Unsupervised Machine Learning: Probabilistic Factor Analysis

Tensor Factorisation

Suleiman Ali Khan
University of Helsinki, Finland

suleiman.khan@helsinki.fi

May 18, 2017
Overview

1. **Tensor Factorisation**
   - Introduction to Tensors
   - CP - Canonical Decomposition
   - Tucker Model Family

2. **Multi-tensor Factorizations**
   - Multi-view tensor Factorisation
   - Generalised Multi-tensor Factorisation
What are Tensors?

- **Vectors** are 1st order tensors: $x_i$ (one mode)
- **Matrices** are 2nd order tensors: $X_{ij}$ (two modes)
- Tensors of order $\geq 3$ are called **higher-order tensors** $X_{i,j,q,r,s}$

In the following we will only use third-order tensors, i.e. “cubes”
Tensors in use 1/5

Figure: Kao et al. ISI 2013, modelled time evolution of discussion issues on twitter.
Tensors in use 1/5

Figure: “pepper”, “spray”, “pregnant”, “woman”, “miscarries”, “police” led to uproar on “occupy” “seattle”
Integrating (biological) responses from different studies.

Figure: Li et. al. Plos CB 2011, discovered meaningful biological modules and protein complex networks using Tensor Factorizations.
Tensors in use 2/5

- Exploring factors of gene expression patterns over replicates of several stimuli.
  - Yener et. al. BMC Systems Biology, 2008
- Time series of several biological data.
  - Li and Ngom, BIBM 2010
- Integrative analyses of metabolic and gene expression networks
  - Jensen et. al. PloS one, 2013
Relational data is a collection of relationships among multiple objects.

Figure: Hayashi et. al. KIS 2012 study: Temporal sequence of measurements from various distributed sensors: sensor types × locations × time.
Distributed sensors:

- Environmental modelling for sensitive wildlife and habitats.
  - Temperature, Humidity, Light, at different locations and time points. Sun et. al., 2007

- Traffic conditions in tunnels for anomaly detection.
  - Temperature, Light, Vehicles, at different tunnels and time points.
Tensors in use 4/5

Extensively popular in Neuroinformatics and Neuroimaging!

Figure: Zhou et. al. 2014 study: Brain regions linked with ADHD diagnosis status
Tensors in use 5/5

- Community detection
  - Anandkumar, Ge, Hsu, and S. Kakade 2013
- Parsing
  - Cohen, Satta, and Collins 2013
- Knowledge base completion
  - Chang et al. 2014, Singh, Rocktaschel, and Riedel 2015
- Topic modelling
- Crowdsourcing
  - Zhang et al. 2014
- Mixture models
  - Anandkumar, Ge, Hsu, S. M. Kakade, et al. 2013
- Psychometrics and Chemometrics

Figure: uses multidimensional data arrays (tensors).
What are Tensors?

Formally, tensors are defined as

An $n^{th}$ order tensor is an element of the tensor product of $n$ vector spaces, i.e.

$$\mathbf{X} \in \mathbf{V}^{(1)} \otimes \mathbf{V}^{(2)} \otimes ... \otimes \mathbf{V}^{(n)}$$

- Less formally: tensors are multidimensional arrays, i.e. data structures with (possibly) more than two indices ($X_{i,j,q,r,s}$)
- The order of an tensor is the number of modes (i.e, number of indices needed to identify elements in the array)
Example of a Third-Order Tensor

\[ \mathbf{T} \in \mathbb{R}^{10 \times 7 \times 5} \]

Mode-1 has dimension \( I \)

Mode-2 has dimension \( J \)

Mode-3 has dimension \( K \)

\[ \begin{array}{c}
\text{Block } = \mathbf{T}_{1,1,1} \\
\text{Block } = \mathbf{T}_{3,7,2} \\
\text{Block } = \mathbf{T}_{1,5,4}
\end{array} \]
**Notations**

**Slices** are two-dimensional sections of a tensor (i.e. matrices)

![Slices of a third-order tensor](image)

**Figure**: Slices of a third-order tensor.
**Fibers** are higher-order analogues of rows and columns in matrices.

![Diagram of tensor fibers](image)

**Figure**: Fibres of a third-order tensor.
Notations

Rank-one tensors

An $N$-way tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ is rank one if it can be written as the outer product of $N$ vectors, i.e.,

$$\mathcal{X} = \mathbf{x}^{(1)} \circ \mathbf{x}^{(2)} \circ \ldots \circ \mathbf{x}^{(N)}$$

Figure: Rank-one third-order tensor.
A tensor $\mathcal{X} \in \mathcal{R}^{l_1 \times l_2 \times \ldots \times l_N}$ is diagonal if $x_{i_1,i_2,\ldots,i_N} \neq 0$ only if $i_1 = i_2 = \ldots = i_N$. We use $\mathcal{I}$ to denote the *identity tensor* with ones on the superdiagonal and zeros elsewhere.

**Figure**: Three-way identity tensor: $\mathcal{I} \in \mathcal{R}^{I \times I \times I}$
Tensor Products

The **Kronecker product** of matrices $A \in \mathcal{R}^{I \times J}$ and $B \in \mathcal{R}^{K \times L}$ is denoted by $A \otimes B$. The result is a matrix of size $(IK) \times (JL)$ and is defined by:

$$A \otimes B = \begin{bmatrix}
a_{11}B & a_{12}B & \ldots & a_{1J}B \\
a_{21}B & a_{22}B & \ldots & a_{2J}B \\
\vdots & \vdots & \ddots & \vdots \\
a_{I1}B & a_{I2}B & \ldots & a_{IJ}B
\end{bmatrix}$$
The **Khatri-Rao product** is the “matching columnise” Kronecker product. Given matrices $\mathbf{A} \in \mathcal{R}^{I \times K}$ and $\mathbf{B} \in \mathcal{R}^{J \times K}$, their Khatri-Rao product is denoted by $\mathbf{A} \odot \mathbf{B}$. The result is a matrix of size $(IJ) \times (K)$ and is defined by:

$$\mathbf{A} \odot \mathbf{B} = \begin{bmatrix} a_1 \otimes b_1 & a_2 \otimes b_2 & \cdots & a_K \otimes b_K \end{bmatrix}$$
The **mode-n (matrix) product** of a tensor \( \mathcal{X} \in \mathcal{R}^{I_1 \times I_2 \times \ldots \times I_N} \) with matrix \( \mathbf{A} \in \mathcal{R}^{J \times I_n} \) is denoted by \( \mathcal{X} \times_n \mathbf{A} \) and is of the size \( \mathcal{R}^{I_1 \times \ldots \times I_{n-1} \times J \times I_{n+1} \times \ldots \times I_N} \); and is defined as:

\[
(\mathcal{X} \times_n \mathbf{A})_{i_1 \ldots i_{n-1} j_{n+1} \ldots i_N} = \sum_{i_n=1}^{I_n} x_{i_1 i_2 \ldots i_N} a_{j_{n}}
\]

i.e. each mode-\( n \) fibre is multiplied by the matrix \( \mathbf{A} \).
Activity 2

Describe or Draw:

- 2-way tensor
- rank-3 tensor
- column fibres and lateral slices of a 3rd-order tensor
- a identity tensor of 3-modes
- Khatri-Rao $\odot$ product of two order-2 tensors.

Write your student number.
Issues with Tensor representations!

Tensor representation is generally high-dimensional and large-scale

- e.g. 1,000 users 1,000 movies 365 days
  = total 365,000,000 relationships/data points

- For Netflix: 29,000,000,000,000 relationships

Dimensional reduction techniques such as tensor factorization have to be used.
Why Tensor Factorizations?

- When data sets have more dimensions than just samples and features, they may present *tensorial relationships/structure* that can not be captured by matrix methods.

- In order to capture the more structured patterns of such data sets and *avoid overfitting*, tensor methods use more constrained formulations that have *fewer parameters* than their matrix counterparts.
History?

- 1904: FA (Spearman)
- 1936: CCA (Hotelling)
- 1956: Varimax (Keiser)
- 1970: Tensor Factorisations (Carroll and Chang70 - Harshman70)
Parallel Proportional Profiles

Cattell 1944, stated that two independent realizations of similar data sets could be decomposed jointly with a simultaneous factor analysis. This process learns a common projection matrix that differs only in the scale of factors for the two data sets and captures the intrinsic axis of the underlying factors.
C - Canonical Decomposition [CANDECOMP, Carroll & Chang ’70]
P - Parallel Factor Analysis [PARAFAC, Harshmen ’70]

Parallel factor analysis extends Cattell’s conceptualization to a tensor (a set of matrices placed together), allowing the projections of FA to differ only in their scales in the third mode. CP can therefore be seen as multiple factor analysis of a single phenomenon being performed simultaneously, to infer the empirically meaningful factors.
CP - CANDECOMP/PARAFAC

Figure: CANDECOMP/PARAFAC (CP) factorizes a tensor $\mathcal{X}$ into $K$ components in each mode and is equivalent to sum of rank one component tensors.
The CP decomposition is defined in a symmetric way to factorize a tensor into a sum of rank-one tensors, where each rank one tensor is the outer product of vector loadings in all modes. For a third-order tensor $X \in \mathbb{R}^{N \times D \times L}$, a rank-$K$ CP is represented as:

$$X = \sum_{k=1}^{K} z_k \odot w_k \odot u_k + \epsilon = (Z \odot U) \times_2 W + \epsilon,$$

where $Z \in \mathbb{R}^{N \times K}$ and $U \in \mathbb{R}^{L \times K}$ and $W \in \mathbb{R}^{D \times K}$ are the latent variables corresponding to the three modes.
CP - Rank

\[ \text{O}(!) \]

\[ 1 \leq \text{Rank} \leq \min(NL, ND, DL)! \] Iterative solutions are too slow.
$O(!!)$

$1 \leq \text{Rank} \leq \min(NL, ND, DL)!$ Iterative solutions are too slow.

Rank-$k$ CP solution is not guaranteed to be the best rank-$k$ approximation.

- if the true number of factors is larger than $k \rightarrow$ different factors
- if $k$ is set larger than the actual value $\rightarrow$ artificial splits
CP - Rank

$O(!!)$

$1 \leq \text{Rank} \leq \min(NL,ND,DL)!$ Iterative solutions are too slow.

Rank-k CP solution is not guaranteed to be the best rank-k approximation.

- if the true number of factors is larger than $k \rightarrow$ different factors
- if $k$ is set larger than the actual value $\rightarrow$ artificial splits

Best rank-k approximation may not even exist 😊.
Practical application of CP can occasionally suffer from degenerate solutions.

**Degeneracy** is:
- two or more components become highly (inversely) correlated in all the modes; *(linear dependency)* and
- some of the loading values becoming arbitrarily large *(infinite length)*

Occurs when:
- best rank-k approximation does not exist, as data comes from a non-trilinear structure.
CP - Degeneracy

\[ Y^{(s)} = a_s \circ b_s \circ c_s \quad \quad Y^{(t)} = a_t \circ b_t \circ c_t \]

\[ \text{Vec}(Y^{(s)}) \quad \quad \text{Vec}(Y^{(t)}) \]

\[ Y^{(s)} + Y^{(t)} \text{ remains “small” and contributes to a better model fit} \]

**Figure**: Two diverging components. Solution: Orthogonality (Krijnen 2008) or Non-negativity (Lim & Comon 2009).
Bayesian CP

\[ x_{n,l,d} \sim \mathcal{N}(z_n^T (w_d \ast u_l), \tau^{-1}) \]
\[ z_{n,k} \sim \mathcal{N}(0, 1) \]
\[ u_{l,k} \sim \mathcal{N}(0, 1) \]
\[ w_{d,k} \sim h_k \mathcal{N}(0, (\alpha_{d,k})^{-1}) + (1 - h_k)\delta_0 \]
\[ h_k \sim Bernoulli(\pi_k) \]
\[ \pi_k \sim Beta(a^{\pi}, b^{\pi}) \]
\[ \alpha_{d,k} \sim Gamma(a^{\alpha}, b^{\alpha}) \]
\[ \tau \sim Gamma(a^\tau, b^\tau) \]

Khan et. al. 2014/2016/2017
### Bayesian CP

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Amino Acid</th>
<th>Flow Injection</th>
<th>Kojima Girls</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>5 x 201 x 61</td>
<td>12 x 50 x 45</td>
<td>4 x 153 x 20</td>
</tr>
<tr>
<td>Factors</td>
<td>3</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>pf-test</td>
<td>3.0 ± 0.0</td>
<td>4.5 ± 0.5</td>
<td>2.0 ± 0.1</td>
</tr>
<tr>
<td>Bayesian CP</td>
<td>3.0 ± 0.0</td>
<td>4.5 ± 0.5</td>
<td>2.0 ± 0.1</td>
</tr>
<tr>
<td>ARDCP</td>
<td>3.1 ± 0.3</td>
<td>4.0 ± 0.0</td>
<td>1.2 ± 0.4</td>
</tr>
<tr>
<td>Prediction RMSE</td>
<td>0.0257 ± 0.0003</td>
<td><strong>0.045 ± 0.010</strong></td>
<td><strong>0.189 ± 0.025</strong></td>
</tr>
<tr>
<td>ARDCP</td>
<td><strong>0.0278 ± 0.0035</strong></td>
<td>0.065 ± 0.001</td>
<td><strong>0.305 ± 0.051</strong></td>
</tr>
<tr>
<td>CP</td>
<td><strong>0.0256 ± 0.0003</strong></td>
<td><strong>0.053 ± 0.001</strong></td>
<td>1.643 ± 4.098</td>
</tr>
<tr>
<td>pTucker</td>
<td><strong>0.0250 ± 0.0003</strong></td>
<td>0.049 ± 0.001</td>
<td>0.236 ± 0.055</td>
</tr>
</tbody>
</table>

Kojima Girls is a benchmark datasets containing degenerate components.
CP - Scaling

- Matrices: sample mean 0, variance 1
- Tensors: ?
The Tucker model family defines several levels of factorisation, and has three main forms, Tucker 1, 2, 3.

**Tucker-1**

The 1-mode factorization of Tucker, is the most relaxed formulation that decomposes only one of the modes $\sim$ matrix factorisation of a matricized tensor.
Tucker-2 and Tucker-3 factorize two and all three of the modes, respectively.

Tucker-3 enforces more structure and is characterized by interactions between a different set of factors in each mode.

- Unlike in CP, in the Tucker-3 model, a factor does not represent an additive source of information.
- Rather, it represents a pattern of variation in a given mode only.
- The factors are thought to have generated the data by interacting with several other patterns of variation (factors).
Figure: Tucker 3-mode decomposition factorizes the tensors in three different modes. The interactions are modelled between components via a core tensor $\mathcal{G}$. 
Tucker-3

3rd-order data tensor

\[ X \approx \sum_{p=1}^{K_1} \sum_{q=1}^{K_2} \sum_{r=1}^{K_3} g_{p,q,r} z_p \circ w_q \circ u_r. \]
Issues with Tucker-3

- The complex interaction of factors via $G$ makes interpretation of factors difficult.
- Moreover, in contrast to the CP factorization, the Tucker model is not guaranteed to provide unique solutions.
- The Tucker component matrices can be rotated, and the core tensor $G$ counter-rotated to obtain infinite number of models with an equal fit.
Probabilistic Tucker Model

Component-wise sparsity using a Laplace prior:

\[ P(\mathcal{X}|G, Z, W, U, \sigma) \sim \mathcal{N}(G \times_1 Z \times_2 W \times_3 U, \sigma^2) \]

\[ P_{\text{Laplace}}(G|\alpha^G) = \left( \frac{\alpha^G}{2} \right)^{K_1 K_2 K_3} \exp[-\alpha^G] \]

\[ P_{\text{Laplace}}(Z|\alpha^Z) = \prod_k \left( \frac{\alpha_k^Z}{2} \right)^N \exp[-\alpha_k^Z|Z_k|_1] \cdot \]

Mørup & Hansen 2009.
Probabilistic Tucker Model

Component-wise sparsity using a Laplace prior yields model selection as well as sparsity in the core matrix.

Figure: Learned sparse $G$ for several datasets.
Activity 1

**Problem:** In a brain-activity measurement experiment, 5 pupils were made to hear a song of 1 minute. The brain-activity measurements of each of the pupils were then recorded at an interval of 1 second. The brain-activity measurement can be assumed to be represented by values on x-y 2D-plane. The measurements are normally distributed with higher values indicating higher brain-activity.

**Task:** Your task is to design an unsupervised machine learning model that can identify the relationships between the song (or parts of it) and the brain activities (or parts of it). Describe the Pros/Cons of your model. You can invent a new model or use the learned concepts.
Extending Factorisation to multiple tensors

Given a collection of $M$ paired tensors (views):

$$\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \ldots, \mathbf{X}^{(M)} \in \mathbb{R}^{N \times D_m \times L}$$

Each tensor should factorize into

- a view-specific matrix of loadings $\mathbf{W}^{(m)} \in \mathbb{R}^{D_m \times K}$ and
- a low($K$)-dimensional tensor $\mathbf{Y} \in \mathbb{R}^{N \times K \times L}$ common for all views:

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

Noise tensor: $\mathbf{\epsilon}^{(m)} \in \mathbb{R}^{N \times D_m \times L}$

Notations: $\mathbf{Y}$ (Tensor), $\mathbf{Y}$ (Matrix), $\times_2$: Mode2 product, analogous to matrix product.
Extending Factorisation to multiple tensors

Tensor $Y$ forms the shared latent tensor.

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

For a joint CP-type factorization, MTF becomes:

$$X^{(m)} = \sum_{k=1}^{K} z_k \odot u_k \odot w_k^{(m)} + \epsilon^{(m)}$$

$$= (Z \odot U) \times_2 W^{(m)} + \epsilon^{(m)}.$$  

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

Notations, $\odot$: KhatriRao product, i.e. outer product of columns.
Extending Factorisation to multiple tensors
Multi-view Tensor Factorization

The combined factorization should be capable of decomposing the tensors into factors shared between all, some, or a single tensor.

- **White**: The loadings $W_k^{(m)}$ are zero for the components $k$ that are not active in view $m$.
- **Black**: Non-zero.

$$\mathbf{U} \approx \mathbf{X}(1) \mathbf{W}(1) + \mathbf{X}(2) \mathbf{W}(2) + \mathbf{X}(3) \mathbf{W}(3)$$
Graphical Model

\[ x^{(m)} \sim \mathcal{N}\left((Z \odot U) \times_2 W^{(m)}, I(\tau^{(m)})^{-1}\right) \]
BMTF: Two layers of sparsity

**View-wise Sparsity**

Controlled by $h_{k}^{(m)}$ over all view$(m)$-component$(k)$ pairs $W_{:,k}^{(m)}$, 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.

**Feature-wise Sparsity**

Controlled by $\alpha_{d,k}^{(m)}$, sparsity is induced across all the $D_M$ features in each $W_{d,k}^{(m)}$. Regularizes the solution to handle for (weak) degeneracies without orthognality constraints.
BMTF Model Specification

\[ x_{n,l,d}^{(m)} \sim \mathcal{N}\left((z_n * u_l)\,^T\, w_d^{(m)}\right), \left(\tau^{(m)}\right)^{-1} \]

\[ z_{n,k} \sim \mathcal{N}(0, 1) \]

\[ u_{l,k} \sim \mathcal{N}(0, (\beta_{l,k})^{-1}) \]

\[ w_d^{(m)} \sim h_k^{(m)} \mathcal{N}(0, (\alpha_{d,k}^{(m)})^{-1}) + (1 - h_k^{(m)})\delta_0 \]

\[ h_k^{(m)} \sim \text{Bernoulli}(\pi_k) \]

\[ \pi_k \sim \text{Beta}(a^\pi, b^\pi) \]

\[ \beta_{l,k} \sim \text{Gamma}(a^\beta, b^\beta) \]

\[ \alpha_{d,k}^{(m)} \sim \text{Gamma}(a^\alpha, b^\alpha) \]

\[ \tau^{(m)} \sim \text{Gamma}(a^\tau, b^\tau) \]
Several methods fall out as special cases of BMTF. Which model does BMTF become:

<table>
<thead>
<tr>
<th>For $M = 1$</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>For $M = 2$</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>For $L = 1$</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>For $L = 1$ &amp; $M = 2$</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Assignment 1C: Quiz

Several methods fall out as special cases of BMTF. Which model does BMTF become:

For $M = 1$?
The model reduces to sparse Bayesian CP factorization / Bayesian Tensor Principal Component Analysis (PCA), which can automatically infer the number of components.

For $M = 2$?

For $L = 1$?

For $L = 1$ & $M = 2$?
Several methods fall out as special cases of BMTF. Which model does BMTF become:

| $M = 1$ | The model reduces to sparse Bayesian CP factorization / Bayesian Tensor Principal Component Analysis (PCA), which can automatically infer the number of components. |
| $M = 2$ | Bayesian Tensor Canonical Correlation Analysis (CCA). |
| $L = 1$ | Bayesian Group Factor Analysis |
| $L = 1 \& M = 2$ | Bayesian Canonical Correlation Analysis |
Two tensors $A \in \mathbb{R}^{30 \times 40 \times 20}$ and $B \in \mathbb{R}^{30 \times 50 \times 20}$, composed of one shared and two view-specific components.

**Left:** True loadings are drawn for the three components.

**Right** BMTF run with $K = 4$. 
Case Study: Toxicogenomics

The key question BMTF can answer is, which patterns are specific to individual types of cancer and which occur across cancers, and which of them are related to drugs effectiveness.
Toxicogenomics: Comp 1 shows well-known HSP response
Case Study: Functional Neuroimaging

Multi-view $M = 7$ fMRI views from subjects exposed to multiple audiovisual stimuli.

- Each view is:
  - Time Points $\times$
  - Brain Region $\times$
  - Subjects
- Goal: Identify responses that generalize across subjects and describe relationships of different “presentation” conditions.
Important Points

- automatic Rank determination and solution for degeneracies
- basic tool for integrative tensor analysis.
Generalised Multi-tensor Factorisation

Figure: Khan et. al. Machine Learning 2016
Generalised Multi-tensor Factorisation

Activity 4: Design the model! - Either make Plate Diagram or write generative model/distributions 😊

Figure: Khan et. al. Machine Learning 2016
relaxed CP-Tucker Factorisation

Figure: An illustration of joint factorization of a matrix $\mathcal{X}^{(1)}$ and a tensor $\mathcal{X}^{(2)}$. The generation of the loading matrices corresponding to the tensor, $\mathcal{W}^{(2)}$, is shown in more detail on the right ($D_2 = 5$ and $K = 2$). MTF does a trilinear CP decomposition, as shown by the $\mathcal{W}^{(2)}$ slabs that equal $\mathbf{V}$ with just a scale difference. rMTF allows deviation from this, as illustrated by the slight changes in the $\mathcal{W}^{(2)}$ patterns.
relaxed CP-Tucker Factorisation

\[
\begin{align*}
    x_{n,d,l} & \sim \mathcal{N}\left( z_n^T w_{l,d}, \tau_l^{-1} \right) \\
    w_{l,d,k} & \sim h_{l,k} \mathcal{N}\left( u_{l,k} v_{d,k}, (\alpha_{d,k})^{-1} \right) + (1 - h_{l,k}) \delta_0 \\
    z_{n,k}, u_{l,k}, v_{d,k} & \sim \mathcal{N}(0, 1) \\
    \alpha_{d,k} & \sim \text{Gamma}(a^\lambda, b^\lambda) \\
    h_{l,k} & \sim \text{Bernoulli}(\pi_k) \\
    \pi_k & \sim \text{Beta}(a^\pi, b^\pi) \\
    \tau_{t,l} & \sim \text{Gamma}(a^\tau, b^\tau) .
\end{align*}
\]
Formally, for $m$ data sets collected into the tensor $\hat{X}$ the model is

$$\hat{X}_{:,i,:}/l \sim W_{:,i,:}/W_{:,i,:}/l,$$

where block-structure $\{g_1,g_2,g_3,g_4,g_5\}$ is imposed by the binary variable $h_{b,k,l}$ for $k \in 1 \ldots K$, $l \in 1 \ldots L$ via a spike and slab prior. The relaxed formulation is embedded by assuming that the $\sum L_i$ slabs of $\mathcal{W}$ are drawn from the mean matrix $V \in \mathbb{R}^{\sum D_i \times K}$ as

$$w_{d,k,l} \sim \begin{cases} h_{b_d,k,l}v_{d,k} + (1 - h_{b_d,k,l})\delta_0, & \text{if } l \text{ is a matrix-slab}, \\ h_{b_d,k,l}\mathcal{N}(v_{d,k}u_{l,k}, \lambda^{-1}) + (1 - h_{b_d,k,l})\delta_0, & \text{otherwise}. \end{cases}$$

$$v_{d,k} \sim \mathcal{N}(0, (\alpha_{d,k})^{-1})$$

$$\alpha_{d,k} \sim \text{Gamma}(a^\alpha, b^\alpha),$$

where $b_d$ denotes which group feature $d$ belongs to and the other priors remain unchanged.
Generalised Multi-tensor Factorisation

Figure: Tensor collections with arbitrary pairing between the modes can be assembled into a single large sparse tensor. If the first and the second mode contain groups \( \{g_1, \ldots, g_G\} \) and the third mode groups \( \{f_1, \ldots, f_F\} \), the whole data collection forms a \( \sum_{i=1}^{G} |g_i| \times \sum_{i=1}^{G} |g_i| \times \sum_{j=1}^{F} |f_j| \) tensor.
Acknowledgements

- Kolda and Badar, SIAM, 2009
- Maximilian Nickel, (University of Munich, slides, www.cip.ifi.lmu.de/~nickel/iswc2012-slides)