Sparse Coding and Automatic Relevance Determination for Multi-way models

Morten Mørup and Lars Kai Hansen
DTU Informatics
Technical University of Denmark
2800 Kgs. Lyngby
Email: {mm.lkh}@imm.dtu.dk
Telephone: +45 4525 3900
Fax: +45 4587 2599

***Abstract*—Multi-way modeling has become an important tool in the analysis of large scale multi-modal data. An important class of multi-way models is given by the Tucker model which decomposes the data into components pertaining to each modality as well as a core array indicating how the components of the various modalities interact. Unfortunately, the Tucker model is not unique. Furthermore, establishing the adequate model order is difficult as the number of components are specified for each mode separately. Previously, rotation criteria such as VARIMAX has been used to resolve the non-uniqueness of the Tucker representation [7]. Furthermore, all potential models have been exhaustively evaluated to estimate the adequate number of components of each mode. We demonstrate how sparse coding can prune excess components and resolve the non-uniqueness of the Tucker model while Automatic Relevance Determination in Bayesian learning form a framework to learn the adequate degree of sparsity imposed. On a wide range of multi-way data sets the proposed method is demonstrated to successfully prune excess components thereby establishing the model order. Furthermore, the non-uniqueness of the Tucker model is resolved since among potential models the models giving the sparsest representation as measured by the sparse coding regularization is attained. The approach readily generalizes to regular sparse coding as well as the CandeComp/PARAFAC model as both models are special cases of the Tucker model.**

I. Introduction

Tensor decompositions are in frequent use today in a variety of fields including psychometric, image analysis, web data mining, bio-informatics, neuroimaging and signal processing [8]. While the analysis of tensors have been somewhat restricted due to memory and computational limitations growing attention has lately been given to the analysis of data with tensorial structure fueled by the increased memory capacity and computational power of modern computers. Tensors, i.e., $\mathcal{X} \in \Re^{I_1 \times I_2 \times \ldots \times I_N}$, also called multi-way arrays, multidimensional matrices or hypermatrices are generalizations of vectors (first order tensors) and matrices (second order tensors). The two most commonly used decompositions of tensors are the Tucker model [20] and the more restricted CandeComp/PARAFAC (CP) model [5].

The Tucker model reads

$$\begin{aligned}\mathcal{X}_{i_1,i_2,\ldots,i_N} &= \mathcal{R}_{i_1,i_2,\ldots,i_N} + \mathcal{E}_{i_1,i_2,\ldots,i_N} \\ &= \textstyle\sum_{j_1 j_2 \ldots j_N} \mathcal{G}_{j_1,j_2,\ldots,j_N} a^{(1)}_{i_1,j_1} a^{(2)}_{i_2,j_2} \cdot \ldots \cdot a^{(N)}_{i_N,j_N} + \mathcal{E}_{i_1,i_2,\ldots,i_N}.\end{aligned}$$

where $\mathcal{G} \in \mathbb{R}^{J_1 \times J_2 \times \ldots \times J_N}$ and $\boldsymbol{A}^{(n)} \in \mathbb{R}^{I_n \times J_n}$ while $\mathcal{E}$ is the approximation error. To indicate how many vectors pertain to each modality it is customary also to denote the model a Tucker$(J_1, J_2, \ldots, J_N)$. Using the n-mode matrizicing operation [8] the model can also be expressed as

$$\boldsymbol{X}_{(n)} \quad = \quad \boldsymbol{A}^{(n)} \boldsymbol{Z}_{(n)} + \boldsymbol{E}_{(n)},$$

where

$$\boldsymbol{Z}_{(n)} = \boldsymbol{G}_{(n)} (\boldsymbol{A}^{(1)} \otimes \ldots \otimes \boldsymbol{A}^{(n-1)} \otimes \boldsymbol{A}^{(n+1)} \otimes \ldots \otimes \boldsymbol{A}^{(N)})^\top.$$

Furthermore, using the n-mode tensor product $\times_n$ [10] given by

$$(\mathcal{Q} \times_n \mathbf{P})_{i_1,i_2,\ldots,j_n,\ldots i_N} = \sum_{i_n} \mathcal{Q}_{i_1,i_2,\ldots,i_n,\ldots i_N} \mathbf{P}_{j_n,i_n},$$

the model is stated as

$$\mathcal{X} = \mathcal{G} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \times_3 \ldots \times_N \mathbf{A}^{(N)}.$$

Thus, the Tucker model represents the data spanning the n^{th} modality by the vectors (loadings) given by the J_n columns of $\boldsymbol{A}^{(n)}$ such that the vectors of each modality interact with the vectors of all remaining modalities with strengths given by a so-called core tensor $\mathcal{G}$. As a result, the Tucker model encompasses all possible linear interactions between vectors pertaining to the various modalities of the data. The CP model is a special case of the Tucker model where the size of each modality of the core array $\mathcal{G}$ is the same, i.e., $J_1 = J_2 = \cdots = J_N$ while interaction is only between columns of same indices such that the only non-zero elements are found along the diagonal of the core, i.e., $\mathcal{G}_{j_1,j_2,\ldots,j_N} \neq 0$ iff $j_1 = j_2 = \ldots = j_N$. Thus, the CP model can by appropriate scaling of each component be expressed as a Tucker model with unit diagonal core, i.e. $\mathcal{G}_{CP} = \mathcal{I}$. As such, the regular sparse coding model based on a factor analysis type representation can be formulated as a 2-way Tucker model with diagonal core. Notice, in the Tucker model a rotation of a given loading matrix $\boldsymbol{A}^{(n)}$ can be compensated by a counter rotation of the core $\mathcal{G}$ [8]. For the CP model it is not possible in general to rotate the loadings and still keep the core diagonal. Thus, the CP model is unique up to scale and permutation [9].

As the CP model corresponds to the Tucker model with diagonal core – Tucker decompositions in which only some off diagonal elements are non-zero can be considered a representational interpolation between the Tucker and CP decomposition. Regularizing the core and loadings of the Tucker model by sparse priors using a sparse coding approach it becomes possible to simplify the core and turn of excess components. Furthermore, such restrictions on the core can potentially eliminate the rotation degeneracy of a Tucker decomposition. Thus, by adequately controlling the degree of pruning we can select the model order and simplify the core at the cost of estimating a conventional multi-way model.

We will use a standard approach in Bayesian inference referred to as Automatic Relevance Determination (ARD) [1], [17]. Traditionally, ARD has been based on Gaussian priors yielding a ridge regression type of selection. Here, we will derive an ARD approach based on the Laplace prior. Contrary to Gaussian priors, the Laplace prior favors sparse representations. Optimizing for sparse representation is related to the classic rotation criteria such as VARIMAX [6] and maximum Likelihood independent component analysis (ICA) based on sparse priors [13]. However, rather than rotating an estimated solution, the estimation process is directly posed as a tradeoff between simplicity of the representation and fitting the data. Thus, a sparse representation is strongly related to the principle of parsimony, i.e., among all possible accounts the simplest is considered the best [13]. If no formal prior information is given parsimony can be considered a reasonable guiding principle to avoid overfitting, see also [13] and references therein. In the present paper we describe the ARD approach based on sparse priors on the Tucker model. For a full analysis of the presented framework with comparison to existing model order heuristics as well as comparison between sparse and Gaussian priors see [**?**].

II. Sparse Coding and Automatic Relevance Determination for Multi-way models

Automatic Relevance Determination is a hierarchical Bayesian approach widely used for model selection [1], [17]. In ARD hyperparameters explicitly represents the relevance of different features by defining the range of variation for these features, usually by modeling the width of a zero-mean Gaussian prior imposed on the model parameters. If the width becomes zero, the corresponding feature cannot have any effect on the predictions. Hence, ARD optimizes these hyperparameters to discover which features are relevant. While ARD based on Gaussian priors can prune excess components Gaussian priors do not in general admit sparse representation within the active components hence does not necessarily favor simple parsimonious representations. However, the Laplace prior is known to admit sparse representation as it corresponds to a l_1 regularization thus is the closest convex proxy to minimizing for the number of non-zero elements in the model [4]. Therefore, we consider Laplace priors on the model parameter $\boldsymbol{\theta}_d$, i.e. $P_{Laplace}(\boldsymbol{\theta}_d|\alpha_d) = \prod_j \frac{\alpha_d}{2} exp[-\alpha_d|\theta_{j,d}|]$. In a Bayesian framework, the least squares objective $SSE = \|\mathcal{X} - \mathcal{R}\|_F^2 = \sum_{i_1,i_2,\ldots,i_n}(\mathcal{X}_{i_1,i_2,\ldots,i_n} - \mathcal{R}_{i_1,i_2,\ldots,i_n})^2$, corresponds to minimizing the negative log-likelihood assuming the entries in $\mathcal{X}$ are independent, identically distributed (i.i.d.) with Gaussian noise, i.e. $P(\mathcal{X}|\mathcal{R},\sigma^2) = (2\pi\sigma^2)^{-\frac{I_1 I_2 \cdots I_N}{2}} \exp[-\frac{\|\mathcal{X}-\mathcal{R}\|_F^2}{2\sigma^2}]$.

Assigning Laplace priors for the loadings and core the posterior likelihood can be written as

$$\begin{aligned} L &= P(\mathcal{G}, \boldsymbol{A}^{(1)}, \ldots, \boldsymbol{A}^{(N)}|\mathcal{X}, \sigma^2, \boldsymbol{\alpha}^{\mathcal{G}}, \boldsymbol{\alpha}^{(1)}, \ldots, \boldsymbol{\alpha}^{(n)}) \\ &\propto P(\mathcal{X}|\mathcal{R},\sigma^2)P(\mathcal{G}|\alpha^{\mathcal{G}})P(\boldsymbol{A}^{(1)}|\boldsymbol{\alpha}^{(1)}) \cdots P(\boldsymbol{A}^{N}|\boldsymbol{\alpha}^{(N)}). \end{aligned}$$

Thus the negative log likelihood based on Laplace priors is proportional to

$$-\log L \propto \mathrm{c} + \frac{1}{2\sigma^2}\|\mathcal{X} - \mathcal{R}\|_F^2 + \sum_n \sum_{j_n} \alpha_{j_n}^{(n)} |\boldsymbol{a}_{j_n}^{(n)}|_1 + \alpha^{\mathcal{G}}|\mathcal{G}|_1$$

$$+\frac{1}{2} I_1 I_2 \cdots I_N \log \sigma^2 - \sum_n \sum_{j_n} I_n \log \alpha_{j_n}^{(n)} - J_1 J_2 \cdots J_n \log \alpha^{\mathcal{G}}.$$

Where c is a constant. Notice, how first line corresponds to the regular l_1-regularized least squares (sparse coding) problem when alternatingly solving for the loadings of each mode keeping the remaining loadings fixed. Thus, each alternating subproblem has the form

$$\arg\min_{\boldsymbol{A}^{(n)}} \frac{1}{2}\|\boldsymbol{X}_{(n)} - \boldsymbol{A}^{(n)}\boldsymbol{Z}_{(n)}\|_F^2 + \lambda_d \sum_j |a_{j,d}|.$$

The normalization constants in the likelihood terms are given in the second line. It is due to these normalization terms that it is possible to learn the values of σ^2, $\boldsymbol{\alpha}^{(n)}$ and $\alpha^{\mathcal{G}}$.

To solve the sparse coding problem of each alternating step we used the simple gradient based procedure given in Algorithm 1. In table IV it is demonstrated that this approach is very efficient for undercomplete representations, i.e. $I_n \geq J_n$, which is normally the case for the Tucker decomposition. The approach readily generalizes to non-negativity constrained optimization by truncating negative values to zero (i.e. using projected gradient) [12]. In Algorithm 2 the algorithm for sparse ARD Tucker is given. For further details consult the Matlab implementation available for download at www.mortenmorup.dk. In general the crux of the ARD approach is that it estimates an optimal tradeoff between optimizing the likelihood of the data and the likelihood of the model parameters. Since σ^2,$\alpha^{(n)}$ and $\alpha^{\mathcal{G}}$ weights the importance of the likelihood of the data and model parameters in the objective respectively - good estimates of these parameters are the crux for the ARD approach to work well. Finally, the better the noise model as well as component priors fit the true structure of the data the better the ARD framework will work. Since estimating σ^2 from the data has a tendency of underestimating the value of σ^2 due to over-fitting, i.e. the models ability to fit noise we used the following more viable approach described in [?] to set σ^2 based on the assumption that the modelled signal ($\mathcal{R}$) and noise ($\mathcal{E}$) are uncorrelated,

$$\sigma^2 = \|\mathcal{X}\|_F^2/(I_1 I_2 \cdots I_n(1 + 10^{\mathrm{SNR}/10})).$$

where SNR is a user defined signal to noise ratio. In all the experiments we used a fixed value of $\mathrm{SNR} = 0\mathrm{dB}$ assuming

Algorithm 1 Gradient Based Sparse Coding (GBSC): $\boldsymbol{A} = \text{GBSC}(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{\lambda})$, solves $\arg\min_{\boldsymbol{A}} \frac{1}{2}\|\boldsymbol{X} - \boldsymbol{A}\boldsymbol{Z}\|_F^2 + \sum_j \lambda_j |\boldsymbol{a}_j|_1$

repeat
  Take gradient step according to LS-objective
    $\boldsymbol{A}^{new} \leftarrow \boldsymbol{A}^{old} - \mu(\boldsymbol{A}\boldsymbol{Z} - \boldsymbol{X})\boldsymbol{Z}^\top$
  Take gradient step according to l_1-regularization
  if $|a_{i,j}^{new}| < \mu\lambda_j$ **then**
    $a_{i,j}^{new} = 0$
  else
    $a_{i,j}^{new} = a_{i,j}^{new} - \mu\lambda_j \operatorname{sign}(a_{i,j}^{new})$
  end if
  Estimate μ by line-search
until Convergence

the same degree of signal as noise in the data. In [?] the sensitivity of this parameter to the obtained decomposition was investigated and it was found that the parameter had little impact for conservative choices of SNR.

III. ARD Tucker Analysis of Multi-way Data

In figure 2 is given the estimated cores obtained by the Sparse ARD Tucker algorithm on the following five datasets:
Synthetic Data: A data set with Tucker(3,4,5) structure was randomly generated with size $30 \times 40 \times 50$. All the factors as well as the core array were drawn from a normal N(0,1)-distribution, i.e. with zero mean and variance 1. Gaussian i.i.d. noise was added to the data such that $\text{SNR} = 0\text{dB}$.
Flow Injection Analysis: This data set is described in [14], [19] and is given by the absorption spectra over time for three different chemical analytes measured in 12 samples with different concentrations, i.e. $12(\text{samples}) \times 100(\text{wavelengths}) \times 89(\text{times})$, ideally this dataset form a Tucker(3,6,4) model.
Amino Acid Fluorescence: This data set is described in [3] and contains the excitation and emission spectra of five samples of different amounts of tyrosine, tryptophane and phenylalanine forming a $5(\text{samples}) \times 51(\text{excitation}) \times 201(\text{emission})$ array. Hence the data can be described by a three component CP model.
Sugar process data: This data set contain emission and excitation spectra measurements in 265 samples forming a $265(\text{samples}) \times 571(\text{emissions}) \times 7(\text{excitations})$ array [2]. The data was in [2] modeled by a four component CP model where the number of components were estimated based on an extensive split half analysis.
Dorrit fluorescence data: This data set contains the emission and excitation spectra of 27 synthetic samples containing different concentrations of four chemical analytes forming a $27(\text{samples}) \times 551(\text{emissions}) \times 24(\text{ecitations})$ array [18]. The data is adequately modeled by a four component CP model.
Since the components of the four chemometrics data sets are non-negative the estimated models for these data were constrained to be non-negative.

Clearly, regularization has both removed excess components and reduced the non-zero elements in the core, for a detailed comparison to existing methods for estimating the Tucker and CP model order see [?].

IV. Tuning the Pruning in regular Sparse Coding

The proposed ARD approach readily generalize to the regular sparse coding model proposed in [16]. Which corresponds to the Tucker model given by $\boldsymbol{X} = \boldsymbol{A}^{(1)}(\boldsymbol{G}\boldsymbol{A}^{(2)})^\top = \boldsymbol{A}^{(1)}\boldsymbol{I}\boldsymbol{S}^\top = \boldsymbol{A}^{(1)}\boldsymbol{S}^\top$. As in sparse coding we will impose sparsity on $\boldsymbol{A}^{(1)}$ henceforth denoted $\boldsymbol{A}$ while requiring that the energy of each component $\|\boldsymbol{s}_j\|_F = 1$. Hence, we minimize the following objective

$$-\log L \propto \text{c} + \frac{1}{2\sigma^2}\|\boldsymbol{X} - \boldsymbol{A}\boldsymbol{S}^\top\|_F^2 + \sum_j \alpha_j |\boldsymbol{a}_j|_1 + \frac{1}{2} I_1 I_2 \log\sigma^2 - \sum_j I_1 \log\alpha_j. \quad s.t. \quad \|\boldsymbol{s}_j\|_F = 1,$$

by alternatingly solving for $\boldsymbol{A}$, $\boldsymbol{S}$ and $\boldsymbol{\alpha}$, i.e. such that $\alpha_j = \frac{I_1}{|\boldsymbol{a}_j|_1}$ with $\sigma^2 = \|\boldsymbol{X}\|_F^2/(I_1 I_2(1 + 10^{0/10}))$. For details on the implementation see the ARDSC.M Matlab implementation available for download at www.mortenmorup.dk.

The result obtained by analyzing the natural images described in [16] is given in figure 3. 20 components have been extracted that well correspond to the Gabor like simple cell receptive field properties reported in [16].

V. Conclusion:

Model selection is perhaps one of the most challenging problems in unsupervised learning. We demonstrated how sparse coding and a simple Bayesian framework based on Automatic Relevance Determination could be adapted to the Tucker model. Sparsity enabled to prune excess components while the ARD framework enabled to learn the adequate degree of sparsity. Since the CP model and the regular sparse coding model can be considered the n-way and 2-way case of the Tucker model with diagonal core the proposed framework readily generalizes to these model.

Acknowledgement

This research was supported by the European Commission through the EU FP6 NEST Pathfinder grant PERCEPT (043261).

References

[1] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, August 2006.
[2] R. Bro. Exploratory study of sugar production using fluorescence spectroscopy and multi-way analysis. *Chemom. Intell. Lab. Syst.*, 46:133–147, 1999.
[3] Rasmus Bro. Parafac: Tutorial and applications. *Chemometrics and Intelligent Laboratory Systems*, 38:149–171, 1997.
[4] David Donoho. For most large underdetermined systems of linear equations the minimal l^1-norm solution is also the sparsest solution. *Communications on Pure and Applied Mathematics*, 59(6):797–829, 2006.
[5] R. A. Harshman. Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-modal factor analysis. *UCLA Working Papers in Phonetics*, 16:1–84, 1970.
[6] H. F. Kaiser. The varimax criterion for analytic rotation in factor analysis. *Psychometrica*, 23:187–200, 1958.

Algorithm 2 Sparse Tucker estimation based on Automatic Relevance Determination (ARD)

set $J_1, J_2, \ldots, J_n$ large enough to encompass all potential models, $\sigma^2 = \|\mathcal{X}\|_F^2/(I_1 I_2 \cdots I_n(1+10^{\text{SNR}/10}))$, set $\alpha_{\mathcal{G}} = 0$, $\boldsymbol{\alpha}^{(n)} = \mathbf{0}$ and initialize by random $\boldsymbol{A}^{(n)}$ for $n = 1, 2, \ldots, N$
repeat
  $\boldsymbol{Q} = \boldsymbol{A}^{(1)} \otimes \boldsymbol{A}^{(2)} \otimes \ldots \otimes \boldsymbol{A}^{(N)}$, $\text{vec}(\mathcal{G}) \leftarrow \text{gbsc}(\text{vec}(\mathcal{X}), \boldsymbol{Q}, \sigma^2 \alpha_{\mathcal{G}})$, $\alpha_{\mathcal{G}} = \min\{\frac{J_1 J_2 \cdots J_N}{|\mathcal{G}|_1}, \frac{1}{\epsilon}\}$
  $\mathcal{R} = \mathcal{G} \times_1 \boldsymbol{A}^{(1)} \times_2 \boldsymbol{A}^{(2)} \times_3 \ldots \times_N \boldsymbol{A}^{(N)}$
  for n=1:N **do**
    $\boldsymbol{Z}_{(n)} = (\mathcal{R} \times_n \boldsymbol{A}^{(n)^\dagger})_{(n)}$, $\boldsymbol{A}^{(n)} \leftarrow \text{gbsc}(\boldsymbol{X}_{(n)}, \boldsymbol{Z}_{(n)}, \sigma^2 \alpha^{(n)})$, $\boldsymbol{\alpha}_d^{(n)} = \min\{\frac{J_n}{|\boldsymbol{A}_d^{(n)}|_1}, \frac{1}{\epsilon}\}$
    If $\alpha_{j_n}^{(n)} = \frac{1}{\epsilon}$ **then** $J_n = J_n - 1$, $\boldsymbol{A}^{(n)} = \boldsymbol{A}_{\setminus j_n}^{(n)}$, $\mathcal{G} = \mathcal{G}_{\setminus j_n}$, $\boldsymbol{\alpha}^{(n)} = \boldsymbol{\alpha}_{\setminus j_n}^{(n)}$ **end**
    $\boldsymbol{R}_{(n)} = \boldsymbol{A}^{(n)} \boldsymbol{Z}_{(n)}$
  end for
until convergence

	100×256	256×256	1000×256	2500×256
SIGNSEARCH	0.0750 ± 0.0359	0.1984 ± 0.1342	$\mathbf{0.3734 \pm 0.1759}$	$\mathbf{1.6969 \pm 0.6441}$
CONJUGATE GRADIENT	0.4172 ± 0.0651	1.1219 ± 0.2560	9.0297 ± 1.8055	45.6297 ± 12.0142
LARS	0.0453 ± 0.0226	$\mathbf{0.1313 \pm 0.0787}$	0.4313 ± 0.1477	1.9813 ± 0.6342
NNQP	0.5703 ± 0.0696	0.9313 ± 0.0748	2.8719 ± 0.1389	15.5047 ± 0.7882
GBSC	$\mathbf{0.0125 \pm 0.0066}$	0.3172 ± 0.2121	2.0688 ± 1.0760	22.8828 ± 12.2846

TABLE I

COMPARISON OF THE CPU TIME USAGE FOR VARIOUS SPARSE CODING ALGORITHMS ON DIFFERENT PROBLEM SIZES. THE CONJUGATE GRADIENT ALGORITHM WAS OBTAINED FROM *www.l1-magic.org*, WHEREAS THE NON-NEGATIVE QUADRATIC PROGRAMMING METHOD (NNQP) AND LARS METHOD WERE OBTAINED FROM *www.sparselab.stanford.edu*. THE SIGNSEARCH ALGORITHM WAS KINDLY PROVIDED BY H. LEE [11]. THE PROBLEM SOLVED IS $\arg\min_{\boldsymbol{a}} \frac{1}{2}\|\boldsymbol{x}^\top - \boldsymbol{a}\boldsymbol{Z}\|_F^2 + \lambda\|\boldsymbol{a}\|_1$, FOR $\lambda = 0.05$. $J \times I$ DENOTES THE SIZE OF $\boldsymbol{Z}$. THE MEAN AND STANDARD DEVIATION IS GIVEN FOR 10 RANDOMLY GENERATED PROBLEMS, EACH GIVEN BY SETTING $\boldsymbol{Z}$ TO J RANDOMLY CHOSEN IMAGE PATCHES FROM THE NATURAL IMAGES DATA SET GIVEN IN [15] AND $\boldsymbol{x}$ TO A RANDOMLY SELECTED IMAGE PATCH, NOT ALREADY USED IN THE DICTIONARY, $\boldsymbol{Z}$.NOTICE, SIGNSEARCH AND LARS FIND THE GLOBAL OPTIMUM. THE REMAINING ALGORITHMS WERE STOPPED, WHEN THEIR DEVIATION FROM THE TRUE MINIMUM WAS LESS THAN 10^{-4}. FOR $J \leq I$, THE GBSC IS THE FASTEST OF ALL THE ALGORITHMS, BUT FOR OVER-COMPLETE PROBLEMS, I.E., $J \gg I$, THE GBSC ALGORITHM IS NOT IN GENERAL AS EFFECTIVE AS THE ALGORITHMS WHICH USE HESSIAN INFORMATION. THEREFORE GBSC IS ATTRACTIVE FOR THE ARD TUCKER MODEL ESTIMATION GIVEN IN ALGORITHM 2 SINCE IN GENERAL $J \leq I$.

[7] Henk A.L. Kiers. Joint orthomax rotation of the core and component matrices resulting from three-mode principal component analysis. *Journal of Classification*, 15:245–263, 1998.

[8] T.G. Kolda and B.W. Bader. Tensor decompositions and applications. *SIAM Review, to appear*, 2008.

[9] J.B Kruskal. Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. *Linear Algebra Appl.*, 18:95–138, 1977.

[10] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. Multilinear singular value decomposition. *SIAM J. MATRIX ANAL. APPL.*, 21(4):1253–1278, 2000.

[11] H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. *In Proceedings of the Neural Information Processing Systems (NIPS)*, 19, 2007.

[12] C.-J. Lin. Projected gradient methods for non-negative matrix factorization. *Neural Computation*, 19:2756–2779, 2007.

[13] M. Mørup. *Decomposition Methods for Unsupervised Learning*. PhD thesis, Technical University of Denmark, 2008.

[14] L Nørgaard and C. Ridder. Rank annihilation factor analysis applied to flow injection analysis with photodiode-array detection. *Chemometrics and Intelligent Laboratory Systems*, 23(1):107–114, 1994.

[15] B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609, 1996.

[16] Bruno A. Olshausen and David J. Field. Sparse coding of sensory inputs. *Current Opinion in Neurobiology*, 14:481–487, 2004.

[17] Yuan (Alan) Qi, Thomas P. Minka, Rosalind W. Picard, and Zoubin Ghahramani. Predictive automatic relevance determination by expectation propagation. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, page 85, New York, NY, USA, 2004. ACM.

[18] Jordi Riu and R Bro. Jack-knife estimation of standard errors and outlier detection in parafac models. *Chemometrics and Intelligent Laboratory Systems*, 65(1):35–49, 2003.

[19] Age K. Smilde, Roma Tauller, Javier Saurina, and Rasmus Bro. Calibration methods for complex second-order data. *Analytica Chimica Acta*, 398:237–251, 1999.

[20] L. R. Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31:279–311, 1966.

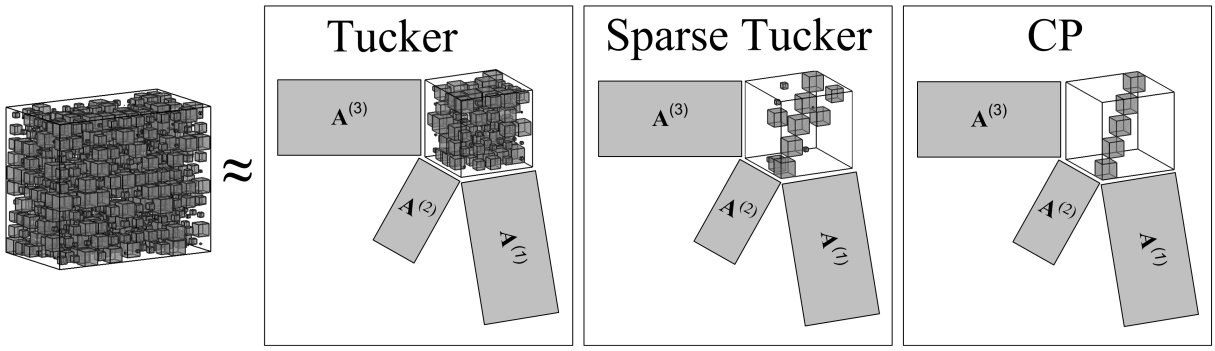

Fig. 1. Illustration of Tucker decomposition of a 3-way tensor given to the left. The Tucker model is given to the left, sparse Tucker model in the middle and CP model to the right. Whereas the Tucker model encompass all potential interaction between the components of each modality through the core array $\mathcal{G}$, the CP model only allow for interactions between columns of $\boldsymbol{A}^{(n)}$ with same indices. The sparse Tucker model can be considered a model between the Tucker and CP model where interactions are present within a few of the components across the various modalities by imposing sparsity on the core. We will impose sparsity on the core and loadings to prune excess components while estimate the adequate degree of sparsity using a Bayesian approach named Automatic Relevance Determination (ARD).

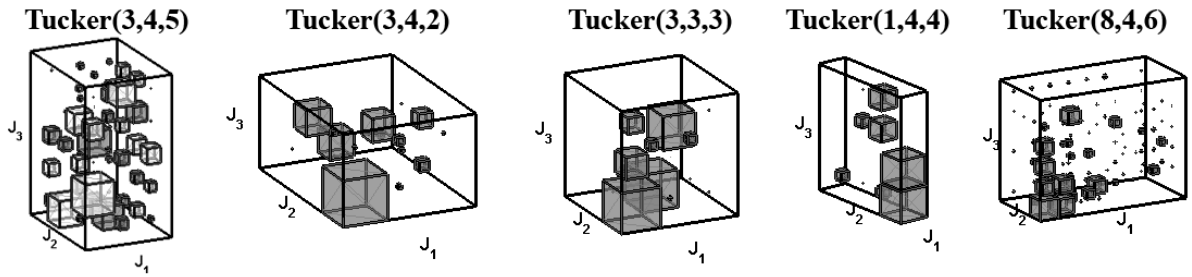

Fig. 2. The estimated cores for five different multi-way datasets – a Tucker(3,4,5) synthetic dataset as well as four 3-way chemometrics data sets obtained from *www.models.kvl.dk/research/data/*. A Tucker(10,10,10) model was fitted to all datasets but using sparse coding to prune excess components and ARD to update the l_1-regularization strengths on the model parameters the models were reduced to form simpler models. Given are the estimated model orders and cores (size of boxes indicate interaction strengths such that gray boxes denote positive interactions and white boxes negative interactions). The estimated model orders relate well to the expected model order based on the number of true chemical compounds in the data.

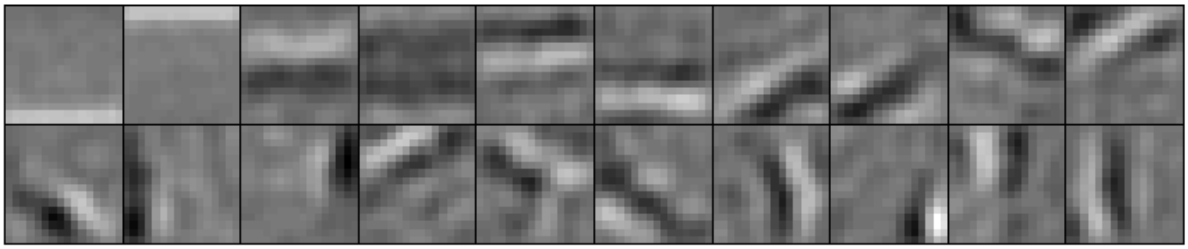

Fig. 3. Analysis of the regular sparse coding problem using the proposed ARD approach to tune the pruning parameters. 250 components were fitted to the data however all but 20 components were turned off by the proposed ARD framework. The components correspond well to the Gabor like simple cell receptive field properties reported in [16].