theorem 4.13. *Let $\boldsymbol{P}$ be a permutation matrix that consists of R permutation cycles and let C_r denote the rth cycle, where the number of tensor entries in C_r is denoted $|C_r|$. Then the range of the matrix $\boldsymbol{V} \in \mathbb{R}^{J^D \times R}$ such that*

$$v_{\overline{j_1, j_2, \ldots, j_D}, r} = \begin{cases} \dfrac{1}{\sqrt{|C_r|}} & \text{if } w_{j_1, j_2, \ldots, j_D} \in C_r, \\ 0 & \text{otherwise}, \end{cases} \tag{4.5}$$

spans the eigenspace of $\boldsymbol{P}$ corresponding to an eigenvalue $\lambda = 1$. In other words, $\boldsymbol{V} = \sqrt{\boldsymbol{P}_0}$. Also, $\boldsymbol{V}^T \boldsymbol{V} = \boldsymbol{I}_R$.

Proof. The equality $\boldsymbol{P}\boldsymbol{V} = \boldsymbol{V}$ follows from each column of $\boldsymbol{V}$ containing nonzero values at tensor entries of a particular permutation cycle of $\boldsymbol{P}$. The orthogonality follows directly from the permutation cycles being disjoint and each column of $\boldsymbol{V}$ being unit-norm due to the scaling with $\sqrt{|C_r|}$. ■

Theorem 4.13 tells us that the dimension of the subspace induced by the prior is equal to the number of permutation cycles R. A basis for the skew-$\boldsymbol{P}$-invariant eigenspace can be built in a similar way by retaining the cycles of even order and alternating the sign of the entries $v_{\overline{j_1, j_2, \ldots, j_D}, r}$ in each column.

Example 4.14. Consider a 20×20 Hankel matrix. Using Theorem 4.5, one would need to construct the 400×400 Hankel permutation matrix $\boldsymbol{H}$ and construct $\boldsymbol{P}_0$ by adding $232,792,560$ terms together. Using Theorem 4.13, the sparse 400×39 matrix $\boldsymbol{V}$ can be constructed directly containing 400 nonzero entries.

5. Solving the inverse problem.

In this section two different aspects when solving the inverse problem are discussed. First, we briefly discuss how to sample the posterior and how a change of variables, originally proposed in [11], can exploit fast implementations of the matrix vector product $\boldsymbol{P}_0 \boldsymbol{w}$. The second aspect relates to kernel methods, where $(\boldsymbol{A}, \boldsymbol{b})$-constrained tensor priors are used to define new structured tensor kernel functions.

5.1. Change of variables.

Squaring the condition number when solving the normal equation of (1.3) can be avoided by solving its square-root version

$$\begin{pmatrix} \sqrt{\boldsymbol{\Sigma}^{-1}}\boldsymbol{\Phi} \\ \sqrt{\boldsymbol{P}_0^{-1}} \end{pmatrix} \boldsymbol{w}_+ = \begin{pmatrix} \sqrt{\boldsymbol{\Sigma}^{-1}}\boldsymbol{y} \\ \sqrt{\boldsymbol{P}_0^{-1}}\,\boldsymbol{w}_0 \end{pmatrix}$$

instead. In order to sample the posterior one requires the posterior square-root covariance $\sqrt{\boldsymbol{P}_+}$ and mean vector $\boldsymbol{w}_+$. Both can be computed from the SVD of the square-root precision matrix

$$\begin{pmatrix} \sqrt{\boldsymbol{\Sigma}^{-1}}\boldsymbol{\Phi} \\ \sqrt{\boldsymbol{P}_0^{-1}} \end{pmatrix} = \boldsymbol{U}\,\boldsymbol{S}\,\boldsymbol{V}^T \ \text{ as } \ \boldsymbol{w}_+ = \boldsymbol{V}\boldsymbol{S}^{-1}\boldsymbol{U}^T \begin{pmatrix} \sqrt{\boldsymbol{\Sigma}^{-1}}\boldsymbol{y} \\ \sqrt{\boldsymbol{P}_0^{-1}}\,\boldsymbol{w}_0 \end{pmatrix}, \ \sqrt{\boldsymbol{P}_+} = \boldsymbol{V}\boldsymbol{S}^{-1}.$$

When the SVD computation is too expensive, iterative solvers such as hybrid projection methods [10, 11] or randomized solvers [41] can be used. In [11] a change of variables was introduced to avoid explicit construction of the square-root prior precision matrix $\sqrt{\boldsymbol{P}_0^{-1}}$. By defining $\boldsymbol{x} := \boldsymbol{P}_0^{-1}(\boldsymbol{w}_+ - \boldsymbol{w}_0)$ and $\boldsymbol{z} := \boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{w}_0$, the square-root linear system is transformed into

$$\begin{pmatrix} \sqrt{\boldsymbol{\Sigma}^{-1}}\boldsymbol{\Phi}\boldsymbol{P}_0 \\ \boldsymbol{I} \end{pmatrix} \boldsymbol{x} = \begin{pmatrix} \sqrt{\boldsymbol{\Sigma}^{-1}}\boldsymbol{z} \\ 0 \end{pmatrix}.$$

The desired posterior mean $\boldsymbol{w}_+$ can then be recovered from $\boldsymbol{w}_+ = \boldsymbol{P}_0\boldsymbol{x} + \boldsymbol{w}_0$. This formulation is especially beneficial when the matrix vector product $\boldsymbol{P}_0\boldsymbol{x}$ can be implemented in a computationally efficient manner, for example using Algorithm 4.1.

5.2. Structured tensor kernel functions.

When the tensor $\boldsymbol{\mathcal{W}}$ is much larger than the data size N, then the $O(J^{3D})$ computational complexity of computing (1.3) is replaced with at least $O(N^2)$ by solving the corresponding dual problem

$$(\boldsymbol{\Phi}\,\boldsymbol{P}_0\,\boldsymbol{\Phi}^T + \boldsymbol{\Sigma})\,\boldsymbol{v} = \boldsymbol{y}. \tag{5.1}$$

An additional benefit is that no matrix inverse of $\boldsymbol{P}_0$ is required so that Theorems 2.1, 4.5, and 4.13 can be applied directly. The matrix $\boldsymbol{\Phi}\,\boldsymbol{P}_0\,\boldsymbol{\Phi}^T$ is called the kernel matrix $\boldsymbol{K}$ and each entry $k_{i,j}$ is defined as the evaluation of a kernel function

$$k_{i,j} = k(\boldsymbol{x}_i, \boldsymbol{x}_j) := \boldsymbol{\varphi}(\boldsymbol{x}_i)^T\,\boldsymbol{P}_0\,\boldsymbol{\varphi}(\boldsymbol{x}_j).$$

Choosing $\boldsymbol{P}_0$ as a covariance matrix of an $(\boldsymbol{A},\boldsymbol{b})$-constrained tensor allows us to define new kernel functions. The kernel trick in machine learning refers to the fact where the kernel function can be evaluated without every explicitly computing the possibly large feature vectors $\boldsymbol{\varphi}(\cdot)$. In the case of $\boldsymbol{P}$-invariant tensors one can exploit the particular structure of $\boldsymbol{P}_0$ as described in Theorem 4.5 or use Algorithm 4.1 to achieve this goal.

Example 5.1 (Centrosymmetric polynomial kernel). Let $\sqrt{c} \in \mathbb{R}$ and $d \in \mathbb{N}$. The polynomial kernel function is defined as

$$\begin{aligned} k(\boldsymbol{x}_i, \boldsymbol{x}_j) &= \boldsymbol{\varphi}(\boldsymbol{x}_i)^T\,\boldsymbol{I}\,\boldsymbol{\varphi}(\boldsymbol{x}_j), \\ &= \underbrace{\left(\sqrt{c} \quad \boldsymbol{x}_i^T\right) \otimes \cdots \otimes \left(\sqrt{c} \quad \boldsymbol{x}_i^T\right)}_{d \text{ times}}\ \boldsymbol{I}\ \underbrace{\left(\sqrt{c} \quad \boldsymbol{x}_j^T\right)^T \otimes \cdots \otimes \left(\sqrt{c} \quad \boldsymbol{x}_j^T\right)^T}_{d \text{ times}} \\ &= (c + \boldsymbol{x}_i^T\,\boldsymbol{x}_j)^d. \end{aligned}$$

The expression $(c + \boldsymbol{x}_i^T\,\boldsymbol{x}_j)^d$ is obtained from writing the identity matrix $\boldsymbol{I}$ as a Kronecker product of smaller identity matrices and applying the mixed product property. The polynomial kernel function can therefore be interpreted as using a unit covariance matrix $\boldsymbol{P}_0$. We can now define the centrosymmetric polynomial kernel function k_2 by using the polynomial feature vectors $\boldsymbol{\varphi}(\cdot)$ and replacing $\boldsymbol{I}$ with the covariance matrix $(\boldsymbol{I}+\boldsymbol{J})/2$ of centrosymmetric tensors. From Theorem 4.5, it then follows that

$$
\begin{aligned}
k_2(\boldsymbol{x}_i,\boldsymbol{x}_j) &= \frac{1}{2}\,\boldsymbol{\varphi}(\boldsymbol{x}_i)^T\,(\boldsymbol{I}+\boldsymbol{J})\,\boldsymbol{\varphi}(\boldsymbol{x}_j),\\
&= \frac{1}{2}\,\underbrace{\left(\sqrt{c}\quad \boldsymbol{x}_i^T\right)\otimes\cdots\otimes\left(\sqrt{c}\quad \boldsymbol{x}_i^T\right)}_{d \text{ times}}(\boldsymbol{I}+\boldsymbol{J})\,\underbrace{\left(\sqrt{c}\quad \boldsymbol{x}_j^T\right)^T\otimes\cdots\otimes\left(\sqrt{c}\quad \boldsymbol{x}_j^T\right)^T}_{d \text{ times}},\\
&= \frac{1}{2}(c+\boldsymbol{x}_i^T\boldsymbol{x}_j)^d+\frac{1}{2}\left(\left(\sqrt{c}\quad \boldsymbol{x}_i^T\right)\boldsymbol{J}_d\left(\sqrt{c}\quad \boldsymbol{x}_j^T\right)^T\right)^d.
\end{aligned}
$$

Also here the explicit construction of $\boldsymbol{\varphi}(\cdot)$ is avoided by writing the matrix $\boldsymbol{J}\in\mathbb{R}^{J^D\times J^D}$ as a Kronecker product of the smaller permutation matrix $\boldsymbol{J}_d\in\mathbb{R}^{J\times J}$ with itself d times and using the mixed-product property.

6. Applications.

In this section we demonstrate the use of Theorems 2.1, 4.5, and 4.13 in three different applications. Practical implementations on how to sample various $(\boldsymbol{A},\boldsymbol{b})$-constrained tensor priors are explained in Application 6.1. We consider lower triangular tensors, tensors for which the sum over the last index adds up to 1, symmetric tensors and Hankel tensors. Application 6.2 considers the problem of completing a Hankel matrix from noisy partial measurements by solving it as a Bayesian inverse problem. The estimate of the completed Hankel matrix when using a Hankel prior is compared to the estimate where no prior is used. In Application 6.3 learning a classifier for handwritten digits is solved as a Bayesian inverse problem. The classifier obtained with the commonly used Tikhonov prior is compared to several $(\boldsymbol{A},\boldsymbol{b})$-constrained tensor priors.

All applications have been implemented as reactive Pluto [36] notebooks in Julia [8] and are publicly available at https://github.com/kbatseli/AbTensors. The notebook files can be freely downloaded and run on your local machine in Julia. An alternative way to use these notebooks that does not require the installation of Julia is to run them in the cloud via Binder [30]. This can be done by clicking on each of the links on the main Github page.

As discussed in section 2.1 we parameterized the prior covariance matrix $\boldsymbol{P}_0$ with a single hyperparameter σ_P in both Applications 6.2 and 6.3.

6.1. Sampling structured tensor priors.

In this first application we demonstrate how Theorems 2.1, 4.5 and 4.13 are used to sample the priors of different $(\boldsymbol{A},\boldsymbol{b})$-constrained tensors.

Example 6.1 (**Lower triangular tensors**). A first example of an $(\boldsymbol{A},\boldsymbol{b})$-constrained tensor considered here are lower triangular tensors. From Definition 3.2 we know that triangular tensors are described by

$$
\boldsymbol{A}=\begin{pmatrix}\boldsymbol{A}_1\\ \boldsymbol{A}_2\\ \vdots\\ \boldsymbol{A}_{D-1}\end{pmatrix}=\begin{pmatrix}\boldsymbol{L}\otimes\boldsymbol{I}_J\otimes\cdots\otimes\boldsymbol{I}_J\\ \boldsymbol{I}_J\otimes\boldsymbol{L}\otimes\cdots\otimes\boldsymbol{I}_J\\ \vdots\\ \boldsymbol{I}_J\otimes\boldsymbol{I}_J\otimes\cdots\otimes\boldsymbol{L}\end{pmatrix}\in\mathbb{R}^{\frac{(D-1)(J-1)J^{D-1}}{2}\times J^D}
$$

and zero vector $\boldsymbol{b}$. The square root of the covariance matrix is built up by recursive computation of the nullspace of $\boldsymbol{A}$, considering only one block-row of $\boldsymbol{A}$ at a time. The whole $\boldsymbol{A}$ matrix is never explicitly made. In the notebook it is possible to sample lower triangular tensors with orders ranging from 2 up to 5 and dimensions 2 up to 6 by moving the corresponding sliders.

An example of a 6×6 lower triangular matrix sampled in this way is

$$\begin{pmatrix} 0.057705 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ -0.37369 & -0.71027 & 0.0 & 0.0 & 0.0 & 0.0 \\ -1.08755 & 0.600503 & -0.209701 & 0.0 & 0.0 & 0.0 \\ -0.334201 & 1.33079 & 0.926074 & 0.869311 & 0.0 & 0.0 \\ 0.9346 & -0.164181 & -0.230121 & -0.110009 & 0.294956 & 0.0 \\ -1.85009 & 0.771951 & -0.284285 & 0.467705 & -0.700058 & 0.205478 \end{pmatrix}.$$

Example 6.2 (**Tensors with known sum of entries**). In this example we sample tensors $\boldsymbol{\mathcal{W}}$ for which the sum over the last index always adds up to a value of 1:

$$\forall j_1, j_2, \ldots, j_{D-1} : \sum_{j_D} w_{j_1, j_2, \ldots, j_D} = b_{j_1, j_2, \ldots, j_{D-1}} = 1.$$

From Definition 3.5 we know that in this case $\boldsymbol{A} = \mathbf{1}_J^T \otimes \boldsymbol{I}_J \otimes \cdots \otimes \boldsymbol{I}_J$. It is straightforward to verify that a basis for the right nullspace of $\boldsymbol{A}$ is

$$\begin{pmatrix} 1 & 1 & \cdots & 1 \\ -1 & 0 & \cdots & 0 \\ 0 & -1 & \cdots & 0 \\ 0 & 0 & \cdots & -1 \end{pmatrix} \otimes I_J \otimes \cdots \otimes I_J.$$

Sampling the prior can now be done without every constructing a basis for the nullspace explicitly since

$$\begin{aligned} \sqrt{\boldsymbol{P}_0}\, \boldsymbol{x} &= \left(\begin{pmatrix} 1 & 1 & \cdots & 1 \\ -1 & 0 & \cdots & 0 \\ 0 & -1 & \cdots & 0 \\ 0 & 0 & \cdots & -1 \end{pmatrix} \otimes \boldsymbol{I}_J \otimes \cdots \otimes \boldsymbol{I}_J \right) \boldsymbol{x} \\ &= \begin{pmatrix} \boldsymbol{I}_{J^{D-1}} & \boldsymbol{I}_{J^{D-1}} & \cdots & \boldsymbol{I}_{J^{D-1}} \\ -\boldsymbol{I}_{J^{D-1}} & 0 & \cdots & 0 \\ 0 & -\boldsymbol{I}_{J^{D-1}} & \cdots & 0 \\ 0 & 0 & \cdots & -\boldsymbol{I}_{J^{D-1}} \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{x}_{J-1} \end{pmatrix} = \begin{pmatrix} \boldsymbol{x}_1 + \boldsymbol{x}_2 + \cdots + \boldsymbol{x}_{J-1} \\ -\boldsymbol{x}_1 \\ -\boldsymbol{x}_2 \\ \vdots \\ -\boldsymbol{x}_{J-1} \end{pmatrix}. \end{aligned}$$

It is therefore sufficient to sample $\boldsymbol{x} \in \mathbb{R}^{(J-1)\, J^{D-1}}$ from a standard normal distribution and do the operations on the $J-1$ partitions of $\boldsymbol{x}$ as described above to generate the desired sample. An example of a generated 5×5 matrix sample is

$$\begin{pmatrix} -2.55782 & 1.49607 & 0.261848 & 0.920155 & 0.879752 \\ -0.521966 & 1.31607 & -0.287877 & 0.760659 & -0.266883 \\ -0.445579 & -0.330202 & -0.674927 & 0.690687 & 1.76002 \\ -2.48504 & -0.441223 & 2.02517 & 0.428044 & 1.47305 \\ 0.242143 & 2.06895 & -0.220772 & -0.59756 & -0.492757 \end{pmatrix}.$$

Summing each row of this matrix results in a vector of ones. In the notebook one can change the order of the sampled tensor from 2 up to 5 and dimension from 5 up to 10 by using the corresponding sliders.

Example 6.3 (**Symmetric tensors**). Symmetric tensors $\boldsymbol{\mathcal{W}}$ are tensors for which entries are invariant under any index permutation. The permutation matrix $\boldsymbol{S}$ in the symmetric case consists of cyclic permutations where each cycle contains the entry $w_{j_1,\dots,j_D}$ and all entries with corresponding index permutations $w_{\pi(j_1,\dots,j_D)}$. For example, in the case $D=2$ and $J=2$ the permutation matrix $\boldsymbol{S}$ consists of 3 cyclic permutations $w_{1,1}\mapsto w_{1,1}$, $w_{2,1}\mapsto w_{1,2}$, $w_{1,2}\mapsto w_{2,1}$, $w_{2,2}\mapsto w_{2,2}$. The order K of $\boldsymbol{S}$ in this case is 2 since $\boldsymbol{S}^2=\boldsymbol{I}$. According to Theorem 4.5, the square root of the covariance matrix is $\sqrt{\boldsymbol{P}_0}=(\boldsymbol{S}+\boldsymbol{S}^2)/2$. When $D=3$, the order K of the corresponding permutation matrix is 6 and hence $\sqrt{\boldsymbol{P}_0}=(\boldsymbol{S}+\boldsymbol{S}^2+\boldsymbol{S}^3+\boldsymbol{S}^4+\boldsymbol{S}^5+\boldsymbol{S}^6)/6$. Sampling from these priors is done via Algorithm 4.1 where a standard normal sample $\boldsymbol{x}\in\mathbb{R}^{J^D}$ is generated and permuted K times. An example of a sampled symmetric 5×5 matrix is

$$\begin{pmatrix} 0.530979 & -0.0991279 & -0.421909 & -1.76112 & -0.380734 \\ -0.0991279 & -0.388004 & -0.0187159 & 0.165119 & -0.446671 \\ -0.421909 & -0.0187159 & -0.879571 & -0.483638 & 0.624838 \\ -1.76112 & 0.165119 & -0.483638 & -1.35093 & 0.0902599 \\ -0.380734 & -0.446671 & 0.624838 & 0.0902599 & -2.03848 \end{pmatrix}.$$

The notebook allows you to sample symmetric tensors of orders 2 and 3 and dimensions 3 up to 10.

Example 6.4 (**Hankel tensors**). Hankel tensors $\boldsymbol{\mathcal{W}}$ are tensors for which entries with a constant index sum $j_1+\cdots+j_D$ have the same numerical value. The order K of the corresponding permutation matrix $\boldsymbol{P}$ grows very quickly. For example, when $D=2$ and $J=20$ the order K is the least common multiple of $1,2,\dots,20=232,792,560$. Theorem 4.13, however, allows us to construct a matrix $\sqrt{\boldsymbol{P}_0}\in\mathbb{R}^{J^D\times R}$, where R is the number of permutation cycles. For Hankel tensors we have that $R=D(J-1)+1$. An example of a sampled 5×5 Hankel matrix is

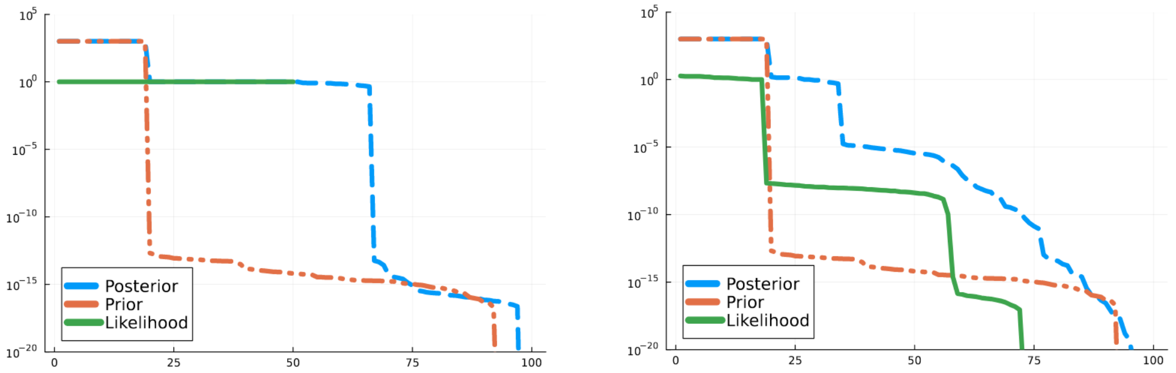
$$\begin{pmatrix} -0.0125803 & -1.1419 & -0.0329509 & -0.333555 & -1.32625 \\ -1.1419 & -0.0329509 & -0.333555 & -1.32625 & 0.20827 \\ -0.0329509 & -0.333555 & -1.32625 & 0.20827 & 0.759984 \\ -0.333555 & -1.32625 & 0.20827 & 0.759984 & -0.340118 \\ -1.32625 & 0.20827 & 0.759984 & -0.340118 & 1.22822 \end{pmatrix}$$

The notebook allows you to sample Hankel tensors of order 2 up to 4 and dimensions 3 up to 10.

6.2. Completion of a Hankel matrix from noisy measurements. Hankel matrices are very common in signal processing and control theory. In this application a Bayesian approach will be used to complete a Hankel matrix based on noisy incomplete measurements. For this we use the following forward model $\boldsymbol{y}=\boldsymbol{\Phi}\,\boldsymbol{w}+\boldsymbol{\epsilon}$, where $\boldsymbol{w}\in\mathbb{R}^{10^2}$ is the vectorization of the true underlying 10×10 Hankel matrix. The $I\times 10^2$ matrix $\boldsymbol{\Phi}$ selects I random entries of $\boldsymbol{w}$ with equal probability. Each row of $\boldsymbol{\Phi}$ contains a single nonzero unit-valued entry at a random location. The number of measurements I can be changed through a slider in the notebook.

The vector $\boldsymbol{\epsilon}$ is a vector of zero-mean Gaussian noise. Given $\boldsymbol{y}$ and $\boldsymbol{\Phi}$, a Bayesian estimate of the underlying Hankel matrix $\boldsymbol{W}$ can be obtained from (1.3) as the posterior mean $\boldsymbol{w}_+$. Another commonly used estimate is the maximum likelihood estimate, which is the $\boldsymbol{w}$ that maximizes the likelihood $p(\boldsymbol{y}|\boldsymbol{w})$. We compare two posterior estimates with the maximum likelihood estimate under two different assumptions on the noise covariance. We fix the sampling rate at 50% and choose $\sigma_\epsilon^2 = 1$. The prior covariance matrix is set to $\sigma_P^2 \boldsymbol{P}_0 = 10^{-6}\boldsymbol{P}_0$, where $\boldsymbol{P}_0$ is covariance matrix of the Hankel prior obtained via Theorem 4.13.

Example 6.5 (**White noise**). First we consider white noise, which implies that $\boldsymbol{\Sigma} = \sigma_\epsilon^2 \boldsymbol{I}$. The singular values of the prior precision $\sqrt{\boldsymbol{P}_0^{-1}}/\sigma_P$, posterior precision $\left(\boldsymbol{\Phi}^T/\sigma_\epsilon \; \sqrt{\boldsymbol{P}_0^{-1}}^T/\sigma_P{}^T \right)$, and likelihood precision $\boldsymbol{\Phi}/\sigma_\epsilon$ are shown in Figure 1a. They provide us with insight on how the prior, posterior and likelihood relate to each other. The likelihood $p(\boldsymbol{y}|\boldsymbol{w})$ only has 50 measurements and gives all of them equal weight. The prior $p(\boldsymbol{w})$ on the other hand only considers 19 nonzero values as a 10 × 10 Hankel matrix has 19 distinct entries. Given the relative high noise variance compared to the prior, the posterior $p(\boldsymbol{w}|\boldsymbol{y})$ "follows" the prior for the first 19 singular values. A prior mean is obtained by averaging over the nonzero antidiagonals of the measurements and using those averages to construct a Hankel matrix. We now compute three different estimates and compare them to the ground truth. The first estimate is obtained from (1.3) with a backslash solve. A second estimate is computed by truncating the SVD of $\left(\Phi^T/\sigma_\epsilon \; P_0^{-T}/\sigma_P \right)^T$ to rank 19 in equation (1.3). The third estimate is the maximum likelihood estimate. For each of these estimates we show the relative error in Table 1. Adding the Hankel prior shows a clear improvement on the completed Hankel matrix. The relative error is 4 times smaller from the inclusion of the prior. Since the first 19 singular values of the posterior are equal to the singular values of the prior, one could expect the estimated posterior mean $\boldsymbol{w}_+$ obtained from truncating the SVD to the first 19 singular



(a) White-noise case. Given the relative high noise variance the posterior follows the prior for the first 19 singular values.

(b) Hankel-noise case. Also in this case we have that the posterior follows the prior for the first 19 singular values.

Figure 1. *Singular values of the square-root precision matrices of the prior, likelihood and posterior distribution. Only* 50% *of the Hankel matrix* $\boldsymbol{W}$ *was measured. The noise variance is* 1 *and the prior variance is* 10^{-6}.

Table 1

Relative errors for three different Hankel matrix completion estimates $\hat{\boldsymbol{w}}$. Smallest relative error is indicated in bold.

	backslash	truncated SVD	max-likelihood
$\frac{\lvert\lvert\boldsymbol{w}-\hat{\boldsymbol{w}}\rvert\rvert_2}{\lvert\lvert\boldsymbol{w}\rvert\rvert_2}$ (white noise)	0.160	**0.137**	0.614
$\frac{\lvert\lvert\boldsymbol{w}-\hat{\boldsymbol{w}}\rvert\rvert_2}{\lvert\lvert\boldsymbol{w}\rvert\rvert_2}$ (Hankel noise)	0.235	**0.137**	0.604
$\frac{\lvert\lvert\boldsymbol{H}\hat{\boldsymbol{w}}-\hat{\boldsymbol{w}}\rvert\rvert_2}{\lvert\lvert\hat{\boldsymbol{w}}\rvert\rvert_2}$	0.12	6.3e-7	0.80

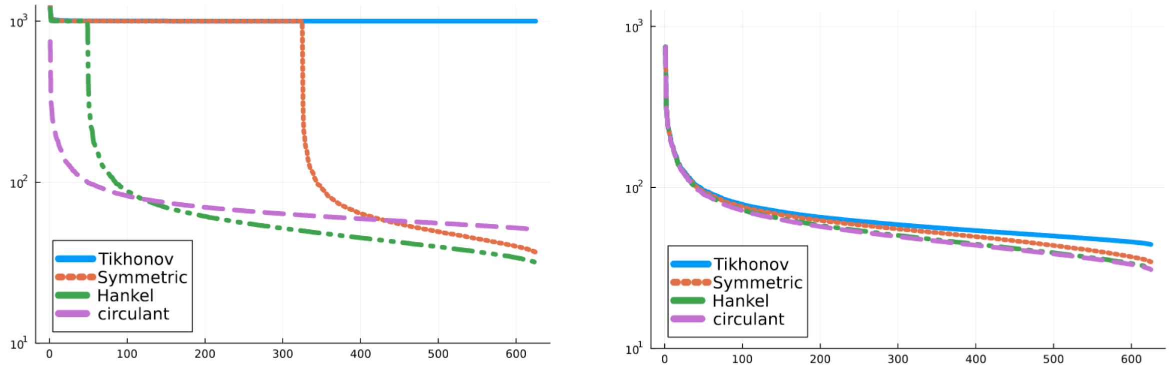
values to be Hankel. In order to confirm this, we also compute the relative Hankel error $||\boldsymbol{H}\,\boldsymbol{w}-\boldsymbol{w}||_2/||\boldsymbol{w}||_2$ for the three estimates in Table 1, where $\boldsymbol{H}$ is the Hankel permutation matrix. Restricting the posterior mean to lie in a subspace spanned by the first 19 right singular vectors indeed enforces a Hankel structure.

Example 6.6 (**Hankel distributed noise**). To investigate the effect of the noise covariance on the estimates we now consider noise $\boldsymbol{e}$ that also has a Hankel structure. In other words, the covariance matrix for $p(\boldsymbol{e})$ is $\sigma_\epsilon^2\,\boldsymbol{P}_0$, whereas the prior covariance is $\sigma_P^2\,\boldsymbol{P}_0$. With the noise being Hankel, this means that the perturbation $\boldsymbol{\epsilon}$ of $\boldsymbol{w}$ will have a Hankel structure as well. This can be modeled via the forward model $\boldsymbol{y}=\Phi(\boldsymbol{w}+\boldsymbol{\epsilon})$, where now $p(\Phi\boldsymbol{\epsilon})=\mathcal{N}(0,\sigma_\epsilon^2\,\Phi\,P_0\,\Phi^T)$. Figure 1b shows the singular values of the square-root precision matrices. The number of nonzero singular values of the likelihood now consists of two plateaus. Again, the posterior follows the prior for the first 19 singular values. Since now measurements of entries along the same antidiagonal are identical, less information is to be extracted from the measurements. This explains the first drop of Figure 1b at the 19th singular value for both the likelihood and posterior. Less information also means that we can expect our estimate to be worse compared to the white noise case. The relative errors are now indeed higher, as seen in Table 1. Note however that the estimate obtained by truncating the SVD remains the same.

6.3. Bayesian learning of MNIST classifier. In this application we learn a classifier for images of 10 handwritten digits. The classifier is trained on the MNIST data [23], which consists of 60,000 pictures for training and 10,000 pictures for test. Each picture $\boldsymbol{x}_n$ is of size 28×28. We pick 10,000 random samples from the training set and convert each picture $\boldsymbol{x}_n$ into $25^2=625$ random Fourier features $\boldsymbol{\varphi}(\boldsymbol{x}_n)_j=\mathrm{Re}(e^{-i\,\boldsymbol{v}_j^T\boldsymbol{x}_n})$ [32]. The 625 frequency vectors $\boldsymbol{v}_j$ are sampled from a zero-mean Gaussian with variance $1/5^2\,\boldsymbol{I}$. We use a one-vs-all strategy by learning 10 classifiers at once. Each classifier is trained to distinguish between 1 particular class versus all others. The forward model for our 10 classifiers is then $\boldsymbol{y}=\boldsymbol{\varphi}(\boldsymbol{x})\,\boldsymbol{W}+\boldsymbol{e}$. Each column of $\boldsymbol{W}\in\mathbb{R}^{625\times 10}$ contains the model parameters of one specific classifier. In order to predict the class of a sample $\boldsymbol{x}^*$, we compute $\boldsymbol{y}^*=\boldsymbol{\varphi}(\boldsymbol{x}^*)\,\boldsymbol{W}$ and apply the softmax function

$$\boldsymbol{\sigma}(\boldsymbol{y}^*)=\frac{e^{\boldsymbol{y}_k^*}}{\sum_k e^{\boldsymbol{y}_k^*}}\in\mathbb{R}^{10}.$$

The prediction is then the class with maximal $\boldsymbol{\sigma}(\boldsymbol{y}^*)$. The 10 classifiers are trained on a training data set of pictures $\boldsymbol{X}\in\mathbb{R}^{10,00\times 784}$ and corresponding class labels $\boldsymbol{Y}\in\mathbb{R}^{10,000\times 10}$.



(a) When $\sigma_P^2 = 10^{-6}$ large differences between the different posteriors are observed. The corresponding classifiers are therefore expected to also behave differently.

(b) When $\sigma_P^2 = 10^{-3}$ all differences between the different posteriors have almost vanished. The corresponding classifiers are expected to also behave similarly.

Figure 2. *Singular values of the square-root precision matrices of the posterior distribution for four different priors. The noise variance is fixed to* 1.

Our estimate for $\boldsymbol{W}$ is the mean of the posterior $p(\boldsymbol{W}|\boldsymbol{Y},\boldsymbol{X})$. The residual $\boldsymbol{e}$ is most commonly assumed to be zero-mean white Gaussian noise $p(\boldsymbol{e}) = \mathcal{N}(0, \sigma_\epsilon^2 \boldsymbol{I})$. Likewise, the prior $p(\boldsymbol{W})$ is usually assumed to be a zero-mean normal distribution with a uniform scaling covariance matrix $\boldsymbol{P}_0 = \sigma_P^2\, \boldsymbol{I}$. Such a prior is also called Tikhonov regularization. We compare the performance of the Tikhonov prior to other zero-mean $(\boldsymbol{A}, \boldsymbol{b})$-constrained tensor priors (symmetric, Hankel en circulant), constructed using either Theorem 4.5 or Theorem 5.1. The noise variance σ_ϵ^2 is set to a fixed value of 1. The difference between these different priors in terms of the low-dimensional subspace they specify can be investigated by looking at the singular value profiles of the square-root precision matrices of the corresponding posteriors. These are shown in Figure 2a for $\sigma_P^2 = 10^{-6}$ and in Figure 2b for $\sigma_P^2 = 10^{-3}$. Being confident in the prior ($\sigma_P^2 = 10^{-6}$) has a strong effect on the corresponding posterior, which explains the large differences in singular value profiles. The structured tensor priors are characterized by a relatively quick decay of singular values due to the low-dimensional subspace they enforce compared to the Tikhonov prior, which considers the whole vector space $\mathbb{R}^{625}$. This favors the posterior solution for the classifier parameters to be described by less than 625 independent parameters. The corresponding classifiers can then be expected to also differ a lot on unseen test data. Indeed, applying the obtained classifiers on $10,000$ test images results in a relative number of correctly classified images shown in Table 2. All $(\boldsymbol{A}, \boldsymbol{b})$-constrained priors outperform the conventional Tikhonov prior, with Hankel and circulant tensors having the best performance. These results indicate that the 625 model is overparameterized and that the structured tensor prior is able to remove this model redundancy effectively. By increasing the prior covariance to $\sigma_P^2 = 10^{-3}$ all singular value profiles become very similar, all limiting the total number of "effective" model parameters in the same way. Unsurprisingly, the corresponding classifiers have similar performance as seen in Table 2. No significant classification improvement is observed for the Hankel and circulant priors.

Table 2

Comparison of relative number of correctly classified test images for classifiers learned with different priors. Best classifier indicated in bold.

	Tikhonov	symmetric	Hankel	circulant
$\sigma_P^2 = 10^{-6}$	0.650	0.880	**0.917**	0.915
$\sigma_P^2 = 10^{-3}$	0.917	0.918	**0.920**	0.919

7. Conclusions.

A whole new class of Bayesian priors has been worked-out which could be potentially applied to a variety of different inverse problems. The main focus of this article was mostly on the theoretical foundation and where possible we discussed practical implementations without going into much detail. Although the curse of dimensionality when considering tensors of large order and dimension can be completely resolved via the corresponding dual problem (5.1) as discussed in section 5.2, the computational complexity can still become prohibitively large with increasing sample size. To tackle this complexity, the possibility to represent the prior mean vector and covariance matrix of these priors as exact low-rank tensor decompositions could be investigated.

Acknowledgments. Many thanks to Frederiek Wesel for valuable discussions and feedback. I would also like to sincerely thank the reviewers, whose comments have significantly improved the quality of the manuscript.

REFERENCES

[1] M. Amiridi, N. Kargas, and N. D. Sidiropoulos, *Low-rank characteristic tensor density estimation part* i*: Foundations*, IEEE Trans. Signal Process., 70 (2022), pp. 2654–2668, https://doi.org/10.1109/TSP.2022.3175608.

[2] J. M. Bardsley, *Computational Uncertainty Quantification for Inverse Problems: An Introduction to Singular Integrals*, SIAM, 2018, https://doi.org/10.1137/1.9781611975383.

[3] K. Batselier, *Enforcing symmetry in tensor network MIMO Volterra identification*, IFAC-PapersOnLine, 54 (2021), pp. 469–474, https://doi.org/10.1016/j.ifacol.2021.08.404.

[4] K. Batselier, *Low-rank tensor decompositions for nonlinear system identification: A tutorial with examples*, IEEE Control Syst., 42 (2022), pp. 54–74, https://doi.org/10.1109/MCS.2021.3122268.

[5] K. Batselier, Z. Chen, H. Liu, and N. Wong, *A tensor-based Volterra series black-box nonlinear system identification and simulation framework*, in 2016 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), IEEE, 2016, pp. 1–7.

[6] K. Batselier, Z. Chen, and N. Wong, *Tensor Network alternating linear scheme for MIMO Volterra system identification*, Automatica, 84 (2017), pp. 26–35, https://doi.org/10.1016/j.automatica.2017.06.033.

[7] K. Batselier and N. Wong, *A constructive arbitrary-degree Kronecker product decomposition of tensors*, Numer. Linear Algebra Appl., 24 (2017), https://doi.org/10.1002/nla.2097.

[8] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, *Julia: A fresh approach to numerical computing*, SIAM Rev., 59 (2017), pp. 65–98, https://doi.org/10.1137/141000671.

[9] M. Blondel, M. Ishihata, A. Fujino, and N. Ueda, *Polynomial networks and factorization machines: New insights and efficient training algorithms*, in International Conference on Machine Learning, PMLR, 2016, pp. 850–858.

[10] J. Chung and S. Gazzola, *Computational methods for large-scale inverse problems: A survey on hybrid projection methods*, SIAM Rev., 66 (2024), pp. 205–284, https://doi.org/10.1137/21M1441420.

[11] J. Chung and A. K. Saibaba, *Generalized hybrid iterative methods for large-scale Bayesian inverse problems*, SIAM J. Sci. Comput., 39 (2017), pp. S24–S46, https://doi.org/10.1137/16M1081968.

[12] H. F. de Groote, *On varieties of optimal algorithms for the computation of bilinear mappings* i. *the isotropy group of a bilinear mapping*, Theort. Comput. Sci. Comput. Sci., 7 (1978), pp. 1–24, https://doi.org/10.1016/0304-3975(78)90038-5.

[13] W. Ding, L. Qi, and Y. Wei, *Fast Hankel tensor–vector product and its application to exponential data fitting*, Numer. Linear Algebra Appl., 22 (2015), pp. 814–832, https://doi.org/10.1002/nla.1970.

[14] C. L. Epstein, *Introduction to the Mathematics of Medical Imaging*, SIAM, 2007.

[15] D. F. Gleich, L.-H. Lim, and Y. Yu, *Multilinear pagerank*, SIAM J. Matrix Anal. Appl., 36 (2015), pp. 1507–1541, https://doi.org/10.1137/140985160.

[16] G. H. Golub and C. F. Van Loan, *Matrix Computations*, JHU Press, 2013.

[17] P. C. Hansen, J. G. Nagy, and D. P. O'leary, *Deblurring images: Matrices, spectra, and filtering*, SIAM, 2006, https://doi.org/10.1137/1.9780898718874.

[18] T.-K. Huang and J. Schneider, *Learning hidden Markov models from non-sequence data via tensor decomposition*, Adv. Neural Inf. Process. Syst., 26 (2013).

[19] N. Kargas and N. D. Sidiropoulos, *Supervised learning and canonical decomposition of multivariate functions*, IEEE Trans. Signal Process., 69 (2021), pp. 1097–1107, https://doi.org/10.1109/TSP.2021.3055000.

[20] V. A. Kazeev, B. N. Khoromskij, and E. E. Tyrtyshnikov, *Multilevel Toeplitz matrices generated by tensor-structured vectors and convolution with logarithmic complexity*, SIAM J. Sci. Comput., 35 (2013), pp. A1511–A1536, https://doi.org/10.1137/110844830.

[21] C.-Y. Ko, K. Batselier, L. Daniel, W. Yu, and N. Wong, *Fast and accurate tensor completion with total variation regularized tensor trains*, IEEE Trans. Image Process., 29 (2020), pp. 6918–6931, https://doi.org/10.1109/TIP.2020.2995061.

[22] T. G. Kolda and B. W. Bader, *Tensor decompositions and applications*, SIAM Rev., 51 (2009), pp. 455–500, https://doi.org/10.1137/07070111X.

[23] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, *Gradient-based learning applied to document recognition*, Proc. IEEE, 86 (1998), pp. 2278–2324.

[24] G. Li, L. Qi, and G. Yu, *The Z-eigenvalues of a symmetric tensor and its application to spectral hypergraph theory*, Numer. Linear Algebra Appl., 20 (2013), pp. 1001–1029, https://doi.org/10.1002/nla.1877.

[25] W. Li and M. K. Ng, *On the limiting probability distribution of a transition probability tensor*, Linear Multilinear Algebra, 62 (2014), pp. 362–385, https://doi.org/10.1080/03081087.2013.777436.

[26] J. Liu, P. Musialski, P. Wonka, and J. Ye, *Tensor completion for estimating missing values in visual data*, IEEE Trans. Pattern Anal. Mach. Intell., 35 (2012), pp. 208–220, https://doi.org/10.1109/TPAMI.2012.39.

[27] N. Mastronardi, P. Lemmerling, and S. Van Huffel, *Fast structured total least squares algorithm for solving the basic deconvolution problem*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 533–553, https://doi.org/10.1137/S0895479898345813.

[28] A. Novikov, I. Oseledets, and M. Trofimov, *Exponential machines*, Bull. Polish Acad. Sci.: Techn. Sci., 66 (2018), pp. 789–797, (Special Section on Deep Learning: Theory and Practice).

[29] G. Pillonetto and G. De Nicolao, *A new kernel-based approach for linear system identification*, Automatica, 46 (2010), pp. 81–93, https://doi.org/10.1016/j.automatica.2009.10.031.

[30] P. Jupyter, M. Bussonnier, J. Forde, J. Freeman, B. Granger, T. Head, C. Holdgraf, K. Kelley, G. Nalvarte, M. P. Andrew Osheroff, Y. Panda, F. Perez, B. R. Kelley, and C. Willing, *Binder* 2.0 *- Reproducible, interactive, sharable environments for science at scale*, in Proceedings of the 17th Python in Science Conference, Fatih Akici, D. Lippa, D. Niederhut, and M. Pacer, eds., 2018, pp. 113–120.

[31] L. Qi, H. Chen, and Y. Chen, *Tensor Eigenvalues and their Applications*, Vol. 39, Springer, 2018.

[32] A. Rahimi and B. Recht, *Random features for large-scale kernel machines*, Adv. Neural Inf. Process. Syst., 20 (2007).

[33] S. Särkkä and L. Svensson, *Bayesian Filtering and Smoothing*, Vol. 17, Cambridge University Press, 2023.

[34] E. Stoudenmire and D. J. Schwab, *Supervised learning with tensor networks*, Adv. Neural Inf. Process. Syst., 29 (2016).

[35] J. A. K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle, *Least Squares Support Vector Machines*, World Scientific, Singapore, 2002.

[36] F. van der Plas and M. Bochenski, *fonsp/pluto.jl: v*0.19.42, 2024, https://doi.org/10.5281/zenodo.16882845.

[37] T. Van Gestel, J. A. Suykens, P. Van Dooren, and B. De Moor, *Identification of stable models in subspace identification by using regularization*, IEEE Trans. Automat. Control, 46 (2001), pp. 1416–1420, https://doi.org/10.1109/9.948469.

[38] S. Wahls, V. Koivunen, H. V. Poor, and M. Verhaegen, *Learning multidimensional Fourier series with tensor trains*, in 2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP), IEEE, 2014, pp. 394–398.

[39] F. Wesel and K. Batselier, *Large-Scale Learning with Fourier Features and Tensor Decompositions*, Adv. Neural Inf. Process. Syst., 34 (2021), pp. 17543–17554.

[40] C. K. Williams and C. E. Rasmussen, *Gaussian Processes for Machine Learning*, Vol. 2, MIT Press, Cambridge, MA, 2006.

[41] H. Xiang and J. Zou, *Regularization with randomized SVD for large-scale discrete inverse problems*, Inverse Problems, 29 (2013), 085008, https://doi.org/10.1088/0266-5611/29/8/085008.

[42] H. Zheng, C. Zhou, Z. Shi, and Y. Gu, *Structured tensor reconstruction for coherent DOA estimation*, IEEE Signal Process. Lett., 29 (2022), pp. 1634–1638, https://doi.org/10.1109/LSP.2022.3190768.