Bayesian Multi-View Tensor Factorization

Suleiman A Khan[1] and Samuel Kaski [1,2]

Helsinki Institute for Information Technology HIIT,
[1] Department of Information and Computer Science, Aalto University, Finland
[2] Department of Computer Science, University of Helsinki, Finland
{first.last}@aalto.fi

Abstract. We introduce a Bayesian extension of the tensor factorization problem to multiple coupled tensors. For a single tensor it reduces to standard PARAFAC-type Bayesian factorization, and for two tensors it is the first Bayesian Tensor Canonical Correlation Analysis method. It can also be seen to solve a tensorial extension of the recent Group Factor Analysis problem. The method decomposes the set of tensors to factors shared by subsets of the tensors, and factors private to individual tensors, and does not assume orthogonality. For a single tensor, the method empirically outperforms existing methods, and we demonstrate its performance on multiple tensor factorization tasks in toxicogenomics and functional neuroimaging.

1 Introduction

Tensor Factorization methods decompose data into underlying latent factors or components, taking advantage of the natural tensor structure in the data. A wide range of low-dimensional representations of tensors have been proposed earlier [1]. The most well-known models include the CP CANDECOMP/PARAFAC [2,3] and the Tucker 3-mode factor analysis [4]. Tucker is a more generic model for complex interactions, whereas CP as an additive combination of rank-1 contributions is easier interpretable analogously to matrix factorizations. Recently well-regularized probabilistic tensor factorization methods have been introduced for both CP [5] and Tucker [6], though they are limited to single tensors only.

Two-view tensor models. In order to discover shared patterns between two co-occuring tensors, joint factorization approaches decompose them into correlated factors [7]. Recently, several non-probabilistic methods for Tensor Canonical Correlation Analysis have been introduced [8,9,10] extending the matrix counterparts. The methods impose different constraints but all aim at finding a common latent representation of two paired tensors.

Two-view matrix models. For paired matrices, integration approaches have been thoroughly studied. For an overview on nonlinear Canonical Correlation Analysis (CCA) see [11] and Bayesian CCA see [12].

Multi-view models. Multi-view modeling integrates information from multiple coupled datasets. For unsupervised multi-view modelling, a method has recently been proposed for decomposing several coupled matrices, into components shared by subsets of the matrices, and components private to each matrix.

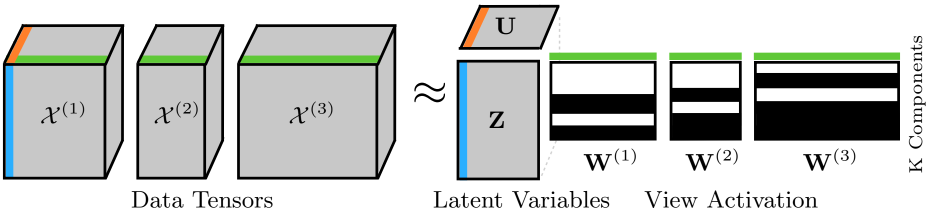

Fig. 1. Multi-view tensor factorization. Datasets $\mathcal{X}^{(1)}, \mathcal{X}^{(2)}, \mathcal{X}^{(3)}$ are simultaneously decomposed into K components. The $\mathbf{Z}$ and $\mathbf{U}$ loadings are common to all tensors, while the view-specific loadings $\mathbf{W}^{(m)}$ show the intrinsic component-view structure in the data. The structure is highlighted in $\mathbf{W}^{(m)}$ with *black* representing a component active in a view (non-zero loadings), while *white* is switched off (zero-loadings).

The method was called Group Factor Analysis [13]. As far as we know, methods for analysing multiple coupled tensors have not been proposed earlier.

In this paper we formulate and address the novel *multi-view tensor factorization problem*, where the task is to decompose multiple coupled or co-occuring tensors into factors that are shared by subsets of the tensors: one, some or all of them. We formulate a Bayesian model to solve the task, allowing automatic model complexity selection and an intrinsic solution for degeneracies. For two views, our model is the first Bayesian Tensor Canonical Correlation Analysis.

The rest of the paper is structured as follows: In section 2 we formulate the novel multi-view tensor factorization problem. In section 3 we present our Bayesian multi-view tensor factorization model and describe its relationship to existing works. In section 4 we validate the model's performance in various settings and demonstrate its application in a novel toxicogenomics setting and a neuroimaging case. We conclude with discussion in section 5.

Notations: We will denote a tensor as $\mathcal{X}$, a matrix $\mathbf{X}$, vector $\mathbf{x}$ and a scalar x. The Frobenius norm of a tensor is defined as $\|\mathcal{X}\| = \sqrt{\sum_n \sum_d \sum_l \mathcal{X}^2_{n,d,l}}$. The Mode-2 product $\times_2$ between a tensor $\mathcal{A} \in \mathbb{R}^{N \times K \times L}$ and a matrix $\mathbf{B} \in \mathbb{R}^{D \times K}$ is the projected tensor $(\mathcal{A} \times_2 \mathbf{B}) \in \mathbb{R}^{N \times D \times L}$. A reshaped Khatri Rao product $\odot$ of two matrices $\mathbf{A} \in \mathbb{R}^{N \times K}$ and $\mathbf{C} \in \mathbb{R}^{L \times K}$ is the "column-wise matched" outer product of K vector-pairs that results in the tensor $(\mathbf{A} \odot \mathbf{C}) \in \mathbb{R}^{N \times K \times L}$. The outer product of two vectors is denoted $\circ$. The *rank* of a tensor $\mathcal{X}$ is the smallest number of rank-1 tensors that generate $\mathcal{X}$ as their sum. The *order* of a tensor is the number of axes in the tensors, also called ways or modes. For notational simplicity the model is presented for third order tensors, while it is trivially extendable to higher orders.

2 Multi-View Tensor Factorization

We formulate the novel Multi-view Tensor Factorization (MTF) problem for a collection of $m = 1, \ldots, M$ paired tensors (views), $\mathcal{X}^{(1)}, \mathcal{X}^{(2)}, \ldots, \mathcal{X}^{(M)} \in \mathbb{R}^{N \times D_m \times L}$, as the combined factorization that decomposes the tensors into factors shared between all, some, or a single tensor. In MTF, each tensor is factorized into a view-specific matrix of loadings $\mathbf{W}^{(m)} \in \mathbb{R}^{D_m \times K}$ and a low-dimensional tensor $\mathcal{Y} \in \mathbb{R}^{N \times K \times L}$ common for all views:

$$\mathcal{X}^{(m)} = \mathcal{Y} \times_2 \mathbf{W}^{(m)} + \boldsymbol{\epsilon}^{(m)} .$$

Here $\boldsymbol{\epsilon}^{(m)} \in \mathbb{R}^{N \times D_m \times L}$ is the noise tensor.

The view-specific matrix of loadings $\mathbf{W}^{(m)}$ then controls which of the factors k from the common tensor are active in each view. For convenience we assume a fixed number of K factors, with the understanding that for methods capable of choosing the number of factors, K is set large enough, and the loadings of extra components will automatically become set to zero.

The tensor $\mathcal{Y}$ forms the shared latent tensor and can be left unconstrained (equivalent to Tucker1 factorization), or can be further constrained to represent any decomposition including Tucker2, Tucker3 or CP. The CP decomposition factorizes a tensor into a sum of rank-1 tensors, where each rank-1 tensor is the outer product of vector loadings in all modes, whereas in Tucker variants the factor interactions are modelled via a core tensor $\mathcal{G}$. This rank-1 component decomposition of CP and its intrinsic axis property from parallel proportional profiles [14], along with uniqueness of solutions [15], gives it a very strong interpretive power. The Tucker model is more flexible, though, the complex interactions via $\mathcal{G}$ and non-uniqueness of solutions make its interpretation more difficult. Therefore, we adapt an underlying CP decomposition for our model.

Figure 1 illustrates MTF for the joint CP-type factorization. More formally,

$$\begin{aligned} \mathcal{X}^{(m)} &= \sum_{k=1}^{K} \mathbf{Z}_k \circ \mathbf{U}_k \circ \mathbf{W}_k^{(m)} + \boldsymbol{\epsilon}^{(m)} \qquad (1) \\ &= (\mathbf{Z} \odot \mathbf{U}) \times_2 \mathbf{W}^{(m)} + \boldsymbol{\epsilon}^{(m)} . \end{aligned}$$

Here $\mathbf{Z} \in \mathbb{R}^{N \times K}$ and $\mathbf{U} \in \mathbb{R}^{L \times K}$ are the common latent variables and the $\mathbf{W}^{(m)}$ are loadings for each view m.

Figure 1 shows the MTF formulation for three tensors, where components (rows) of $\mathbf{W}^{(m)}$ can be active in all, two, or a single view. The loadings $\mathbf{W}_k^{(m)}$ are zero for the components k that are not active in view m. A component active in two or more views has non-zero loadings in the corresponding $\mathbf{W}_k^{(m)}$ and is hence shared between them. This specification comprehensively represents the intrinsic structure of the tensor collection.

3 Bayesian Multi-View Tensor Factorization

We formulate a Bayesian treatment of the MTF problem of Equation 1, by complementing it with priors for model parameters. Figure 2 summarizes the dependencies between the variables in the decomposition of the M observed tensors $\mathcal{X}^{(m)}$ as a graphical model. The main idea is incorporated in plate M, which represents the view-specific loadings $\mathbf{W}^{(m)}$, having two layers of sparsity:

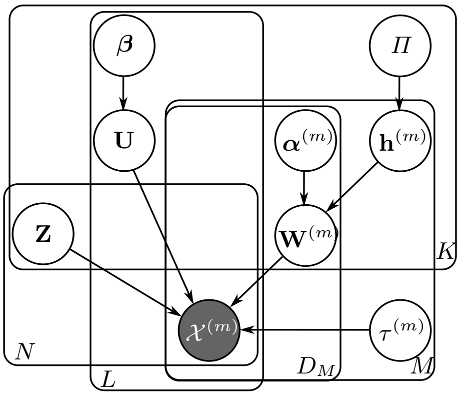

Fig. 2. Plate diagram for Bayesian multi-view tensor factorization.

1) *view-wise* sparsity controlled by $\mathbf{h}^{(m)}$ and 2) *feature-wise* sparsity (across the D_M features) controlled by $\boldsymbol{\alpha}^{(m)}$. The view-wise sparsity acts as an on/off switch and allows the model to automatically learn which views share each factor, and also the total number of factors in the data. The plate K represents probabilistic CP decomposition for each view, where $\mathbf{Z}$ and $\mathbf{U}$ are the latent variables.

The distributional assumptions of our model (explained in detail below) are:

$$\begin{aligned}
\mathcal{X}_{n,l}^{(m)} &\sim \mathcal{N}((\mathbf{Z}_n \odot \mathbf{U}_l) \times_2 \mathbf{W}^{(m)}, \mathbf{I}(\tau^{(m)})^{-1}) \\
\mathbf{Z} &\sim \mathcal{N}(0, I) \\
\mathbf{U}_{l,k} &\sim \mathcal{N}(0, (\boldsymbol{\beta}_{l,k})^{-1}) \\
\mathbf{W}_{d,k}^{(m)} &\sim \mathbf{h}_k^{(m)} \mathcal{N}(0, (\boldsymbol{\alpha}_{d,k}^{(m)})^{-1}) + (1 - \mathbf{h}_k^{(m)})\delta_0 \\
\mathbf{h}_k^{(m)} &\sim \mathit{Bernoulli}(\pi_k) \\
\pi_k &\sim \mathit{Beta}(a^{\pi}, b^{\pi}) \\
\boldsymbol{\beta}_{l,k} &\sim \mathit{Gamma}(a^{\beta}, b^{\beta}) \\
\boldsymbol{\alpha}_{d,k}^{(m)} &\sim \mathit{Gamma}(a^{\alpha}, b^{\alpha}) \\
\tau^{(m)} &\sim \mathit{Gamma}(a^{\tau}, b^{\tau})
\end{aligned}$$

where $Gamma(a, b)$ is parameterized by *shape* a, *rate* b.

The coupled $N \times L$ samples in each tensor $\mathcal{X}^{(m)}$ are modelled via the product of loadings, with a view-specific observation precision $\tau^{(m)}$. For the latent variables, we assume *a priori* independence, and induce an element-wise automatic relevance determination ARD prior [16] on $\mathbf{U}_{l,k}$ to encourage sparsity.

To infer the interactions between views and components, we make the model view-wise sparse via a Spike and Slab prior [17] on the projection weights $\mathbf{W}^{(m)}$. The spike and slab prior has two parts, one being a delta δ_0 function centered at zero and the other some continuous distribution (usually Gaussian). We replace the Gaussian with an element-wise ARD prior to additionally allow feature-level sparsity in our model. The ARD is a Normal-Gamma prior that specifies the precision $\boldsymbol{\alpha}_{d,k}^{(m)}$ controling the scale of each variable. Our element-wise d, k, m formulation of ARD encourages the loadings within a component-view pair to be sparse. In the spike and slab construct, the binary value $\mathbf{h}_k^{(m)}$ drawn from a Bernoulli distribution gives the component-view activation. If $\mathbf{h}_k^{(m)} = 1$, the component k is active in view m and the loadings $\mathbf{W}_k^{(m)}$ are sampled from a corresponding element-wise ARD prior, whereas if $\mathbf{h}_k^{(m)} = 0$, the component-view pair is not active and the loadings $\mathbf{W}_k^{(m)}$ are set to zero via δ_0, inducing view-wise sparsity.

Learning the $\mathbf{h}^{(m)}$ activities allows automatic determination of the number and sharing of factors between the views. This is because if K is set to be large enough, the model will switch off $\mathbf{h}_k^{(m)}$, for all the extra k, m pairs. This yields the underlying sharing pattern of the views, even producing empty components that are not active in any view. The presense of empty components indicates that K was set to a large enough value, and the amount of non-empty components gives the rank of the view collection. In the construct, π_k represents probability of activation of each component.

The joint probability of data and parameters can be factorized as follows, and inference is performed via Gibbs sampling:

$$\begin{aligned}p(\mathcal{X}^{(1)}, \mathcal{X}^{(2)}, ..., \mathcal{X}^{(M)}, \Theta) = &\prod_{m=1}^{M} \prod_{n=1}^{N} \prod_{l=1}^{L} p(\mathcal{X}_{n,l}^{(m)} | \mathbf{Z}_n, \mathbf{U}_l, \mathbf{W}^{(m)}, \tau^{(m)}) \\ &p(\tau^{(m)}) p(\mathbf{Z}_n) \prod_{k=1}^{K} p(\mathbf{U}_{l,k} | \boldsymbol{\beta}_{l,k}) p(\boldsymbol{\beta}_{l,k}) \\ &\prod_{d=1}^{D^{(m)}} p(\mathbf{W}_{d,k}^{(m)} | \boldsymbol{\alpha}_{d,k}^{(m)}, \mathbf{h}_k^{(m)}) p(\boldsymbol{\alpha}_k^{(m)}) . p(\mathbf{h}_k^{(m)} | \pi_k) p(\pi_k)\end{aligned}$$

Degeneracies can complicate the practical use of CP when analyzing real data [18]. Most degeneracies occur due to non-trilinear structure in the data and are identified by strong negative correlations between two components. To overcome the problem, researchers have proposed adding orthogonality and non-negativity constraints that address it by hindering correlations [18,19], but may also effect the model's ability to discover PARAFAC's intrinsic axes.

In our Bayesian formulation, we impose an element-wise ARD prior on the component loadings $\mathbf{W}^{(m)}$,$\mathbf{U}$. The element-wise prior regularizes the solution allowing determination of precise factor loadings, and is a construct less strict than orthogonality. Our model should therefore be able to handle weak degeneracies, via a flexible composition that still allows identifying PARAFAC's intrinsic axes.

3.1 Special Cases and Related Problems

We next present special cases of our model and relate them to the existing works.

Sparse Bayesian CP. For $m = 1$ (a single view) our model reduces to sparse Bayesian CP factorization, which can automatically infer the number of components. In this special case our formulation goes very close to the Bayesian CP [20], the main differences being that they use MAP estimation and do not have feature-level sparsity.

Other Bayesian versions of CP include a variant specialized for temporal datasets [5], the fully conjugate model [21], and an exponential family framework [22]. For Tucker factorizations, Chu and Ghahramani [6] formulated Tucker in a probabilistic framework (pTucker) while [23] presented a non-linear variant using Gaussian processes. All of these follow different assumptions; however, unlike our method, none of them automatically learns the rank of the tensors. Instead, repetitive methods of rank identification are used, though they pose serious scalablity issues for large tensors [1].

Bayesian Tensor CCA. For $m = 2$, our model is the first Bayesian Tensor CCA. The model is related to tensor-CCAs in the classical domain, specifically to [8,10]. An additional technical difference, besides our Bayesian treatment, is that the earlier works assume the two tensors to be paired in a single mode (N), while we assume pairing in both N and L. Both settings are sensible and applicability depends on the nature of the data.

There have also been fusion studies on coupled matrix-tensor factorization, where values in a tensor were predicted with side information from a matrix, or vice versa. A gradient-based least squares optimization approach was presented in [24], while [25,26] used generalized linear models in a coupled matrix-tensor factorization framework to solve link prediction and audio processing tasks.

In the matrix domain, a related multi-view problem was recently studied under the name of Group Factor Analysis [13]. The goal there was to perform a joint factor analysis of multiple matrices to find relationships between datasets. Their method also finds components shared between subsets of views but, naturally, works only for matrices.

4 Experiments

We have applied our model on both simulated and real datasets. We will first demonstrate in a simulated example the model's ability to correctly separate shared and view-specific components, as well as precisely identify the factor mode loadings. We next compare our model to the existing state-of-the-art methods on benchmark single-view datasets, to validate that in the single-view special case our algorithms are comparable. We then validate our model's performance on simulated multi-view tensors and compare to the single-view tensor methods and the multi-view matrix methods as the existing baselines, ascertaining the advantage gained by the multi-view tensor decomposition. Finally, we apply our method on multi-view real data tensors on a new problem from toxicogenomics and a functional neuroimaging dataset, demonstrating the interpretative power and diverse applicability of the model.

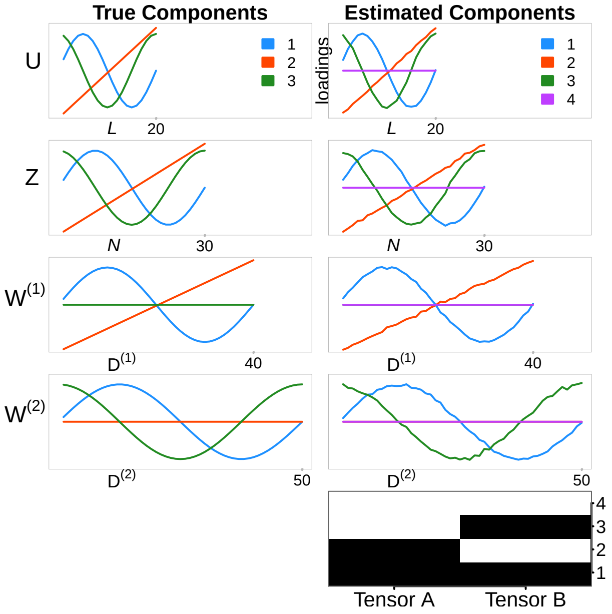

Fig. 3. Demonstration of BMTF decomposing two tensors $\mathcal{A}$ and $\mathcal{B}$ simultaneously, finding the one shared and two view-specific components. **Left:** Loadings are drawn for the three components (1 shared, 2 specific) embedded in the data. **Right-Bottom:** component-view activation $\mathbf{h}_k^{(m)}$ for a $K = 4$ BMTF run. **Right:** Loadings of the four BMTF components reveal the shared and specific components.

Our model detects the number and type of components automatically, as long as it is run with a large enough K, resulting in the extra components getting zero loadings. The practical procedure we followed is to increase K until empty components are found. The experiments were run with the hyperparameters $a^\pi, b^\pi, a^\alpha, b^\alpha, a^\beta, b^\beta, a^\tau, b^\tau$ initialized to 1. To account for high noise in real datasets, the noise hyperparameters a^τ, b^τ were initialized assuming a signal-to-noise ratio of 1. All remaining model parameters were learned using Gibbs sampling while discarding the first 10,000 samples as the burn-in and using the next 10,000 samples for estimating the posterior. Our R implementation of the model is available at `http://research.ics.aalto.fi/mi/software/bmtf/`.

4.1 Simulated Illustration

We first demonstrate the ability of our BMTF to decompose the data into factors in a two-view setting. For this purpose two tensor datasets $\mathcal{A}$ and $\mathcal{B}$ were created using three underlying components, one of which is shared between both tensors, while one is specific to each. Figure 3-left shows the 3-mode loadings used to create the two tensors, where $\mathbf{Z}$ and $\mathbf{U}$ are the common 1^{st} and 2^{nd} mode loadings between both tensors while $\mathbf{W}^{(1)}$ and $\mathbf{W}^{(2)}$ are the 3^{rd} mode loadings for tensor $\mathcal{A}$ and tensor $\mathcal{B}$, respectively. The shared component (blue) has non-zero loadings in both $\mathbf{W}^{(1)}$ and $\mathbf{W}^{(2)}$ while the specific ones have non-zero $\mathbf{W}^{(m)}$ loadings in only the corresponding view.

BMTF was run with $K = 4$, i.e., larger than the number of embedded components (=3). Figure 3-bottom-right plots the learned $\mathbf{h}_k^{(m)}$ values for the $M = 2$ views and $K = 4$ components. The plot shows that one component is active in both views (black) while one component active in each view, demonstrating that the model correctly separates the shared and view-specific effects. The fourth component was rightly detected as not active in any of the views, as the data come from only three components, indicating that the model identifies the correct number of components by switching off the extra ones. The discovered loadings for the 4 components are plotted in Figure 3-right. The plots show that the loadings are identified correctly in this simulated example.

4.2 Single View

As discussed in Section 3.1, our method also solves the CP problem as a special case when run on a single dataset. We compare our formulation to the existing state-of-the-art single-view methods on benchmark datasets to validate that our performance is at least comparable. These single-view methods have not been generalized to multi-view tensors where our main contribution lies.

Comparison Methods. We compare to the following state-of-the-art approaches.

ARDCP: Mørup *et. al.* [20] formulated CP in a Bayesian framework and automatically learn the number of components, using MAP estimation. In comparison to them, our model is fully Bayesian and additionally element-wise sparse.

pTucker: Chu and Ghahramani [6] presented a probabilistic version of the Tucker model. Tucker is more flexible than CP, though not easy to interpret.

CP: We also compare to the most widely used and updated classical CP implementation from the *N-way Toolbox* (v3.31 of July 2013, `http://www.models.life.ku.dk/nwaytoolbox`). The implementation solves the factorization using the well established Alternating Least Squares ALS algorithm [27]. On the computational side, per-iteration complexity of BMTF exceeds ARDCP and CP only due to computing $K \times K$ covariance matrices, which is small compared to the rest of the computation. Tucker is costlier than CP as it needs to solve for the core tensor as well, while pTucker reduces its costs with custom solutions.

Datasets. We use the three commonly used benchmark datasets in tensor modeling from `http://www.models.life.ku.dk/nwaydata`, namely Amino Acids, Flow Injection Analysis, and Kojima Girls datasets.

We test our model for both its ability to find the number of components and to model the data correctly in a missing value setting. We randomly selected half of the values in the datasets for training the models and predicted the remaining half. The split was repeated independently 100 times. BMTF and ARDCP learned the number of components for each split. CP and pTucker were run with the number of components estimated from the full data using the defacto standard *pftest* from *N-way toolbox* [27].

Table 1. Detection of number of factors, and ability to find the intrinsic structure. The table lists the number of factors of the three real datasets determined by *pftest (on full data)* from *N-way Toolbox* and compares the ability of BMTF with other state of the art methods in a) learning the number of factors and b) prediction error, when data contains missing values.

Data set	Amino Acid	Flow Injection	Kojima Girls
Size	5 x 201 x 61	12 x 50 x 45	4 x 153 x 20
Factors			
pftest	3	4	2
BMTF	3.0 ± 0.0	4.5 ± 0.5	2.0 ± 0.1
ARDCP	3.1 ± 0.3	4.0 ± 0.0	1.2 ± 0.4
Prediction RMSE			
BMTF	0.0257±0.0003	**0.045±0.010**	**0.189±0.025**
ARDCP	0.0278±0.0035	0.065±0.001	0.305±0.051
CP	0.0256±0.0003	0.053±0.001	1.643±4.098
pTucker	**0.0250±0.0003**	0.049±0.001	0.236±0.055

Results are presented in Table 1. Both BMTF and ARDCP recovered the number of components well despite 50% missing values, with the mean being close to the number obtained by *pftest* on *full data*. The result clearly shows that automatic component selection works even in the presence of missing values.

Prediction RMSE results for the first two datasets Amino Acids and Flow Injection show that all methods perform almost comparably and none goes exceedingly wrong, confirming that our method compares well with state-of-the-art single-view methods. The third dataset Kojima Girls shows a major difference in the performance of the methods. This dataset is known to have a degeneracy problem, and hence the standard CP fails to model the data correctly. ARDCP seems to perform better in comparison to CP, and close examination reveals that this is because ARDCP tends to skip the degenerate component as can be seen from the mean component number of 1.2. Using fewer components is one way of avoiding the effect of degeneracies. Our method does both, finding the correct number of components and being able to cope with degeneracies as is shown by the best performance. With its flexible parametrization the Tucker is also able to correctly model non-trilinear structure in the data, which is a characteristic of degeneracies [28]; hence does not suffers from the degeneracy problem.

4.3 Multi-View

To validate the performance of our model in multi-view settings, we applied it to simulated data sets that have all types of factors, i.e., factors specific to just one view, factors shared between a small subset of views and factors shared between most of the views. We show that the model can correctly discover the structure as the number of views is increased, while the baseline approaches are unable to find the correct result.

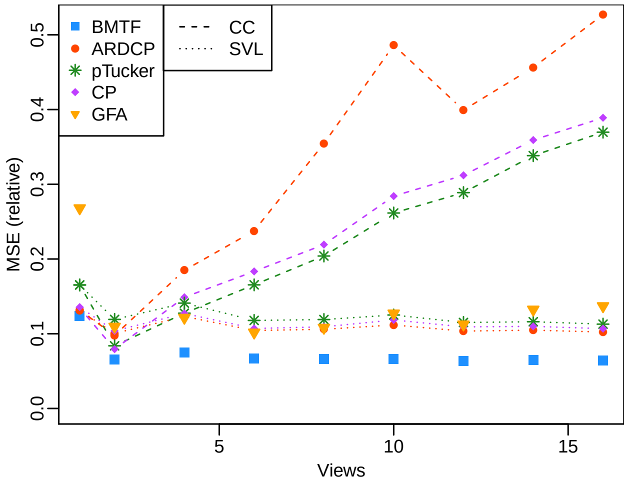

Fig. 4. Performance of Bayesian multi-view tensor factorization compared to single-view tensor methods and multi-view matrix methods (baselines). The number of views increases on the x-axis while the relative mean square error of recovering the underlying data is plotted on the y-axis. The single-view methods were tested in two settings a) CC marked with dashed-lines, where all the tensors are concatenated; b) SVL as dotted-lines, where models are learned for each tensor seperately.

We simulated a data set consisting of $M = 16$ views with dimensions $N = 20$, $L = 5$ and D_m randomly sampled between 10 and 100, using a manually constructed set of K=31 factors of the various types. For each component, the loadings $\mathbf{Z}_{:,k}$, $\mathbf{U}_{:,k}$ and $\mathbf{W}^{(m)}_{:,k}$ were randomly sampled from the standard normal distribution for all active m. For the non-active views m in the component k, the $\mathbf{W}^{(m)}_{:,k}$ were set to zero. The views were then created as:

$$\overline{\mathcal{X}}^{(m)} = \sum_k \mathbf{Z}_{:,k} \circ \mathbf{U}_{:,k} \circ \mathbf{W}^{(m)}_{:,k}$$

$$\mathcal{X}^{(m)} = \overline{\mathcal{X}}^{(m)} + \boldsymbol{\epsilon}^{(m)}$$

where $\overline{\mathcal{X}}^{(m)}$ is the true underlying data while $\boldsymbol{\epsilon}^{(m)}$ is a noise tensor sampled from a normal distribution with mean zero and variance equivalent to that of $\overline{\mathcal{X}}^{(m)}$.

We ran BMTF for $M = 1, \ldots, 16$. The single-view tensor methods were run in two settings, a) on a concatenation of all views [CC], b) single view learning [SVL], where a model is learned for each view seperatly. BMTF found the correct number of components in all cases while ARDCP[CC] failed to detect the correct number for $M \geq 4$. The other two methods, CP and pTucker, were run with the true number of factors. In single view learning [SVL], the methods were unable to identify the sharing between components, as they do not solve the multi-view

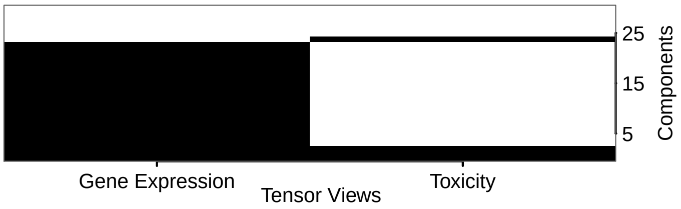

Fig. 5. Component activations in the toxicogenomics dataset indicate 3 shared components between the disease-specific gene expression responses and toxicity measurements of the drugs. The presence of several empty components indicates that $K = 30$ was enough to model the data.

problem addressed by BMTF. For completeness, we also compare our method to multi-view matrix FA (GFA) [13] by matricizing the tensors $\mathcal{X}^{(M)} \in \mathbb{R}^{N \times D_m \times L}$ into matrices $\mathbf{X}^{(M)} \in \mathbb{R}^{(N \times L) \times D_m}$.

We measured the models' performance in terms of the recovery error of the missing data. Defining $\hat{\mathcal{X}}^{(m)}$ as the model's estimate of the data, the recovery error is computed as the relative mean square error $\|\hat{\mathcal{X}}^{(m)} - \overline{\mathcal{X}}^{(m)}\|^2 / \|\mathcal{X}^{(m)} - \overline{\mathcal{X}}^{(m)}\|^2$ averaged over all the views.

Figure 4 plots the recovery error of our method as a function of the number of views. Our model's performance is stable as the number of views increases and outperforms all the baseline tensor and matrix alternatives. Single-view methods, applied to a data set which contains all tensors concatenated, deteriorate rapidly; while by learning each tensor seperately they are unable to discover the shared pattern. The matricized method (GFA) performs comparably to the single-view tensor methods. The experiment confirms that the specific multi-view tensor problem cannot be optimally solved with methods not designed for the purpose, and that our method fulfills its promise.

4.4 Application Scenarios and Interpretation

We next demonstrate the method at work on multi-view tensor datasets in potential use cases of BMTF. The first application represents a new problem at the juncture of toxicity and bioinformatics, while the second is a functional neuroimaging case.

Toxicogenomics. We analyzed a novel drug toxicity response problem, where the tensors arise naturally when gene expression responses of multiple drugs are measured for multiple diseases (different cancers) across the genes. The data contain two views, the measurement of post-treatment gene expression, and sensitivity of the cells to the drug. The key question that BMTF can answer is, which parts of the responses are specific to individual types of cancer and which occur across cancers, and which of them are related to drugs effectiveness. These patterns, if uncovered, can help understand the mechanisms of toxicity [29].

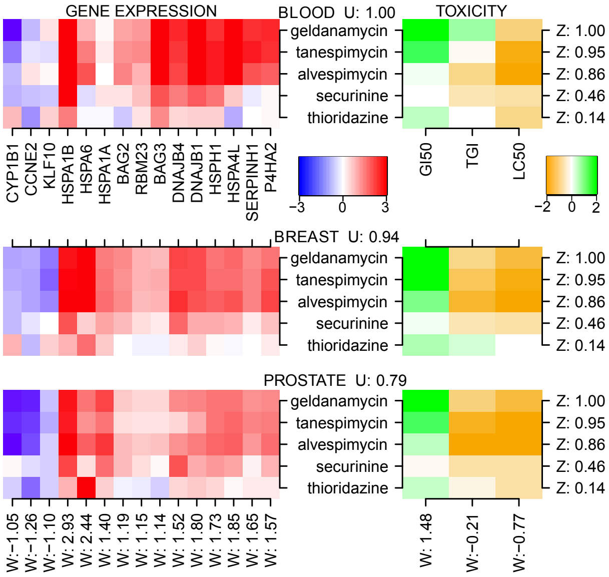

Fig. 6. Component 1 captures the well-known heatshock protein response. The top genes (left) and toxicity indicators (right) from the two views are plotted as columns, and the three different cancers as rows. The component links the strong upregulation of the heatshock protein genes (red) to high toxicity (green) in the top three drugs, all of which are heatshock protein inhibitors.

The dataset contained two views. The first, $m = 1$, contained the post-treatment differential gene expression responses $D_1 = 1106$ of several drugs $N = 78$ as measured over multiple cancer types $L = 3$. The second, $m = 2$, contained the corresponding drug sensitivity measurements $D_2 = 3$. The gene expression data were obtained from the connectivity map [30] that contained response measurements of three different cancers: Blood Cancer, Breast Cancer and Prostate Cancer. The data were processed so that gene expression values represent up (positive) or down (negative) regulation from the untreated (base) level. Strongly regulated genes were selected, resulting in $D_1 = 1106$. The drug screen data for the three cancer types were obtained from the NCI-60 database [31], measuring toxic effects of drug treatments via three different criteria: GI50 (50% growth inhibition), LC50 (50% lethal concentration) and TGI (total growth inhibition). The data were processed to represent the drug concentration used in the connectivity map to be positive when toxic, and negative when non-toxic.

BMTF was run with K=30, resulting in 3 components shared between both the gene expression and toxicity views, revealing that some patterns are indeed shared (Figure 5). These shared components form hypotheses about underlying biological processes that characterize toxic responses of the drugs.

Table 2. Prediction RMSE of BMTF in comparison to existing methods on toxicogenomics and neuroimaging datasets. The mean prediction performance over 100 runs of independent sets of missing values (50% missing) is given, along with one standard error of the mean. BMTF outperformed all other methods significantly with t-test p-values $< 10^{-6}$ on toxicogenomics data, and p-values $< 10^{-4}$ on neuroimaging data.

				CC		SVL	
		BMTF	GFA	ARDCP	CP	ARDCP	CP
Toxicogenomics	Mean	**0.4811**	0.5223	0.8919	5.3713	0.6438	5.0699
	StdError	**0.0061**	0.0041	0.0027	0.0310	0.0047	0.0282
Neuroimaging	Mean	**0.5105**	0.5144	0.6224	0.5740	0.5725	0.5611
	StdError	**0.0004**	0.0004	0.0003	0.0004	0.0003	0.0010

The first component captures the well-known "Heatshock Protein" response. The response is characterized by strong upregulation of heatshock genes in all cancers (Figure 6-left) and corresponding high toxicity indications (Figure 6-right). The response is being activated by the heat shock protein (HSP90) inhibitor drugs, all of which have the highest loadings in the component (the top three drugs). The HSP inhibition response has been well studied for treatment of cancers [32] evaluating its therapeutic efficacy. Had the biological action not already been discovered, our component could have been a key in revealing it.

Component 2 represents toxic mechanisms via inhibition of protein synthesis (details not shown) and Component 3 via damaging of cell DNA. Both of these components reveal interesting cancer type-specific findings, detailed interpretations of which are under way. The experiment validates that the model is able to find useful factors from multiple-tensor data.

We also evaluated BMTF for predicting missing values on the toxicogenomics data. BMTF outperformed the single-view methods[1] and matrix methods significantly with t-test p-values $< 10^{-6}$, on the prediction RMSE of 100 independent runs (Table 2). Additionally, the tensors of BMTF are easier to interpret than the corresponding $(L \times N) \times D_m$ matrices of matricized GFA, and the reformed tensors of single view CP.

Functional Neuroimaging. As the second demonstration we analysed a multi-view functional neuroimaging dataset, which comes from subjects exposed to multiple audiovisual stimuli. The data contained $M = 7$ views, representing the different audio and audiovisual stimuli, each composed of three songs. The different views are brain recordings made under different "presentations" of the same songs: purely auditory ones including singing (A:Sing), piano (A:Piano) *etc*, and audiovisual speaking with both voice and image of speaker (AV:Speech) *etc*. The views have a natural tensor structure where brain activity was recorded with fMRI from $L = 10$ subjects over the course of the experiment ($N = 162$ time points) in $D_m = 32$ regions of interest (data from [13]).

[1] pTucker failed to complete even on 50GB of RAM, hence was excluded.

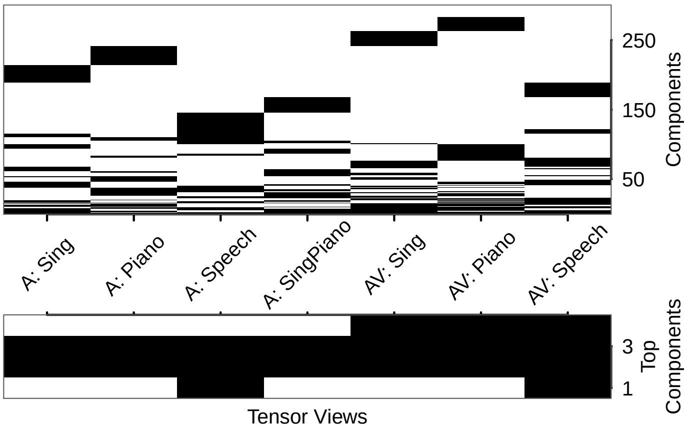

Fig. 7. Top: Component activations in the neuroimaging dataset. The components shared between subsets of views capture potentially interesting variation, separated from the view-specific "structured noise" or non-interesting variation. **Bottom:** Zoomed inset of top components based on subject (**U**) loadings. The first component is active in both speech views.

BMTF was run with $K = 300$ and the $\mathbf{h}^{(m)}$ profile is shown in Figure 7. The plot indicates that there exist several potentially interesting components shared between different subsets of views. The large number of view-specific components model "structured noise", i.e., mostly brain activity not related to the stimuli.

The goal of the fMRI study was to find responses that generalize across the subjects and describe relationships of the different presentation conditions (views). We selected components generalizing across subjects by sorting them based on the subject (**U**) loadings, and explain the first one here to concretize what the method can produce. The first component is active in the speech-related views, pure audio (A:Speech), and combined audio-visual (AV:Speech) views, indicating that it captures speech-related activity. A closer look at the $\mathbf{W}^{(m)}$ loadings for the views shows activation of the same auditory regions of the brain, demonstrating the signal is neuroscientifically relevant.

Quantitatively, BMTF fits the data better than simpler alternatives as demonstrated by the missing value prediction in Table 2, while in comparison to the analysis of [13], it extracts more components having consistent behaviour over the subjects, indicating that taking the tensorial nature of data into account improves detection of structure.

5 Discussion

We introduced a novel multi-view tensor factorization problem, of collectively decomposing multiple paired tensors into factors. We factorize the tensors into

PARAFAC-type (equivalently, CP-type) components, each shared by a subset of the tensors, from one to all. We introduced a Bayesian multi-view tensor factorization (BMTF) model that solves the problem via a joint CP-type decomposition of tensors while learning the precise type and number of factors automatically. In the special case of two tensors, our method is simultaneously also the first Bayesian tensor canonical correlation analysis (CCA) method. The model can also be considered as an extension of the matrix-based Group Factor Analysis method [13] to tensors.

We validated the model's performance in identifying components on simulated data. The model was then demonstrated on a new toxicogenomics problem and a neuroimaging dataset, yielding interpretable findings with detailed interpretations on-going. Initial evidence suggests that taking the tensor nature of data into account makes the results more accurate and precise. In particular, the model is able to handle degenerate solutions well, making the formulation applicable to a wider set of datasets.

Acknowledgments. The work was supported by the Academy of Finland (140057; Finnish Centre of Excellence COIN, 251170) and the FICS doctoral programme. We also acknowledge Aalto Science-IT resources.

References

1. Kolda, T., Bader, B.: Tensor decompositions and applications. SIAM Review **51**(3) (2009) 455–500
2. Carroll, J.D., Chang, J.J.: Analysis of individual differences in multidimensional scaling via an n-way generalization of Eckart-Young decomposition. Psychometrika **35**(3) (1970) 283–319
3. Harshman, R.A.: Foundations of the parafac procedure: models and conditions for an explanatory multimodal factor analysis. UCLA Working Papers in Phonetics **16** (1970) 1–84
4. Tucker, L.R.: Some mathematical notes on three-mode factor analysis. Psychometrika **31**(3) (1966) 279–311
5. Xiong, L., Chen, X., Huang, T.K., Schneider, J., Carbonell, J.G.: Temporal collaborative filtering with bayesian probabilistic tensor factorization. In: Proceedings of SIAM Data Mining. Volume 10. (2010) 211–222
6. Chu, W., Ghahramani, Z.: Probabilistic models for incomplete multi-dimensional arrays. In: Proceedings of AISTATS, JMLR W&CP. Volume 5. (2009) 89 – 96
7. Lee, S.H., Choi, S.: Two-dimensional canonical correlation analysis. Signal Processing Letters, IEEE **14**(10) (2007) 735–738
8. Kim, T.K., Cipolla, R.: Canonical correlation analysis of video volume tensors for action categorization and detection. IEEE Transactions on Pattern Analysis and Machine Intelligence **31**(8) (2009) 1415–1428
9. Yan, J., Zheng, W., Zhou, X., Zhao, Z.: Sparse 2-d canonical correlation analysis via low rank matrix approximation for feature extraction. IEEE Signal Processing Letters **19**(1) (2012) 51–54
10. Lu, H.: Learning canonical correlations of paired tensor sets via tensor-to-vector projection. In: Proceedings of IJCAI. (2013) 1516–1522

11. Hardoon, D.R., Szedmak, S., Shawe-Taylor, J.: Canonical correlation analysis: An overview with application to learning methods. Neural Computation **16**(12) (2004) 2639–2664
12. Klami, A., Virtanen, S., Kaski, S.: Bayesian canonical correlation analysis. Journal of Machine Learning Research **14** (2013) 965–1003
13. Virtanen, S., Klami, A., Khan, S.A., Kaski, S.: Bayesian group factor analysis. In: Proceedings of AISTATS, JMLR W&CP 22. (2012) 1269–1277
14. Cattell, R.B.: Parallel proportional profiles and other principles for determining the choice of factors by rotation. Psychometrika **9**(4) (1944) 267–283
15. Kruskal, J.B.: Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. Linear Algebra and its Applications **18**(2) (1977) 95 – 138
16. Neal, R.M.: Bayesian learning for neural networks. Springer-Verlag (1996)
17. Mitchell, T.J., Beauchamp, J.J.: Bayesian variable selection in linear regression. Journal of the American Statistical Association **83**(404) (1988) 1023–1032
18. Krijnen, W., Dijkstra, T., Stegeman, A.: On the non-existence of optimal solutions and the occurrence of degeneracy in the candecomp/parafac model. Psychometrika **73**(3) (2008) 431–439
19. Srensen, M., Lathauwer, L., Comon, P., Icart, S., Deneire, L.: Canonical polyadic decomposition with a columnwise orthonormal factor matrix. SIAM Journal on Matrix Analysis and Applications **33**(4) (2012) 1190–1213
20. Mørup, M., Hansen, L.K.: Automatic relevance determination for multiway models. Journal of Chemometrics **23**(7-8) (2009) 352 – 363
21. Hoff, P.D.: Hierarchical multilinear models for multiway data. Computational Statistics & Data Analysis **55**(1) (2011) 530 – 543
22. Hayashi, K., Takenouchi, T., Shibata, T., Kamiya, Y., Kato, D., Kunieda, K., Yamada, K., Ikeda, K.: Exponential family tensor factorization: an online extension and applications. Knowledge and Information Systems **33**(1) (2012) 57–88
23. Xu, Z., Yan, F., Qi, A.: Infinite tucker decomposition: Nonparametric bayesian models for multiway data analysis. In: Proceedings of ICML. (2012) 1023–1030
24. Acar, E., Rasmussen, M.A., Savorani, F., Naes, T., Bro, R.: Understanding data fusion within the framework of coupled matrix and tensor factorizations. Chemometrics and Intelligent Laboratory Systems **129** (2013) 53 – 63
25. Ermis, B., Acar, E., Cemgil, A.T.: Link prediction in heterogeneous data via generalized coupled tensor factorization. Data Mining and Knowledge Discovery (2013) 1–34
26. Yilmaz, K.Y., Cemgil, A.T., Simsekli, U.: Generalised coupled tensor factorisation. In: Proceedings of NIPS. (2011) 2151–2159
27. Andersson, C.A., Bro, R.: The N-way toolbox for MATLAB. Chemometrics and Intelligent Laboratory Systems **52**(1) (2000) 1 – 4
28. Lundy, M.E., Harshman, R.A., Kruskal, J.B.: A two-stage procedure incorporating good features of both trilinear and quadrilinear models. Multiway data analysis (1989) 123–130
29. Hartung, T., Vliet, E.V., Jaworska, J., Bonilla, L., Skinner, N., Thomas, R.: Food for thought ... systems toxicology. ALTEX **29**(2) (2012) 119–128
30. Lamb, J., et al.: The connectivity map: Using gene-expression signatures to connect small molecules, genes, and disease. Science **313**(5795) (2006) 1929–1935
31. Shoemaker, R.H.: The nci60 human tumour cell line anticancer drug screen. Nature Reviews Cancer **6**(10) (2006) 813–823
32. Kamal, A., et al.: A high-affinity conformation of HSP90 confers tumour selectivity on HSP90 inhibitors. Nature **425**(6956) (2003) 407–410