Automatic Relevance Determination for Multi-way Models

Morten Mørup and Lars Kai Hansen
DTU Informatics
e-mail: {mm,lkh}@imm.dtu.dk

Abstract

Estimating the adequate number of components is an important yet difficult problem in multi-way modelling. We demonstrate how a Bayesian framework for model selection based on Automatic Relevance Determination (ARD) can be adapted to the Tucker and CP models. By assigning priors for the model parameters and learning the hyperparameters of these priors the method is able to turn off excess components and simplify the core structure at a computational cost of fitting the conventional Tucker/CP model. To investigate the impact of the choice of priors we based the ARD on both Laplace and Gaussian priors corresponding to regularization by the sparsity promoting l_1-norm and the conventional l_2-norm, respectively. While the form of the priors had limited effect on the results obtained the ARD approach turned out to form a useful, simple, and efficient tool for selecting the adequate number of components of data within the Tucker and CP structure. For the Tucker and CP model the approach performs better than heuristics such as the Bayesian Information Criterion, Akaikes Information Criterion, DIFFIT and the numerical convex hull (NumConvHull) while operating only at the cost of estimating an ordinary CP/Tucker model. For the CP model the ARD approach performs almost as well as the core consistency diagnostic. Thus, the ARD framework is a simple yet efficient tool for the estimation of the adequate number of components in multi-way models. A Matlab implementation of the proposed algorithm is available for download at `www.erpwavelab.org`.

1 Introduction

Tensor decompositions are in frequent use today in a variety of fields including psychometrics (28), chemometrics (29; 35), image analysis (40), web data mining (1), bio-informatics (30), neuroimaging (25; 3), and signal processing (34). Tensors, i.e., $\mathcal{X} \in \mathbb{C}^{I_1 \times I_2 \times \ldots \times I_N}$, 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 (39) and the more restricted Canonical Decomposition (CandeComp) and Parallel Factor Analysis (PARAFAC) model proposed independently by (11, 16). We will presently denote the CandeComp/PARAFAC model CP.

The Tucker model reads

$$\mathcal{X}_{i_1,i_2,\ldots,i_N} \approx \mathcal{R}_{i_1,i_2,\ldots,i_N} = \sum_{j_1 j_2 \cdots j_N} \mathcal{G}_{j_1,j_2,\ldots,j_N} \mathbf{A}^{(1)}_{i_1,j_1} \mathbf{A}^{(2)}_{i_2,j_2} \cdot \ldots \cdot \mathbf{A}^{(N)}_{i_N,j_N}.$$

where $\mathcal{G} \in \mathbb{C}^{J_1 \times J_2 \times \ldots \times J_N}$ and $\mathbf{A}^{(n)} \in \mathbb{C}^{I_n \times J_n}$. 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 tensor product $\times_n$ (23) 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

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

The Tucker model represents the data spanning the n^{th} modality by the vectors (loadings) given by the J_n columns of $\mathbf{A}^{(n)}$ 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 be expressed as a Tucker model with diagonal core. In particular, by appropriate scaling of each component the CP model can be expressed as a Tucker model with unit diagonal core, i.e. $\mathcal{G}_{CP} = \mathcal{I}$. The Tucker model can in turn be expressed as the CP model by duplicating components of different indices to form additional CP components (17). Notice, in the Tucker model a rotation of a given loading matrix $\mathbf{A}^{(n)}$ can be compensated by a counter rotation of the core $\mathcal{G}$, i.e., $\mathcal{G} \times_n \mathbf{A}^{(n)} = (\mathcal{G} \times_n \mathbf{P}^{-1}) \times_n (\mathbf{A}^{(n)}\mathbf{P})$. 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 in general unique up to scale and permutation (22).

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, see also figure 1. Hence, 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 $\mathbf{A}^{(n)}$ 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. Several strategies exist for simplifying the Tucker core. In (20) the Tucker solution was rotated such that the Tucker core would have as many small loadings as possible while in (26) the core was regularized penalizing deviation from sparsity on the core. Thus, by regularizing the Tucker model excess components can be turned off and the Tucker core be simplified. We will presently estimate the adequate degree of regularization by a Bayesian approach named Automatic Relevance Determination (ARD). Two types of

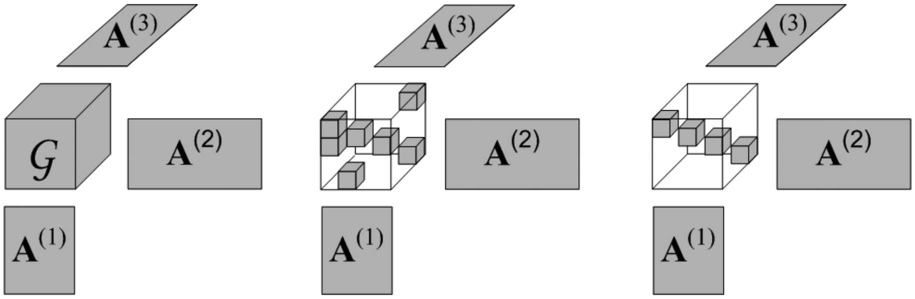

Figure 1: Illustration of the Tucker model (to the left), sparse Tucker model (in the middle) and CP model (to the right).

regularization will be considered; the sparsity promoting l_1 regularization as well as the more conventional l_2 ridge regression regularization. The approach readily generalize to the CP model and will also here be used to estimate the number of components.

Choosing the right model is in particular challenging in the Tucker model as the number of components is specified for each modality separately. This renders heuristics such as the DIFFIT (38), Numerical Convex Hull (12), Bayesian Information Criterion (BIC) (33) and Akaikes Information Criterion (AIC) (2) as well as cross-validation approaches (8; 36) computationally expensive as $J_1 J_2 \cdots J_n$ models have to be evaluated. Furthermore, while model selection for the CP model has been guided by heuristics based on the Core Consistency Diagnostic (10), no such heuristics exist for the Tucker model. In 2-way analysis it is common to evaluate the eigenvalue spectrum and truncate the singular value decomposition (SVD). Although this approach does not have a straightforward multi-linear counterpart (13) approximate approaches have been given forming the fastDIFFIT (19). However, this approach can not account for additional constraints such as non-negativity. In conclusion, no efficient approach for the estimation of the number of components in the Tucker model is known. Thus, the aim of this paper is

- to use regularization to turn off excess components in the CP and Tucker model and thereby select the model order and simplify the core (as proposed in (26)).
- to optimize the amount of regularization from data.
- to achieve these objective 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) (24; 6; 31). Traditionally, ARD has been based on Gaussian priors yielding a ridge regression type of selection. Here, we will derive an ARD approach both based on the Gaussian prior as well as the Laplace prior to understand better the role of priors on the models found. Contrary to Gaussian priors, the Laplace prior favors sparse representations hence attempts to minimize the number of non-zero elements within the active components of the model (14). Optimizing for sparse representation is related to the classic rotation criteria such as VARIMAX (18) and maximum Likelihood independent component analysis (ICA) based on sparse priors (27). 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 (27). If no formal prior information is given parsimony can be considered a reasonable guiding principle to avoid overfitting, see also (27) and references therein.

The paper is structured as follows. In section 2 we will establish the traditional Tucker model in a Bayesian framework and assign priors to all loadings as well as the core array. Based on maximum a posteriori (MAP) estimation we will both estimate the model parameters as well as the parameters controlling the degree of regularization, hence indirectly the model order. In section 3 we will investigate the performance of the proposed approach both in finding the underlying components of data with Tucker structure as well as data with CP structure. Comparison to existing model order selection heuristics will be given.

2 Methods

2.1 Notation

In the following $\mathbf{X}_{(n)}$ is the n-mode matricized version of the tensor $\mathcal{X}$ where the n-mode matricizing operation turn the array $\mathcal{X}^{I_1 \times I_2 \times \ldots \times I_N}$ into a matrix, i.e., $\mathbf{X}_{(n)}^{I_n \times I_1 \ldots I_{n-1} I_{n+1} \ldots I_N}$. The Frobenius and 1-norm of a tensor will be denoted by

$$\|\mathcal{X}\|_F^2 = \sum_{i_1,i_2,\ldots,i_n} \mathcal{X}_{i_1,i_2,\ldots,i_n}^2 \text{ and } |\mathcal{X}|_1 = \sum_{i_1,i_2,\ldots,i_n} |\mathcal{X}_{i_1,i_2,\ldots,i_n}|, \tag{1}$$

while $\mathcal{X}_{\backslash j_n}$ will denote $\mathcal{X}$ with the j_n'th slab removed – this notation also holds for vectors and matrices which can be considered 1st and 2nd order tensors. Using the n-mode multiplication we will define the reconstructed data according to the Tucker model $\mathcal{R}$ with the $\mathbf{A}^{(n)}$'th loading removed by

$$(\mathcal{R} \times_n \mathbf{A}^{(n)^\dagger}) = \mathcal{G} \times_1 \mathbf{A}^{(1)} \times_2, \cdots \times_{n-1} \mathbf{A}^{(n-1)} \times_{n+1} \mathbf{A}^{(n+1)} \ldots \times_N \mathbf{A}^{(N)},$$

where $\mathbf{A}^{(n)^\dagger}$ denotes the Moore-Penrose pseudo inverse. When $\mathbf{A}^{(n)}$ has full column rank we will use the left side otherwise we will use the right hand side of the expression to calculate $(\mathcal{R} \times_n \mathbf{A}^{(n)^\dagger})$.

Algorithm 1 Tucker estimation based on Alternating Least Squares (ALS)

```
set $J_1, J_2, \ldots, J_n$ and initialize by random $\mathbf{A}^{(n)}$ for $n = 1, 2, \ldots, N$
repeat
    $\mathbf{Q} = \mathbf{A}^{(1)} \otimes \mathbf{A}^{(2)} \otimes \ldots \otimes \mathbf{A}^{(N)}$
    $\mathrm{vec}(\mathcal{G}) \leftarrow solve(\mathrm{vec}(\mathcal{X}), \mathbf{Q})$
    $\mathcal{R} = \mathcal{G} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \times_3 \ldots \times_N \mathbf{A}^{(N)}$
    for n=1:N do
        $\mathbf{Z}_{(n)} = (\mathcal{R} \times_n \mathbf{A}^{(n)^\dagger})_{(n)}$
        $\mathbf{A}^{(n)} \leftarrow solve(\mathbf{X}_{(n)}, \mathbf{Z}_{(n)})$
        $\mathbf{R}_{(n)} = \mathbf{A}^{(n)}\mathbf{Z}_{(n)}$
    end for
until convergence
```

2.2 Tucker estimation based on Alternating Least Squares

Using the n-mode matricizing and Kronecker product operation the Tucker model can be written as

$$\begin{aligned} \boldsymbol{X}_{(n)} &\approx \mathbf{R}_{(n)} = \boldsymbol{A}^{(n)}\boldsymbol{Z}_{(n)}, \\ \mathrm{vec}(\mathcal{X}) &\approx \mathrm{vec}(\mathcal{R}) = \mathrm{vec}(\mathcal{G})\mathbf{Q}, \quad \text{where} \\ \boldsymbol{Z}_{(n)} &= \mathbf{G}_{(n)}(\mathbf{A}^{(N)} \otimes ... \otimes \mathbf{A}^{(n+1)} \otimes \mathbf{A}^{(n-1)} \otimes ... \otimes \mathbf{A}^{(1)}) = (\mathcal{R} \times_n \mathbf{A}^{(n)^\dagger})_{(n)}, \\ \mathbf{Q} &= \mathbf{A}^{(1)} \otimes \mathbf{A}^{(2)} \otimes \ldots \otimes \mathbf{A}^{(N)}. \end{aligned}$$

Traditionally, the Tucker model has been estimated using various types of alternating least squares algorithms (4; 21). By alternating least squares the model estimation reduces to a sequence of regular matrix analysis problem where each mode is updated keeping the loadings of the remaining modes fixed. As a result, for least squares minimization the estimation problem can be solved by pseudo-inverses, i.e.

$$\begin{aligned} \boldsymbol{A}^{(n)} &\leftarrow \boldsymbol{X}_{(n)}\boldsymbol{Z}_{(n)}^\dagger, \\ \mathcal{G} &\leftarrow \mathcal{X} \times_1 \boldsymbol{A}^{(1)^\dagger} \times_2 \boldsymbol{A}^{(2)^\dagger} \times_3 ... \times_N \boldsymbol{A}^{(N)^\dagger}. \end{aligned}$$

The alternating least squares (ALS) algorithm for Tucker model estimation is given in algorithm 1. We denote the update of $\mathbf{A}^{(n)}$ and $\mathcal{G}$ formed by the regular linear regression subproblems by $solve(\cdot, \cdot)$. For unconstrained optimization the solution is simply found by pseudo-inverses. However, if $\mathbf{A}^{(n)}$ or $\mathcal{G}$ are constrained to be non-negative the problem becomes a standard quadratic programming problem. The convergence of the algorithm we defined as a relative change in the sum of squared error (SSE $= \|\mathcal{X} - \mathcal{R}\|_F^2$) less than 10^{-9} or when 500 iterations had progressed. We note that while the estimation of each mode keeping the other modes fixed is a convex optimization problem, the joint estimation of all model parameters is a non-convex optimization problem hence potentially prone to local minima.

2.3 Model selection based on AIC, BIC, DIFFIT NumConvHull and Core Consistency Diagnostic

In model selection Akaike's Information Criterion (AIC) and the Bayesian Information Criterion (BIC) have traditionally been used as simple approximations to the expectation of the negative log likelihood and the model evidence respectively (2; 33). Here, the number of components are selected such that the following two quantities are minimized

$$\begin{aligned} \text{AIC} &= -2\log L + K = S\log\frac{\text{SSE}}{S} + K \\ \text{BIC} &= -2\log L + K\log S = S\log\frac{\text{SSE}}{S} + K\log S \end{aligned}$$

Where L is the likelihood of the model, K is the number of parameters in the model, and $S = \prod_n I_n$ the number of data points. For least square estimation this reduces to the expressions to the right where SSE is the sum of squared error. Thus, the criteria defines a tradeoff between reduction in reconstruction error and complexity of the model. Notice that BIC tends to penalize model complexity more heavily than AIC, hence, gives a more conservative estimate of what is considered the best model.

For the Tucker model the DIFFIT procedure (38) has been proposed to estimate the adequate number of components. In the DIFFIT procedure, all potential models are evaluated and the m^{th} model where $m = \sum_n J_n$ given by $\mathcal{R}^m$ with the best explained variance, i.e. $\text{ExpVar}(m) = 1 - \frac{\|\mathcal{X}-\mathcal{R}^m\|_F^2}{\|\mathcal{X}\|_F^2}$ calculated. The DIFFIT for the m^{th} model is then calculated as

$$\begin{aligned} \text{DIF}(m) &= \text{ExpVar}(m) - \text{ExpVar}(m-1) \\ \text{DIFFIT}(m) &= \text{DIF}(m)/\text{DIF}(m+1) \end{aligned}$$

And the model with largest $DIFFIT$ value taken to be the most adequate model when disregarding DIFFIT values based on too small values of DIF (38; 19). Hence, the optimal model is given by the model that has the largest contribution to the explained variance relative to consecutive models corresponding to the region of maximal curvature in the graph of $\{m, ExpVar\}$. An approximate evaluation of DIFFIT forming the fastDIFFIT (19) is given by evaluating the eigenvectors of $\mathbf{X}_{(n)}$ for all n-modes and take the best models $\mathcal{R}^m$ formed by the HOSVD (23). Notice, for a 3-way Tucker model, $m = 3$ (i.e. a Tucker (1,1,1)) and $m = 4$ (i.e. for instance given by a Tucker (2,1,1)) pose the same modeling ability thus $m = 4$ is ignored. A refinement of the above approach correcting for the number of free parameters (FP) for the p^{th} Tucker model $\text{FP}(p) = \sum_n I_n J_n + \prod_n J_n - \sum_n J_n^2$ form the numerical convex hull (Num-

ConvHull)[1] approach (12) given by

$$\begin{aligned}\mathrm{FPDIF}(p) &= \mathrm{FP}(p) - \mathrm{FP}(p-1)\\ \mathrm{NumConvHull}(p) &= \frac{\mathrm{DIF}(p)/\mathrm{FPDIF}(p)}{\mathrm{DIF}(p+1)/\mathrm{FP}(p+1)}.\end{aligned}$$

The approach is motivated by inspecting the convex hull formed by the plot {FP,ExpVar}, i.e. inspecting data points with largest explained variance (ExpVar) relative to degrees of freedom (FP) such that p index over the solutions forming the convex hull. The maximal NumConvHull indicates the region of maximal curvature in the convex hull and as such indicates the optimal tradeoff between improvement in fitting data relative to number of additional free parameters used in the model. For the CP model $FP(d) = d\sum_n I_n$ where $d = J_1 = \ldots = J_n$ such that $\mathrm{FPDIF}(d) = \sum_n I_n$ hence the approach reduces to the regular DIFFIT procedure.

For the CP model the core consistency has been used as a heuristic to access the adequate number of components (10). The core consistency measures the degree of cross-talk between the components of the CP model by estimating the corresponding Tucker model core $\mathcal{G}$ given the CP loadings, i.e.

$$\mathrm{CorConDiag} = 100 \cdot (1 - \frac{\sum_{i_1,i_2,\ldots,i_n}(\mathcal{G}_{i_1,i_2,\ldots,i_n} - \mathcal{I}_{i_1,i_2,\ldots,i_n})^2}{\sum_{i_1,i_2,\ldots,i_n}\mathcal{I}^2_{i_1,i_2,\ldots,i_n}})$$

Where $\mathcal{G}$ is estimated as

$$\mathcal{G} \leftarrow \mathcal{X} \times_1 \boldsymbol{A}_{CP}^{(1)^\dagger} \times_2 \boldsymbol{A}_{CP}^{(2)^\dagger} \times_3 \ldots \times_N \boldsymbol{A}_{CP}^{(N)^\dagger}.$$

$\boldsymbol{A}_{CP}^{(n)}$ is the n-mode loadings of the CP solution. Since the Tucker model encompass all potential interactions between components of the various modes non-zero values in the off-diagonal of the Tucker core indicate that structure in components of different indices over the modalities can combine resulting in so-called cross-talk. Too many components will result in a strong degree of cross-talk across the loadings of the modes thus will yield a low value of the CorConDiag. Too few components on the other hand will exhibit a low degree of cross-talk. Thus, a heuristic for the "correct" number of components is taken to be just before a major drop-off in the graph of $\{d, CorConDiag\}$(10) where $d = J_1 = J_2 = \ldots = J_n$.

2.4 Automatic Relevance Determination for Multi-way models

Automatic Relevance Determination (ARD) is a hierarchical Bayesian approach widely used for model selection (24; 31; 6). In ARD hyperparameters explicitly represents the

[1] The number of free parameters is given by the number of elements in the factors of each mode and core, i.e. first and second term of FP subtracted the reduction in degrees of freedom due to the orthonormality constraints (third term). When imposing sparsity or non-negativity the loadings are no longer necessarily orthogonal. Therefore, the degrees of freedom should no longer be subtracted the third term. As a result, a better definition of FP would here be the number of non-zero elements in the loadings and core. Since the original definition of FP, FP without subtracting the orthonormality term as well as FP given by the number of non-zero elements in core and loadings all shared the same best model for the analyzed data the NumConvHull results reported are for the original formulation of FP given in (12).

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. The reason being that the l_2-regularization penalizes elements by their squares and as such penalizes large values relatively more than small values. The Laplace prior on the other hand 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 (14). Due to their different nature, we will both consider the Gaussian as well as the Laplace priors on the model parameter $\boldsymbol{\theta}_d$, i.e.

$$P_{Gaussian}(\boldsymbol{\theta}_d|\alpha_d) = \prod_j \left(\frac{\alpha_d}{2\pi}\right)^{\frac{1}{2}} \exp[-\frac{\alpha_d}{2}\theta_{j,d}^2]$$

$$P_{Laplace}(\boldsymbol{\theta}_d|\alpha_d) = \prod_j \frac{\alpha_d}{2} \exp[-\alpha_d|\theta_{j,d}|].$$

In a Bayesian framework, the least squares objective

$$\text{SSE} = \|\mathcal{X} - \mathcal{R}\|_F^2 = \sum_{i_1,i_2,\ldots,i_n} (\mathcal{X}_{i_1,i_2,\ldots,i_n} - \mathcal{R}_{i_1,i_2,\ldots,i_n})^2,$$

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.

$$\begin{aligned} P(\mathcal{X}|\mathcal{R},\sigma^2) &= \prod_{i_1,i_2,\ldots,i_N} \frac{1}{\sqrt{2\pi\sigma^2}} \exp[-\frac{(\mathcal{X}_{i_1,i_2,\ldots,i_N} - \mathcal{R}_{i_1,i_2,\ldots,i_N})^2}{2\sigma^2}] \\ &= (2\pi\sigma^2)^{-\frac{I_1 I_2 \cdots I_N}{2}} \exp[-\frac{\|\mathcal{X} - \mathcal{R}\|_F^2}{2\sigma^2}]. \end{aligned}$$

In the following we derive an algorithm for estimating both the number of components as well as the model parameters of the Tucker model based on ARD. We note that the corresponding CP algorithm is derived by fixing the core $\mathcal{G} = \mathcal{I}$.

Assigning Laplace or Gaussian priors for the loadings and core we get

$$\begin{aligned} P_{Laplace}(\mathbf{A}^{(n)}|\boldsymbol{\alpha}^{(n)}) &= \prod_d \left(\frac{\alpha_d^{(n)}}{2}\right)^{I_n} \exp[-\alpha_d^{(n)}|\mathbf{A}_d^{(n)}|_1], \\ P_{Laplace}(\mathcal{G}|\alpha^{\mathcal{G}}) &= \left(\frac{\alpha^{\mathcal{G}}}{2}\right)^{J_1 J_2 \cdots J_n} \exp[-\alpha^{\mathcal{G}}|\mathcal{G}|_1], \\ P_{Gaussian}(\mathbf{A}^{(n)}|\boldsymbol{\alpha}^{(n)}) &= \prod_d \left(\frac{\alpha_d^{(n)}}{2\pi}\right)^{\frac{I_n}{2}} \exp[-\frac{\alpha_d^{(n)}}{2}\|\mathbf{A}_d^{(n)}\|_F^2], \\ P_{Gaussian}(\mathcal{G}|\alpha^{\mathcal{G}}) &= \left(\frac{\alpha^{\mathcal{G}}}{2\pi}\right)^{\frac{J_1 J_2 \cdots J_n}{2}} \exp[-\frac{\alpha^{\mathcal{G}}}{2}\|\mathcal{G}\|_F^2]. \end{aligned}$$

In a hierarchical Bayesian framework we could further assign priors on the hyper-parameters $\boldsymbol{\alpha}^{(n)}$ and $\alpha^{\mathcal{G}}$ (see also (37)), however, we will here use simple uniform (noninformative) priors on the hyper-parameters. As a result, the posterior can be written as

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

Thus the negative log likelihood using Gaussian priors is proportional to

$$\begin{aligned} -\log L &\propto \text{const.} + \frac{1}{2\sigma^2}\|\mathcal{X} - \mathcal{R}\|_F^2 + \frac{1}{2}\sum_n \sum_D \alpha_d^{(n)} \|\mathbf{A}_d^{(n)}\|_F^2 + \alpha^{\mathcal{G}}\|\mathcal{G}\|_F^2 \\ &+ \frac{1}{2} I_1 I_2 \cdots I_N \log \sigma^2 - \frac{1}{2}\sum_n \sum_d I_n \log \alpha_d^{(n)} - \frac{1}{2} J_1 J_2 \cdots J_n \log \alpha^{\mathcal{G}}, \end{aligned}$$

and using Laplace priors proportional to

$$\begin{aligned} -\log L &\propto \text{const.} + \frac{1}{2\sigma^2}\|\mathcal{X} - \mathcal{R}\|_F^2 + \sum_n \sum_d \alpha_d^{(n)} |\mathbf{A}_d^{(n)}|_1 + \alpha^{\mathcal{G}}|\mathcal{G}|_1 \\ &+ \frac{1}{2} I_1 I_2 \cdots I_N \log \sigma^2 - \sum_n \sum_d I_n \log \alpha_d^{(n)} - J_1 J_2 \cdots J_n \log \alpha^{\mathcal{G}}. \end{aligned}$$

Notice, how first line corresponds to a l_2-regularized and l_1-regularized least squares problem respectively while the normalization constants in the likelihood terms are given in the second lines. 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 for the l_2-regularized parameters is equivalent to the regular ridge regression problem, i.e.

$$\frac{1}{2\sigma^2}\|\mathbf{X} - \mathbf{A}\mathbf{S}\|_F^2 + \frac{1}{2}\sum_d \alpha_d \|\mathbf{A}_d\|_F^2,$$

which has the solution

$$\mathbf{A} = \mathbf{X}\mathbf{S}^\top (\mathbf{S}\mathbf{S}^\top + \sigma^2 \operatorname{diag}(\boldsymbol{\alpha}))^{-1}.$$

To solve for the l_1-regularized parameters on the other hand form the regular sparse regression problem also denoted the LASSO or Basis Pursuit De-noising (BPD), i.e.

$$\frac{1}{2\sigma^2}\|\mathbf{X} - \mathbf{A}\mathbf{S}\|_F^2 + \sum_d \alpha_d |\mathbf{a}_d|_1,$$

which has the solution

$$\mathbf{A} = (\mathbf{X}\mathbf{S}^\top - \sigma^2 \operatorname{sign}(\mathbf{A}) \operatorname{diag}(\boldsymbol{\alpha}))(\mathbf{S}\mathbf{S}^\top)^{-1}.$$

For a review of approaches to solve this sparse regression problem see (27). In the following we will simply write $\mathbf{A} \leftarrow \text{solve}_{Sparse/Ridge}(\mathbf{X}, \mathbf{S}, \sigma^2\boldsymbol{\alpha})$ when solving for the parameters either by sparse or ridge regression.

Differentiating the negative log likelihood with respect to the so-called hyperparameters and equating the derivatives to zero we get the following parameter updates using Gaussian priors

$$\sigma^2 = \frac{\|\mathcal{X}-\mathcal{R}\|_F^2}{I_1 I_2 \cdots I_N},\ \alpha_d^{(n)} = \frac{I_n}{\|\mathbf{A}_d^{(n)}\|_F^2},\ \alpha^{\mathcal{G}} = \frac{J_1 J_2 \ldots J_N}{\|\mathcal{G}\|_F^2}.$$

and using Laplace priors

$$\sigma^2 = \frac{\|\mathcal{X}-\mathcal{R}\|_F^2}{I_1 I_2 \cdots I_N},\ \alpha_d^{(n)} = \frac{I_n}{|\mathbf{A}_d^{(n)}|_1},\ \alpha^{\mathcal{G}} = \frac{J_1 J_2 \ldots J_N}{|\mathcal{G}|_1},$$

According to the above updates σ^2 can technically be learned from the data. However, estimating σ^2 from data has a tendency of underestimating the value of σ^2 due to overfitting, i.e. the models ability to fit noise. We therefore used the following more viable approach to set σ^2: Let $\mathcal{X} = \mathcal{R} + \mathcal{E}$ and assume the signal modeled $\mathcal{R}$ and the noise $\mathcal{E}$ are uncorrelated – we then have $\|\mathcal{X}\|_F^2 = \|\mathcal{R}\|_F^2 + \|\mathcal{E}\|_F^2$. As a result, the signal to noise ratio (SNR) is given by $\mathrm{SNR} = 10 \log \frac{\|\mathcal{R}\|_F^2}{\|\mathcal{E}\|_F^2} = 10 \log \frac{\|\mathcal{X}\|_F - \|\mathcal{E}\|_F^2}{\|\mathcal{E}\|_F^2}$. Assuming the noise is normal i.i.d. we have $\|\mathcal{E}\|_F^2 = \sigma^2 \prod_n I_n = \sigma^2 S$ (where $S = \prod_n I_n$), therefore

$$\sigma^2 = \|X\|_F^2 / (S(1 + 10^{\mathrm{SNR}/10})). \tag{2}$$

In all the experiments we used a fixed value of $\mathrm{SNR} = 0\mathrm{dB}$ assuming the same degree of signal as noise in the data. However, in figure 2 we investigated the impact of the choice of SNR.

The algorithm for ARD Tucker estimation is given in algorithm 2. Notice, how the updates of the hyper-parameters correspond to setting the hyper-parameters of the priors such that they match the posteriors distribution as derived in (15). Furthermore, the model selection, i.e. the updates of the hyper-parameters, comes at practically no extra computational cost compared to the ordinary Tucker ALS algorithm. As a result, the cost per iteration of the ARD Tucker is the same as for the ordinary ALS Tucker algorithm.

A few notes to the algorithm: When imposing non-negativity constraints on $\mathbf{A}^{(n)}$ and $\mathcal{G}$ the prior is no longer given by the Laplace but the exponential distribution. However, this only changes the normalization constant by a constant hence does not change the updates. Similarly, the Gaussian prior becomes a rectified Gaussian i.e. the corresponding distribution derived by setting the density of the Gaussian to zero in regions where parameters are negative. Thus, despite the Gaussian being a conjugate prior to the least squares error (i.e., the posterior distribution is also Gaussian), the rectified Gaussian as well as the Laplace prior are not conjugate and can as such not be integrated analytically to estimate the mean and covariance of the posterior (6). As a result, all parameter estimates are based on maximum a posteriori (MAP) estimation. We kept $\boldsymbol{\alpha} = \mathbf{0}$ for the first 25 iterations to avoid basing the ARD on too poor model estimates. For components that had become zero or close to zero we set $\alpha = \frac{1}{\epsilon}$ where $\epsilon = 10^{-9}$. For further details on the algorithm consult the matlab implementation available for download at `www.erpwavelab.org`.

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. Thus, as for the existing heuristics such as BIC, AIC, DIFFIT and NumConvHull

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

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

there is an inherent tradeoff between fitting the data and model complexity invoked by the regularization terms formed by the model parameter priors. As such, BIC, AIC and NumConvHull can be regarded approaches where model complexity is measured with respect to the l_0-norm, i.e. by the number of free model parameters. However, whereas the existing heuristics has to evaluate all potential models the ARD approach prune excess components and learn the model order at the cost of fitting one regular Tucker model. Since σ^2 and α 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. We note that as for the regular ALS Tucker estimation the ARD Tucker algorithm is potentially prone to local minima.

3 Results

We analyzed a total of five different datasets two with Tucker structure and 3 with CP structure. The data with CP structure were both analyzed based on the ARD TUCKER and the ARD CP model. We note again that the CP model is simply formed by setting the Tucker core $\mathcal{G} = \mathcal{I}$. We compared the estimated number of components by the ARD approach to the estimated number of components found by the DIFFIT, NumConvHull, Bayesian Information Criterion, Akaikes Information Criterion and for the CP model also the Core Consistency Diagnostic. To compare the impact of the choice of prior we fitted the models both using Gaussian as well as Laplace priors based on identical initializations. We further investigated the impact of choice of SNR. For brevity we will denote the analysis based on Gaussian priors – ridge ARD, and Laplace priors – sparse ARD.

3.1 Data

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 $\mathrm{SNR} = 0\mathrm{dB}$.
Flow Injection Analysis: This data set is described in (29; 35) and is given by the absorption spectra over time for three different chemical analytes measured in 12 samples with different concentrations, i.e. $12(\mathrm{samples}) \times 100(\mathrm{wavelengths}) \times 89(\mathrm{times})$, ideally this dataset form a Tucker(3,6,4) model.
Amino Acid Fluorescence: This data set is described in (9) and contains the excitation and emission spectra of five samples of different amounts of tyrosine, tryptophane and phenylalanine forming a $5(\mathrm{samples}) \times 51(\mathrm{excitation}) \times 201(\mathrm{emission})$ 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 $265(\mathrm{samples}) \times 571(\mathrm{emissions}) \times 7(\mathrm{excitations})$ array (7). The data was in (7) 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 $27(\mathrm{samples}) \times 551(\mathrm{emissions}) \times 24(\mathrm{ecitations})$ array (32). 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.

3.2 ARD Tucker analysis

In figure 2 the impact of the choice of signal to noise ratio SNR is investigated. For the synthetic data a clear break point around $\mathrm{SNR} = 0\mathrm{dB}$ is found such that lower SNR values identify the correct model order for both sparse and ridge ARD Tucker whereas higher SNR values makes the ARD approach completely fail in identifying the correct number of components as the model fit noise. A similar behavior is found for the remaining data sets. Namely that high SNR values tend to over-estimate the number of components whereas low SNR values perform more stable. As such, the exact choice of SNR seem to have little impact on the model order found as long as SNR is not set too large. Thus, when there is no prior information as to the true SNR of the data it seems to be better to use low estimates of the SNR rather than large SNR values as large SNR values has a tendency to use too many components hence overfit the data. In the following analysis we set $\mathrm{SNR} = 0\mathrm{dB}$.

In table 1 is given the result running 20 ARD Tucker$(10, 10, 10)$ models both using sparse ARD and ridge ARD. From the table it can be seen that both the sparse and ridge ARD Tucker correctly identifies a Tucker$(3, 4, 5)$ component model for the synthetic data. A Tucker$(3, 4, 2)$ and Tucker$(3, 4, 3)$ model for the FIA data respectively hence

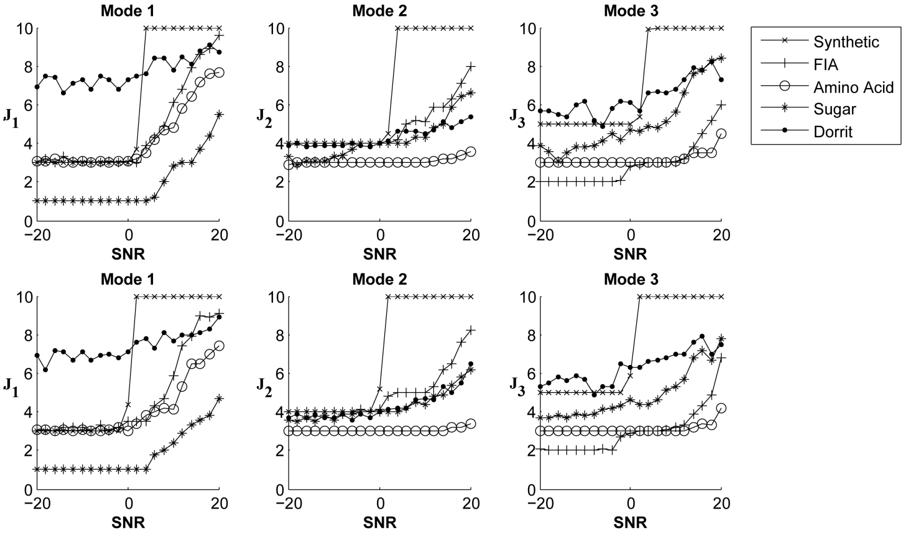

Figure 2: Analysis of the impact of choice of SNR (in dB) on the models identified. Top row gives the mean number of components over 10 runs for each data set of each mode for various choices of SNR using the sparse ARD Tucker. Bottom row gives the corresponding analysis based on the ridge ARD Tucker.

correctly identifying the number of analytes but describing the underlying spectra and time courses by a mixture of fewer components. For the Amino Acid Fluorescence both sparse and ridge ARD Tucker correctly identifies a Tucker$(3, 3, 3)$ component model and for the sugar process a Tucker$(1, 4, 4)$ component model correctly identifying the number of emission and excitation spectra. For the Dorrit data both models identify a Tucker$(8, 4, 6)$ component model hence correctly identifying four emission spectra but wrongfully identifying too many sample components and excitation spectra.

In figure 3 is given the estimated cores for the best sparse and ridge ARD Tucker models. The cores are sorted according to the norm of each slab in each direction, i.e. such that $\|\mathbf{X}_{(1,:,:)}\|_F^{\geq}\|\mathbf{X}_{(2,:,:)}\|_F^2 > ... > \|\mathbf{X}_{(J_3,:,:)}\|_F^2$, $\|\mathbf{X}_{(:,1,:)}\|_F^2 > \|\mathbf{X}_{(:,2,:)}\|_F^2 > ... > \|\mathbf{X}_{(:,J_2,:)}\|_F^2$, $\|\mathbf{X}_{(:,:,1)}\|_F^2 > \|\mathbf{X}_{(:,:,2)}\|_F^2 > ... > \|\mathbf{X}_{(:,:,J_3)}\|_F^2$. Clearly, regularization has removed excess components and reduced the non-zero elements in the core. However, the choice of regularization, i.e. sparse (l_1-regularization) or ridge (l_2-regularization), seem to only have limited effect on the estimated model order as well as the structure of the cores.

In table 2 and 3 is given the result when estimating the number of components of the Tucker model based on DIFFIT, BIC and AIC and in table 4 the results obtained using the NumConvHull approach. For the synthetic data both DIFFIT, NumConvHull and BIC correctly identifies the Tucker$(3, 4, 5)$ component model whereas AIC fails and overestimates the number of components. In The Flow Injection analysis data DIFFIT underestimated the number of components indicating a Tucker$(1, 1, 1)$ whereas NumConvHull as for the sparse ARD approach indicate a Tucker$(3, 4, 2)$ model. Both

Synthetic Data

(J_1, J_2, J_3)	$\sharp_{Sparse}$	$\log P_{Sparse}$	$\sharp_{Ridge}$	$\log P_{Ridge}$
$(3, 4, 5)$	20	$\mathbf{-1.4007 \cdot 10^5}$	18	$\mathbf{-1.3709 \cdot 10^5}$
$(10, 10, 10)$	0	-	2	$-1.4361 \cdot 10^5$

Flow Injection Analysis

(J_1, J_2, J_3)	$\sharp_{Sparse}$	$\log P_{Sparse}$	$\sharp_{Ridge}$	$\log P_{Ridge}$
$(3, 4, 2)$	9	$\mathbf{2.9938 \cdot 10^5}$	1	$2.8122 \cdot 10^5$
$(3, 4, 3)$	7	$2.9858 \cdot 10^5$	6	$\mathbf{2.8273 \cdot 10^5}$
$(3, 5, 3)$	0	-	11	$2.8181 \cdot 10^5$
$(4, 4, 3)$	4	$2.9861 \cdot 10^5$	2	$2.8170 \cdot 10^5$

Amino Acid Fluorescence Analysis

(J_1, J_2, J_3)	$\sharp_{Sparse}$	$\log P_{Sparse}$	$\sharp_{Ridge}$	$\log P_{Ridge}$
$(3, 3, 3)$	17	$\mathbf{-2.5500 \cdot 10^5}$	18	$\mathbf{-2.7767 \cdot 10^5}$
$(4, 3, 3)$	2	$-2.5535 \cdot 10^5$	2	$-2.7799 \cdot 10^5$
$(5, 3, 3)$	2	$-2.5639 \cdot 10^5$	0	-

Sugar Process

(J_1, J_2, J_3)	$\sharp_{Sparse}$	$\log P_{Sparse}$	$\sharp_{Ridge}$	$\log P_{Ridge}$
$(1, 4, 4)$	10	$\mathbf{-3.2450 \cdot 10^5}$	11	$\mathbf{-3.5894 \cdot 10^5}$
$(1, 4, 5)$	7	$-3.2452 \cdot 10^5$	9	$-3.5908 \cdot 10^5$
$(1, 4, 6)$	2	$-3.2495 \cdot 10^5$	0	-
$(2, 4, 4)$	1	$-3.3080 \cdot 10^5$	0	-

Dorrit

(J_1, J_2, J_3)	$\sharp_{Sparse}$	$\log P_{Sparse}$	$\sharp_{Ridge}$	$\log P_{Ridge}$
$(8, 4, 6)$	4	$\mathbf{-1.5272 \cdot 10^6}$	2	$\mathbf{-1.5720 \cdot 10^6}$

Table 1: ARD Tucker analysis based on sparse ARD Tucker and ridge ARD Tucker of the Synthetic Data, Flow Injection analysis data, Amino Acid Fluorescence data, Sugar Process data and Dorrit data. 20 models based on a Tucker$(10, 10, 10)$ component model were fitted for each data set. Given are the estimated models, number of models estimated and the likelihood of the best of each model estimated. The best model is given by the model with largest $\log P$ value indicated in bold. For the Dorrit data we have only given the best of the 20 estimated models.

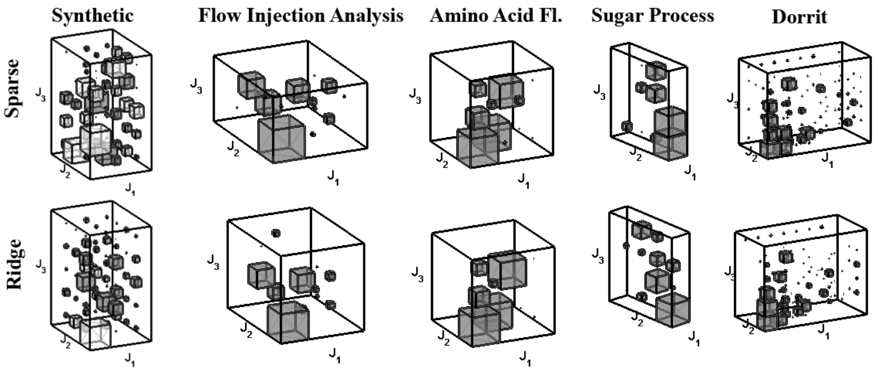

Figure 3: The estimated cores for the five datasets based on sparse ARD Tucker top row and ridge ARD Tucker bottom row. Gray boxes correspond to positive values and white to negative values in the core. The size of the boxes indicate the amplitude of each entry in the core.

AIC and BIC wrongfully estimates a Tucker$(4, 5, 5)$ model. For the Amino Acid Fluorescence DIFFIT and NumConvHull correctly indicate a Tucker(3,3,3) model whereas both BIC and AIC overestimates the number of components indicating a Tucker$(5, 4, 4)$ model. For the Sugar Process both DIFFIT and NumConvHull underestimates the number of components indicating a Tucker$(1, 1, 1)$ and Tucker$(1, 2, 2)$ respectively and for the Dorrit data DIFFIT again underestimates the model order indicating a Tucker$(2, 2, 1)$ model whereas NumConvHull indicates a Tucker$(5, 3, 4)$ model correctly identifying the number of components of the third mode. BIC and AIC overestimates the number of components for the Sugar Process and Dorrit data as both approaches indicate that the largest estimated Tucker$(5, 5, 5)$ models are the most appropriate.

3.3 CP ARD analysis

We finally analyzed the three data set with CP structure by fixing the core in the ARD Tucker to be diagonal ($\mathcal{G} = \mathcal{I}$). In figure 4 the estimated number of components for the three data sets can be found using the CorConDiag, DIFFIT/NumConvHull, BIC, AIC and sparse as well as ridge ARD CP. Notice how both BIC and AIC as for the Tucker analysis fail in estimating the adequate number of components. This is because the Tucker model and in particular the CP model are highly restricted models using only a few parameters to model a large amount of data. Thus, the complexity terms in BIC and AIC grows in general more slowly than the improvement in $\log(\mathrm{SSE})$ thus they tend to favor too complex models. The Core Consistency Diagnostic correctly identifies 3 components in the amino acid fluorescence data, 3-4 components in the sugar process data and 4 components in the Dorrit fluorescence data. The DIFFIT and NumConvHull correctly indicates a 3 component model for the amino acid fluorescence

Synthetic data

m	(J_1, J_2, J_3)	VarExp	DIF	DIFFIT	BIC	AIC
3	(1 , 1 , 1)	0.15389	0.15389	1.40732	278900	277690
5	(1 , 2 , 2)	0.26323	0.10935	4.14367	271620	269480
6	(2 , 2 , 2)	0.28962	0.02639	0.53081	269806	267325
7	(2 , 2 , 3)	0.33934	0.04971	1.91710	266047	263026
8	(2 , 3 , 3)	0.36527	0.02593	0.72668	264150	260669
9	(2 , 3 , 4)	0.40096	0.03569	0.70940	261294	257254
10	(3 , 3 , 4)	0.45126	0.05031	1.21855	256494	252033
11	(3 , 4 , 4)	0.49254	0.04128	3.41348	252373	247392
12	(3 , 4 , 5)	0.50464	0.01209	**17.01277**	**251608**	246007
13	(4 , 4 , 5)	0.50535	0.00071	0.86549	252072	245971
14	(4 , 5 , 5)	0.50617	0.00082	0.99170	252632	245931
15	(5 , 5 , 5)	0.50700	0.00083	-	253137	**245885**

Flow Injection Analysis

m	(J_1, J_2, J_3)	VarExp	DIF	DIFFIT	BIC	AIC
3	(1 , 1 , 1)	0.84522	0.84522	**10.06226**	-641869	-644006
5	(1 , 2 , 2)	0.92922	0.08400	4.14987	-723206	-727374
6	(2 , 2 , 2)	0.94946	0.02024	0.86437	-758993	-763330
7	(2 , 3 , 2)	0.97287	0.02342	1.30707	-824256	-829693
8	(2 , 4 , 2)	0.99079	0.01792	3.85675	-938422	-944960
9	(3 , 4 , 2)	0.99544	0.00465	2.15369	-1013167	-1019916
10	(3 , 4 , 3)	0.99759	0.00216	1.58609	-1080327	-1088145
11	(3 , 5 , 3)	0.99895	0.00136	6.96039	-1167958	-1176929
12	(4 , 5 , 3)	0.99915	0.00020	1.05747	-1189701	-1198957
13	(4 , 5 , 4)	0.99933	0.00018	4.61202	-1214552	-1224962
14	(4 , 5 , 5)	0.99937	0.00004	-	**-1219905**	**-1231468**
15	(5 , 5 , 5)	0.99937	0.00000	-	-1219477	-1231431

Amino Acid Fluoresence

m	(J_1, J_2, J_3)	VarExp	Dif	DIFFIT	BIC	AIC
3	(1 , 1 , 1)	0.64390	0.64390	3.23061	585439	582752
5	(2 , 2 , 1)	0.84321	0.19931	8.60608	537453	532672
6	(2 , 2 , 2)	0.86637	0.02316	0.44808	528371	522938
7	(3 , 2 , 2)	0.91806	0.05169	0.74407	498490	492967
8	(3 , 3 , 2)	0.98752	0.06946	5.85020	385395	377797
9	(3 , 3 , 3)	0.99940	0.01187	**875.40179**	200603	192304
10	(3 , 3 , 4)	0.99941	0.00001	0.10839	199984	190983
11	(4 , 4 , 3)	0.99953	0.00013	2.61932	187135	176561
12	(4 , 3 , 5)	0.99958	0.00005	0.95627	179767	169864
13	(5 , 4 , 4)	0.99963	0.00005	-	**173789**	**162232**
14	(5 , 5 , 4)	0.99963	0.00000	-	176225	162453
15	(5 , 5 , 5)	0.99963	0.00000	-	177173	162539

Table 2: DIFFIT, BIC and AIC analysis of the Synthetic data, Flow Injection analysis data and Amino Acid Fluorescence data. All combinations of models up to a Tucker$(5, 5, 5)$ component model were evaluated (for the analysis of the Flow Injection Analysis data we also included the Tucker$(3, 6, 4)$ model which turned out to be less optimal than the Tucker$(4, 5, 4)$ model). Given are the best models obtained from 3 runs.

Sugar Process

m	(J_1, J_2, J_3)	VarExp	Dif	DIFFIT	BIC	AIC
3	(1 , 1 , 1)	0.94985	0.94985	**26.28563**	591595	588138
5	(1 , 2 , 2)	0.98599	0.03614	10.66366	465543	461424
6	(2 , 2 , 2)	0.98938	0.00339	1.89336	441145	434168
7	(1 , 3 , 3)	0.99117	0.00179	0.52446	420419	415618
8	(2 , 3 , 3)	0.99458	0.00341	3.38900	375058	367346
9	(3 , 3 , 3)	0.99559	0.00101	0.68470	357814	347191
10	(2 , 4 , 4)	0.99706	0.00147	0.84127	315187	306697
11	(3 , 4 , 4)	0.99880	0.00175	5.86729	228849	217375
12	(4 , 4 , 4)	0.99910	0.00030	1.14055	203609	189151
13	(5 , 4 , 4)	0.99936	0.00026	2.03311	172664	155222
14	(5 , 5 , 4)	0.99949	0.00013	1.39400	151068	132859
15	(5 , 5 , 5)	0.99958	0.00009	-	**131511**	**112965**

Dorrit

m	(J_1, J_2, J_3)	VarExp	Dif	DIFFIT	BIC	AIC
3	(1 , 1 , 1)	0.61094	0.61094	4.61661	3143497	3136390
5	(2 , 2 , 1)	0.74328	0.13234	**5.07438**	3002488	2988534
6	(2 , 2 , 2)	0.76936	0.02608	0.59456	2964598	2950314
7	(3 , 3 , 1)	0.81322	0.04386	0.78720	2896379	2875554
8	(3 , 3 , 2)	0.86894	0.05572	4.44743	2770309	2749095
9	(3 , 3 , 3)	0.88147	0.01253	1.02330	2734856	2713253
10	(4 , 3 , 3)	0.89371	0.01224	1.10994	2696390	2674362
11	(5 , 3 , 3)	0.90474	0.01103	1.43948	2657729	2635277
12	(5 , 4 , 3)	0.91240	0.00766	1.54004	2635022	2605900
13	(4 , 4 , 5)	0.91738	0.00498	0.62549	2614666	2585060
14	(5 , 4 , 5)	0.92534	0.00796	0.90730	2579119	2548960
15	(5 , 5 , 5)	0.93410	0.00877	-	**2541883**	**2504935**

Table 3: DIFFIT, BIC and AIC analysis of the Sugar Process data and Dorrit data. All combinations of models up to a Tucker$(5, 5, 5)$ component model was evaluated. The best models obtained from 3 runs are given.

Synthetic data

FP	(J_1, J_2, J_3)	NumConvHull
205	(1 , 2 , 2)	1.32121
285	(2 , 2 , 3)	1.06799
457	(3 , 4 , 4)	3.90349
510	(3 , 4 , 5)	**13.53390**

Flow Injection Analysis

FP	(J_1, J_2, J_3)	NumConvHull
398	(2 , 2 , 2)	2.14066
510	(3 , 3 , 2)	1.30439
609	(3 , 4 , 2)	**8.34957**
705	(3 , 4 , 3)	1.65218
805	(3 , 5 , 3)	1.39208
825	(4 , 5 , 3)	2.61592
843	(5 , 5 , 3)	2.66880
927	(4 , 5 , 4)	3.49294

Claus

FP	(J_1, J_2, J_3)	NumConvHull
534	(3 , 2 , 2)	2.96373
736	(3 , 3 , 2)	1.88249
801	(3 , 3 , 3)	**19.22182**
813	(5 , 3 , 3)	10.73038

Sugar

FP	(J_1, J_2, J_3)	NumConvHull
383	(1 , 2 , 2)	**6.73324**
438	(1 , 3 , 3)	2.19296
491	(1 , 4 , 4)	3.33768
772	(2 , 4 , 4)	2.05301
1051	(3 , 4 , 4)	1.90200
1061	(3 , 4 , 5)	2.73030
1406	(4 , 5 , 5)	1.03639

Dorrit

FP	(J_1, J_2, J_3)	NumConvHull
648	(2 , 1 , 2)	1.23638
1301	(5 , 2 , 3)	1.01415
1328	(5 , 2 , 4)	1.75568
1862	(5 , 3 , 3)	1.71240
1894	(5 , 3 , 4)	**3.15220**
2493	(5 , 4 , 5)	1.90288

Table 4: Numerical convex hull analysis of the five dataset. Given are the degrees of freedom, corresponding models and largest NumConvHull values obtained.

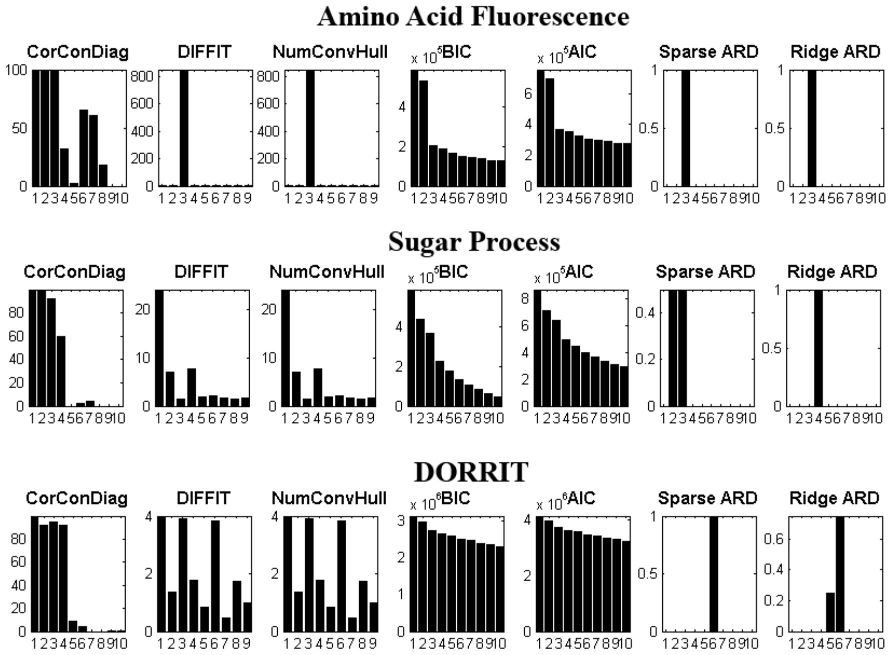

Figure 4: Model order estimation for the amino acid fluorescence data, sugar process data and Dorrit data. To the left is given the Core Consistency Diagnostic (CorConDiag), DIFFIT, NumConvHull, BIC and AIC (To avoid local minima, the best of three decompositions was evaluated). To the right is given the distribution of models from the estimation of 20 different ARD CP models initialized with 10 components based on Laplace priors (sparse ARD) and Gaussian priors (ridge ARD).

data but wrongfully a 1 component model for the Sugar process and a 1,3 or 6 component model for the Dorrit data. Both the sparse and ridge ARD methods correctly identified 3 components in the amino acid fluorescence data, for the sugar process the sparse ARD indicate a 2 or 3 component model whereas the ridge ARD correctly identifies a 4 component model. For the Dorrit data both sparse and ridge ARD indicate a 6 component model. Thus, while the proposed ARD approach here perform better than heuristics such as DIFFIT/NumConvHull, BIC and AIC the Core Consistency Diagnostic seem to work somewhat better in estimating the number of components in the CP model.

4 Discussion

Model selection is perhaps one of the most challenging problems in unsupervised learning. We demonstrated how a simple Bayesian framework based on Automatic Relevance Determination (ARD) could be adapted to multi-way models such as the Tucker and CP models. The proposed ARD framework forms an efficient tool for the automatic estimation of components in the Tucker models and as we saw in the analysis of both synthetic and real data the method indeed effectively extracted reasonable number of components. While the ARD approach outperforms heuristics such as DIFFIT, NumConvHull, AIC and BIC when estimating the number of components in the Tucker and CP model we found that the Core Consistency Diagnostic performed slightly better in estimating the adequate number of components in the CP model. The modelling inadequacies encountered for the ARD CP and Tucker is probably due to incorrect estimates of the SNR, deviation from Gaussianity in the noise, deviation from Gaussian and Laplace distributed components, the fact that the parameters were based on simple MAP estimates and finally due to limited amount of data for the identification of the model order. The reason why no approach correctly established the Tucker$(3, 6, 4)$ structure of the flow injection analysis data and Tucker$(4, 4, 4)$ of the sugar process data is because models with less components almost perfectly accounts for all data (VarExp> 0.99) as seen in table 2 and 3. On the other hand, for the Dorrit data the sparse ARD and ridge ARD failed in correctly identifying 4 components as excess components were able to model substantial parts of the data.

Despite the different nature of the Gaussian and Laplace priors the results found based on the two priors were similar. This is because the ARD framework first and foremost turn off excess components while components that remain active are little influenced by the prior if their parameters are large. Hence, if the d^{th} component of the n^{th} mode is important then $\alpha_d^{(n)}$ will be small rendering the prior noninformative and as a result give little effect in the estimation of that component. Thus, while the ARD framework effectively can turn off excess components the choice of prior seems to only have a limited effect on the components identified. Rather than estimating σ^2 from data we defined σ^2 from a user given signal to noise ratio (SNR). In figure 2 we saw that the results obtained was only to a small degree sensitive to the defined SNR as long as the SNR was not set to high causing the model to overfit the data. Hence, although this parameter is user defined the actual choice of the parameter only has a limited impact on the models obtained.

The ARD approach is computationally inexpensive as the method automatically removes excess components when estimating the model contrary to existing heuristics that requires the estimation and evaluation of all potential models. Thus, the ARD is a simple yet efficient tool for the evaluation of the number of components of multi-way models. Presently, each component of each mode was given its own prior and the priors were either solely Laplace or Gaussian, however, we note that other parameterizations of the priors are conceivable. Furthermore, we considered the most simple framework where loadings and hyper-parameters were based on maximum a posteriori (MAP) estimation. Within the proposed Bayesian framework more involved methods based on sampling approaches to estimate model parameters (5; 6) as well as expectation propagation for the evaluation of predictive performance (31) can be employed to further improve the model order estimation. This should be investigated in future work.

Finally, the ARD approach can only shrink models, i.e. remove components. Thus, once a component has been removed it can no longer be brought back. In particular this requires that J_n be chosen large enough to encompass all potential models. Future research should investigate methods that can adapt the ARD approach to grow if initialized by a model order that is too small.

5 Acknowledgement

We would like to thank the anonymous reviewers for helpful comments. This research was supported by the European Commission through the EU FP6 NEST Pathfinder grant PERCEPT (043261).

References

[1] E. Acar, S. A. Camtepe, M. S. Krishnamoorthy, and B. Yener. Modeling and multiway analysis of chatroom tensors. *Intelligence and Security Informatics, Lecture Notes in Computer Science*, 3495:256–268, 2005.

[2] H. Akaike. A new look at the statistical model identification. *Automatic Control, IEEE Transactions on*, 19(6):716–723, 1974.

[3] A. H. Andersen and W. S. Rayens. Structure-seeking multilinear methods for the analysis of fmri data. *NeuroImage*, 22:728–739, 2004.

[4] C. A. Andersson and R. Bro. Improving the speed of multi-way algorithms: Part I. Tucker3. *Chemometrics and Intelligent Laboratory Systems*, 42:93–103, 1998.

[5] M.J. Beal. *Variational Algorithms for Approximate Bayesian Inference*. PhD thesis, University of London, 2003.

[6] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, August 2006.

[7] R. Bro. Exploratory study of sugar production using fluorescence spectroscopy and multi-way analysis. *Chemom. Intell. Lab. Syst.*, 46:133–147, 1999.

[8] R. Bro, K. Kjeldahl, Age K. Smilde, and H. A. L. Kiers. Cross-validation of component models: A critical look at current methods. *Anal Bioanal Chem*, 390(5):1241–1251, March 2008.

[9] R. Bro. Parafac: Tutorial and applications. *Chemometrics and Intelligent Laboratory Systems*, 38:149–171, 1997.

[10] R. Bro and H. A. L. Kiers. A new efficient method for determining the number of components in parafac models. *Journal of Chemometrics*, 17(5):274–286, 2003.

[11] J. D. Carroll and J. J. Chang. Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition. *Psychometrika*, 35:283–319, 1970.

[12] E. Ceulemans and H. A. L. Kiers. Selecting among three-mode principal component models of different types and complexities : A numerical convex hull based method. *British journal of mathematical and statistical psychology*, 59(1):133–150, 2006.

[13] L. de Lathauwer, Bart de Moor, and J. Vandewalle. On the best rank-1 and rank-(r1,r2, . . . , rn) approximation of higher-order tensors. *SIAM J. MATRIX ANAL. APPL.*, 21(4):1324–1342, 2000.

[14] D. 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.

[15] L. K. Hansen, K. H. Madsen, and T. Lehn-Schiøler. Adaptive regularization of noisy linear inverse problems. In *Proceedings of Eusipco 2006*, 2006.

[16] 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.

[17] R. A. Harshman and M. E. Lundy. Data preprocessing and the extended parafac model. *In: Law, H. G., Snyder, Jr., C. W., Hattie, J. A., and McDonald, R. P. (eds.), Research Methods for Multimode Data Analysis, Praeger, New York,*, pages 216–281, 1984.

[18] H. F. Kaiser. The varimax criterion for analytic rotation in factor analysis. *Psychometrica*, 23:187–200, 1958.

[19] H. A. L. Kiers and A. der Kinderen. A fast method for choosing the numbers of components in tucker3 analysis. *British Journal of Mathematical and Statistical Psychology*, 56:119–125, 2003.

[20] H. 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.

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

[22] 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.

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

[24] D. J. C. Mackay. Bayesian interpolation. *Neural Computation*, 4:415–447, 1992.

[25] M. Mørup, L. K. Hansen, C. S. Hermann, J. Parnas, and S. M. Arnfred. Parallel factor analysis as an exploratory tool for wavelet transformed event-related eeg. *NeuroImage*, 29(3):938–947, 2006.

[26] M. Mørup, L.K. Hansen, and S. M. Arnfred. Algorithms for sparse non-negative Tucker. *Neural Computation*, 20(8): 2112–2131, 2008.

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

[28] T. Murakami and P. M. Kroonenberg. Three-mode models and individual differences in semantic differential data. *Multivariate Behavioral Research*, 38(2):247–283, 2003.

[29] 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.

[30] L. Omberg, G. H. Golub, and O. Alter. A tensor higher-order singular value decomposition for integrative analysis of dna microarray data from different studies. *Proceedings of the National Academy of Science (PNAS)*, 104(47):18371–18376, 2007.

[31] Y. (Alan) Qi, T. P. Minka, R. W. Picard, and Z. 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.

[32] J. 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.

[33] G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6(2):461–464, 1978.

[34] N.D. Sidiropoulos, R. Bro, and G.B. Giannakis. Parallel factor analysis in sensor array processing. *Signal Processing, IEEE Transactions on*, 48(8):2377–2388, 2000.

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

[36] S. C. Strother, J. Anderson, L. K. Hansen, U. Kjems, R. Kustra, J. Sidtis, S. Frutiger, S. Muley, S. LaConte, and D. Rottenberg. The quantitative evaluation of functional neuroimaging experiments: The npairs data analysis framework. *NeuroImage*, 15(4):747–771, 2002.

[37] M.E. Tibbing. Sparse bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1:211–244, 2001.

[38] M. E. Timmerman and H. A. L. Kiers. Three-mode principal components analysis: Choosing the numbers of components and sensitivity to local optima. *British Journal of Mathematical and Statistical Psychology*, 53:1–16, 2000.

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

[40] M. A. O. Vasilescu and D. Terzopoulos. Multilinear analysis of image ensembles: Tensorfaces. In *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part I*, pages 447–460, London, UK, 2002. Springer-Verlag.