Bayesian Methods for Multiway Modeling and Online Hierarchical Clustering

Jørgensen, Philip Johan Havemann

Publication date:
2022

Document Version
Publisher's PDF, also known as Version of record

Bayesian Methods for Multiway Modeling and Online Hierarchical Clustering

Philip Johan Havemann Jørgensen

Kongens Lyngby 2021

Technical University of Denmark
Department of Applied Mathematics and Computer Science
Richard Petersens Plads, building 324,
2800 Kongens Lyngby, Denmark
Phone +45 4525 3031
compute@compute.dtu.dk
www.compute.dtu.dk

Summary (English)

This thesis presents an investigation and development of a number of unsupervised machine learning algorithms using Bayesian statistics. It is still demanding to apply many machine learning algorithms with one reason being that they often require the user to provide expert knowledge, such as a ground truth obtained from the result of an experiment or from querying an expert. As more processes become documented by data collection, more opportunities arise with a growing demand for machine learning algorithms that can learn in an unsupervised manner without requiring the ground truth. Here, Bayesian statistics can provide ways to exploit this potential created by the extensive data collection.

This work includes three papers on unsupervised probabilistic machine learning methods using Bayesian inference. The probabilistic approaches allow for ways of automatically inferring the model complexity required to describe the analyzed data instead of requiring the user to determine it manually. The first paper investigates the benefit of using a probabilistic framework for a multiway decomposition method known as PARAFAC2 commonly applied for dealing with chromatographic data. The second paper proposes approaches for using the same probabilistic framework for the problem of binary or one-class classification as encountered in food authentication tasks. The third paper proposes an online algorithm for learning a clustering on a data stream based on a model known as Bayesian Hierarchical Clustering. Throughout these papers, the power of the probabilistic methods is demonstrated under these circumstances on synthetic and real data.

Summary (Danish)

Denne afhandling præsenterer en undersøgelse og udvikling af en række ikke-superviserede maskinlæringsalgoritmer, som bygger på Bayesiansk statistik. Det er stadig krævende at anvende maskinlæringsalgoritmer, da de ofte kræver, at en bruger forsyner algoritmen med ekspertviden, såsom en underliggende sandhed bestemt fra resultatet af et forsøg, eller svaret fra en ekspert. Som følge af, at stadig flere processer bliver dokumenteret gennem dataindsamling, opstår samtidig et større behov for maskinlæringsalgoritmer, der kan lære uden af have den underliggende sandhed tilgængelig. Her gør Bayeseiansk statistik det muligt at udnytte det potentiale, der findes i de store mængder data der genereres.

Dette arbejde indeholder tre artikler omhandlende ikke-superviserede probabilistiske maskinlæringsalgoritmer, som anvender Bayesiansk inferens. De probabilistike metoder tillader automatisk estimering af modelkompleksiteten, der er nødvendig for at beskrive den analyserede data, i stedet for at antage at en bruger vil kunne bestemme den manuelt. Den første artikel undersøger fordelene, ved at bruge et probabilistisk framework til en multivejsdekompositionsmetode kendt som PARAFAC2, som normalt bruges til kromatografisk data. Det andet paper foreslår en fremgangsmåde, som bruger det samme probabilistiske framework til at løse et binært eller en-klasse klassifikations problem, som kan opstå ved opgaver såsom godkendelse af fødevarer. Det tredje paper foreslår en nyskabende online algoritme til at finde en gruppering af en strøm af data baseret på probabilistiske modelsammenligninger. Gennem disse artikler vises styrken af de probabilistike metoder i disse sammenhænge på syntetisk og rigtig data.

Preface

This thesis was prepared partly at Section for Statistics and Data Analysis and at Section of Cognitive Systems, both belonging to the Department of Applied Mathematics and Computer Science, Technical University of Denmark in fulfillment of the requirements for acquiring a Ph.D. degree in engineering. The project has been part of the Danish Center for Big Data Analytics driven Innovation funded by Innovation Fund Denmark aiming to advance the Danish society to be at the forefront of capitalizing on the potential of data science and big data. The main supervisor was professor Bjarne Kjær Ersbøll, and the co-supervisor was professor Lars Kai Hansen — both at the Department of Applied Mathematics and Computer Science, Technical University of Denmark.

The thesis consists of three papers disseminating the work carried out between January 2017 and July 2021.

Lyngby, 26-July-2021

Philip Johan Havemann Jørgensen

Acknowledgments

I would like to thank my supervisors, Bjarne and Lars, for all their support, guidance and patience, without which the completion of this thesis would have been impossible; my Danish co-authors Søren, Mikkel, Jesper, Kristoffer and, lastly, Morten Mørup who especially spent a lot of time on our many talks and has provided invaluable feedback for this thesis; my great colleagues at DTU Compute, with whom I have shared offices, coffees and many laughs; Amelie Sina Wilde at the National Food Institute, DTU, for a collaboration with many inspiring talks; Tom Heskes and Jesse H. Krijthe for welcoming me to the research stay at Radboud University, Nijmegen, including our collaboration, with the same being true for the whole of the Data Science group at Radboud University; and Innovation Fund Denmark for funding the project.

Finally, I would like to thank my family and friends for all of their support and patience during this period of my life, and especially my significant other, Nanna, who has given me more encouragement than she knows.

Contents

Chapter 1

Introduction

Unsupervised learning is a group of machine learning algorithms able to identify patterns found in data without the use of annotations(Ghahramani, 2004). This thesis presents research on two tasks encountered in unsupervised learning tackled with Bayesian statistics. All the methods described are general in the sense that they are not limited to the applications considered here. Hopefully, the fundamental research in the underlying methodology will find a broad range of applications in numerous domains.

The first task considered in this work is how to decompose a set of raw observations into structured components. Such components can either represent an intrinsic latent structure or be used to perform dimensionality reduction on the data, or both simultaneously. Probably the most well-known method for this is the principal component analysis (PCA) which models bilinear data as found in matrices by identifying an ordered orthonormal basis based on maximizing the variance. Such matrices might represent a single experiment where a set of J features have been recorded for I time points. In cases where the experiment is repeated K times, it can be beneficial to represent them as a collection of matrices known as a tensor or multiway array. Such a representation allows models to capture the variance across experiments where, depending on the type of data, each experiment could be defined by varying experimental conditions, different samples or different subjects. Multiway models are applied to analyze such data taking advantage of the multilinear relationships found in the data(Kroonenberg, 2007; Acar & Yener, 2009; Mørup, 2011). Alternatively, the data can be flattened along one or more dimensions (e.g., by summation or averaging) to obtain a matrix that can be analyzed using multivariate methods. An important multiway model known as PARAFAC2 is here presented and investigated in the context of a probabilistic framework. Multiway data is only becoming more prominent as the development of data capturing technologies makes it cheaper and easier to record a multiway structure.

The second type of unsupervised learning problem considered is known as clustering(R. Xu & Wunsch, 2005; Murtagh & Contreras, 2012). Clustering methods can determine some meaningful grouping of the entities in the data. Often it is assumed that a meaningful clustering is one where grouped elements are closer together than elements outside the group. This could be for an exploratory analysis creating an overview of the data, a predictive analysis evaluating new data points as more data is collected, or even dimensionality reduction of the data. The clusters are determined by an algorithm using some notion of similarity or dissimilarity. The clustering could be determined from greedily building a hierarchy of the data based on their Euclidean distances, or by specifying that K clusters should each be represented by their mean. In both cases, the most meaningful number of clusters given the data is uncertain, which might also change later if new data are collected. So, evaluating the meaningfulness of the clustering algorithm depends both on the notion of distance used and the number of clusters. In this work, we look at a probabilistic hierarchical clustering algorithm able to use different distributions to describe the data, which is inherently able to determine the number of clusters in a data-driven manner. A new algorithm is developed to allow this model to work when data is collected sequentially.

The alternative to such methods would be to use supervised methods on annotated data. Obtaining annotated data involves considerations on what it will cost. Often it will require some expert or experiment to provide the annotation — both of which might be costly to employ in time or monetary value. The interest in trying to limit these costs is evident from the number of subfields dedicated to methods aiming to do so: *active learning*(Settles, 2009) tries to selectively query the most informative samples, *semi-supervised learning*(van Engelen & Hoos, 2020) tries to integrate information from data without annotations to solve supervised problems, *transfer learning*(Pan & Yang, 2010; Patricia & Caputo, 2014) tries to use information previously learned on an annotated tasks to improve the results on a new task, *positive-unlabeled learning*(Elkan & Noto, 2008) tries to determine which unlabeled samples should be considered positive and which sample should be considered negative from annotations of positive samples alone.

Unsupervised learning is the only subfield not requiring any annotations out of all of these approaches. Comparing the results of an unsupervised method to a supervised one on the same task — where the annotations are simply unaccounted for by the unsupervised method — will be expected to be in favor of the latter, as it incorporates more information from the data. However, unsupervised learning or some variant of the above-mentioned approaches are advantageous for tasks where annotations remain expensive or next-to-impossible to obtain. In case no meaningful annotations can be obtained, unsupervised learning might be the only solution.

In addition to the fully unsupervised tasks considered, an application of one-class classification is investigated in the context of the probabilistic PARAFAC2. Anomaly, novelty or outlier detection are concerned with learning tasks trying to detect abnormal data points, like fraudulent behavior, fault diagnosis of mechanical or digital processes or novel results in scientific experiments(Hodge & Austin, 2004). Such learning tasks can be either supervised, unsupervised or somewhere in between, depending on the exact application and availability of annotations and data. One-class classification(Minter, 1975; Tax, 2002) is a special case of such learning problems which aims to discriminate between instances of a specific class of interest and instances belonging to any other class. For this type of learning, it is often assumed that the available data only includes examples of the class of interest.

The main approach taken in this thesis to solve these different problems is based on Bayesian statistics inferring an approximate probability distribution of the model parameters. This is usually more computationally demanding than obtaining point estimates of the parameters, but provides elegant solutions to problems such as missing data, overfitting, determining model complexity, comparing models and making predictions(Bishop, 2006; Gelman, 2008). Especially, the ability to adjust the model complexity to the data is important for the unsupervised settings. Here, it can assist in determining the number of components in a decomposition or clustering model as well as to adapt a clustering model to new or changing concepts.

1.1 Research Contributions

This thesis contributes to advancing Bayesian inference for multiway modeling and clustering by investigating the benefits of using the probabilistic PARAFAC2 model and its estimated model for downstream tasks such as one-class classification, as well as how Bayesian Hierarchical Clustering can be performed in an online setting. These contributions come from answering the following research questions:

(a) What are the merits of the probabilistic PARAFAC2 as opposed to conventional PARAFAC2 modeling not accounting for uncertainty?

(b) How does the probabilistic PARAFAC2 model compare to the conventional PARAFAC2 and methods for flattening multiway data as a preprocessing step for downstream tasks?

(c) How can the probabilistic PARAFAC2 model be used for one-class classification in the context of food authentication?

(d) How can Bayesian Hierarchical Clustering be advanced to an online setting where data arrives sequentially?

(e) How can an online algorithm for Bayesian Hierarchical Clustering be scaled?

The answers to these questions will be presented in the following and summarized and discussed in the conclusion. The remainder of this thesis is arranged into three main parts. The second chapter reviews the fundamental theory underlying the research in the three papers included in the thesis. The third chapter describes the work on the probabilistic PARAFAC2 model covered by two of the papers, where the first paper introduces the probabilistic framework and investigates its basic properties and benefits, while the second dives into applying the model for food authentication by performing feature extraction as well as one-class modeling. The fourth chapter describes the work presented in the third and final paper introducing an online algorithm for the Bayesian Hierarchical Clustering model. The titles of these three papers included in this thesis are as follows:

(i) *Probabilistic PARAFAC2* (In preparation for resubmission)

(ii) *Probabilistic PARAFAC2 for Food Authentication* (In preparation for submission)

(iii) *Online Bayesian Hierarchical Clustering* (In preparation for resubmission)

Chapter 2

Theory

In this chapter, we review the theoretical background for the work in this thesis. This includes Bayesian inference using conjugate priors, variational inference, unsupervised learning in the form of decomposition models and clustering, as well as one-class classification.

2.1 Probabilistic Machine Learning

Probabilistic machine learning, also known as Bayesian machine learning, is based on a framework quantifying the uncertainty of all unknown quantities such as model parameters and model predictions. The framework applies probability theory and probability distributions to express these uncertainties. The probabilistic approach inherently deals with issues such as model selection and overfitting, as we will see, while simultaneously being conceptually appealing.

In general, we specify a likelihood $p(D|\theta)$ which is a probability distribution over the observations D with parameters θ. This is commonly viewed as a function of θ with the observations considered to be fixed. Additionally, the framework requires us to specify our belief about which values of the parameters θ are sensible. This is stated by the prior $p(\theta|\alpha)$ which is another probability distribution with hyperparameters α. Combining the likelihood and prior results in the joint distribution of the observations and parameters following the product rule of probability: $p(D, \theta|\alpha) = p(D|\theta)p(\theta|\alpha)$. This joint distribution provides probabilities about the model parameters given a set of observations — considered to be fixed — based on our choice of model and, before having seen any data, our belief about its parameters. Assuming the specified probability distributions are suitable for the observations, the next step is to update our belief, or prior, about the parameters θ according to a set of observations. This is done by computing the

posterior $p(\theta|D,\alpha)$ given by Bayes' rule(Bayes, 1764)

$$p(\theta|D,\alpha) = \frac{p(D,\theta|\alpha)}{p(D|\alpha)}, \tag{2.1}$$

where the term in the denominator $p(D|\alpha)$ is given by marginalizing over the parameters, i.e. $p(D|\alpha) = \int p(D,\theta|\alpha)d\theta$, and commonly referred to as the evidence. The posterior is a new probability distribution over the parameters θ given the data D and hyperparameters α. After having computed the posterior, probability theory states how to answer several questions like "how well does the model explain the observations?", "what is a good estimate of the parameters?", "how certain are we about predictions on new observations for this type of data?", as we will see in a few moments.

The probabilistic approach to machine learning has gained much attention in the last decade or so(Gelman et al., 2013; Ghahramani, 2015; Murphy, 2012, 2021) — and with good reason. From a practical perspective, scientific questions can rarely be answered with a simple yes or no due to limitations in data sampling or understanding of the problem, so the ability to quantify the plausibility of states becomes very appealing, or even necessary, to find meaningful answers(Tipping, 2004). Also, probability theory from a Bayesian perspective can be accepted as a logical basis for dealing with uncertainty(Cox, 1946; Jaynes, 2012; Van Horn, 2003).

2.1.1 Bayesian Inference

To evaluate (2.1) we need to compute the evidence given by $p(D) = \int p(D|\theta)p(\theta)d\theta$, which often is intractable for non-trivial models. Notice, here and in the following, we have omitted explicitly writing the dependency on the hyperparameters α to enhance the readability of the expressions. Conjugate priors — a prior belonging to the same probability distribution family as the posterior — lead to closed-form solutions, but for non-conjugate priors approximation methods such as Markov Chain Monte Carlo sampling or variational inference may be required to estimate the posterior when it is otherwise intractable.

2.1.1.1 Conjugate Priors

In general, a prior $p(\theta)$ belonging to a family of distributions $\mathcal{P}$ is conjugate to a likelihood $p(D|\theta)$ if the resulting posterior $p(\theta|D)$ also belongs to $\mathcal{P}$ (Raiffa & Schlaifer, 1961; Gelman et al., 2013). Conjugate priors with the same functional form as the likelihood are referred to as natural conjugate priors, and these exist only for likelihoods belonging to the exponential families(Fink, 1997).

Computing the posterior for a conjugate prior has a closed-form solution given by its sufficient statistics. One simply has to compute these sufficient statistics from the samples and hyperparameters. Below we give an example of a conjugate prior for a multivariate normal likelihood with unknown mean and precision. The conjugate prior is the Normal-inverse-Wishart distribution. This model is employed in Paper iii.

Normal-inverse-Wishart Assuming our observations $D_n = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ are identical and independently distributed according to a multivariate normal distribution with unknown mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$, we have

$$p(\mathbf{x}_i|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \tag{2.2}$$

with the joint distribution being $p(D|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{i=1}^{n} p(\mathbf{x}_i|\boldsymbol{\mu}, \boldsymbol{\Sigma})$, as the observations are assumed independent and identically distributed given the mean and covariance. The posterior distribution of the mean and variance can be computed using Bayes' theorem

$$p(\boldsymbol{\mu}, \boldsymbol{\Sigma}|D_n) = \frac{p(D_n|\boldsymbol{\mu}, \boldsymbol{\Sigma})p(\boldsymbol{\mu}, \boldsymbol{\Sigma})}{p(D_n)}. \tag{2.3}$$

Using a Normal-inverse-Wishart prior here makes this computation straight-forward, as it is conjugate for the multivariate normal likelihood with unknown mean and variance. It is written as

$$p(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \sim \text{NIW}(\boldsymbol{\mu}_0, \lambda, \boldsymbol{\Psi}, \nu), \tag{2.4}$$

where the probability density function of this distribution is

$$f(\boldsymbol{\mu}, \boldsymbol{\Sigma}|\boldsymbol{\mu}_0, \lambda, \boldsymbol{\Psi}, \nu) = \mathcal{N}(\boldsymbol{\mu}|\boldsymbol{\mu}_0, \lambda^{-1}\boldsymbol{\Sigma})\mathcal{W}^{-1}(\boldsymbol{\Sigma}|\boldsymbol{\Psi}, \nu). \tag{2.5}$$

Here, $\mathcal{W}^{-1}$ is the inverse Wishart distribution over positive-definite matrices parameterized by a scale matrix $\boldsymbol{\Psi}$ and ν degrees of freedom.

Since the prior is conjugate to the likelihood the posterior is also a Normal-inverse-Wishart distribution,

$$p(\boldsymbol{\mu}, \boldsymbol{\Sigma}|D_n) \sim \text{NIW}(\boldsymbol{\mu}', \lambda', \boldsymbol{\Psi}', \nu'), \tag{2.6}$$

with parameters given by(Murphy, 2007)

$$\begin{aligned}
\bar{D} &= \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i, \\
\boldsymbol{\mu}' &= \frac{\lambda\boldsymbol{\mu}_0 + n\bar{D}}{\lambda + n}, \\
\lambda' &= \lambda + n, \\
\nu' &= \nu + n, \\
\boldsymbol{\Psi}' &= \boldsymbol{\Psi} + \frac{\lambda n}{\lambda + n}(\bar{D} - \boldsymbol{\mu}_0)(\bar{D} - \boldsymbol{\mu}_0)^T + \sum_{i=1}^{n}(\bar{D} - \boldsymbol{\mu}_0)(\bar{D} - \boldsymbol{\mu}_0)^T.
\end{aligned}$$

So, we see that using a conjugate prior results in a closed-form solution for the posterior distribution with parameters being a function of the hyperparameters and data.

2.1.1.2 Approximate Inference

Specifying a conjugate prior was early on the only option to compute an intractable posterior, but the rise of computers has since made it possible to approximate it numerically(Fink, 1997). This is commonly being approached in one of two ways: by sampling from an approximation of

the posterior distribution or by optimizing a bound between the posterior and the approximation. In this work, we focus on the latter as it tends to be computationally faster than the sampling approaches making it more viable for larger data.

Variational Inference is a framework using optimization to approximate the posterior distribution(Jordan, Ghahramani, Jaakkola, & Saul, 1999; Blei, Kucukelbir, & McAuliffe, 2016). A search for the best approximation among a family of distributions $\mathcal{Q}$ is performed by solving the following optimization problem

$$q^\star(\theta) = \arg\min_{q\in\mathcal{Q}} \ \mathrm{KL}\big[q(\theta)\big\|p(\theta|D)\big]., \tag{2.7}$$

where KL$[\cdot]$ is the Kullback-Leibler (KL) divergence. The posterior $p(\theta|D,\alpha)$ still consists of the intractable evidence $p(D)$, but by expanding the KL divergence we get

$$\begin{aligned}
\mathrm{KL}\big[q(\theta)\big\|p(\theta|D)\big] &= \mathbb{E}[\log q(\theta)] - \mathbb{E}[\log p(\theta|D)], &(2.8)\\
&= \mathbb{E}[\log q(\theta)] - \mathbb{E}[\log p(\theta, D) - \log p(D)], &(2.9)\\
&= \mathbb{E}[\log q(\theta)] - \mathbb{E}[\log p(\theta, D)] + \log p(D), &(2.10)\\
& &(2.11)
\end{aligned}$$

which we can rearrange into an expression of the log evidence as

$$\log p(D) = \mathrm{KL}\big[q(\theta)\big\|p(\theta|D)\big] + \mathbb{E}[\log p(\theta, D)] - \mathbb{E}[\log q(\theta)], \tag{2.12}$$

and since KL$[\cdot] \geq 0$ the log evidence is lower bounded by the remaining terms which are suitably denoted the evidence lower bound (ELBO), i.e. $\mathrm{ELBO}(q) = \mathbb{E}[\log p(\theta, D)] - \mathbb{E}[\log q(\theta)]$. Maximizing the ELBO is equivalent to minimizing the KL divergence. Specifically, the problem in (2.7) is replaced by

$$q^\star(\theta) = \arg\max_{q(\theta)\in\mathcal{Q}} \ \mathrm{ELBO}(q). \tag{2.13}$$

Now, given a joint distribution $p(\theta, D)$ some variational family $\mathcal{Q}$ needs to be selected for the optimization problem to be fully specified. The complexity of the optimization problem depends on the choice of this variational family. A convenient choice is the family known as the mean-field family as it factorizes over the parameters by assuming they are mutually independent written as $q(\boldsymbol{\theta}) = \prod_j q_j(\boldsymbol{\theta}_j)$. This independence of the parameters simplifies the computational complexity of the optimization problem as illustrated by the following algorithm known as coordinate ascent variational inference(CAVI)(Blei et al., 2016; Bishop, 2006).

The CAVI algorithm iterates through each variational factor $q_j(\theta_j)$ updating them by maximizing the ELBO as a function of the j'th factor, thereby effectively climbing the full ELBO to a local optimum. Following the derivation by (Blei et al., 2016) writing the ELBO as a function of q_j looks like

$$\begin{aligned}
\mathrm{ELBO}(q_j) &= \mathbb{E}_j[\mathbb{E}_{-j}[\log p(\theta_j, \theta_{-j}, D)]] - \mathbb{E}_j[\mathbb{E}_{-j}[\log q_j(\theta_j) + \log q_{-j}(\theta_{-j})]],\\
&= \mathbb{E}_j[\mathbb{E}_{-j}[\log p(\theta_j, \theta_{-j}, D)]] - \mathbb{E}_j[\log q_j(\theta_j)] - \mathbb{E}_{-j}[\log q_{-j}(\theta_{-j})],\\
&= \mathbb{E}_j[\mathbb{E}_{-j}[\log p(\theta_j, \theta_{-j}, D)]] - \mathbb{E}_j[\log q_j(\theta_j)] + c, &(2.14)
\end{aligned}$$

where $\mathbb{E}_{-j}[\cdot]$ denotes the expectation over the variational distribution except for factor q_j. Except for the constant c, this is equal to the negative KL divergence between q_j and the expectation of the logarithm of the joint distribution $p(\theta_j, \theta_{-j}, D)$ with respect to q_{-j}. As the KL divergence is nonnegative and equals zero if and only if $q = p$, the optimal variational factor $q_j^\star(\theta_j)$ maximizing (2.14) must therefore be

$$q_j^\star(\theta_j) = \exp\big(\mathbb{E}_{-j}[\log p(\theta_j, \theta_{-j}, D)]\big). \tag{2.15}$$

Each variational factor is iteratively updated by (2.15) until the ELBO convergences to a local optimum resulting in a variational distribution $q(\theta)$.

Variational inference is, generally speaking, approximating a probability distribution by optimizing some distance measure between the target distribution and the approximated distribution. However, while the KL divergence and mean-field family is well-suited for being optimized, the former does not lead to the tightest ELBO(Leisink & Kappen, 2001; Barber & van Laar, 2011) and the latter is limited in its expressibility from the strong independence assumptions(Blei et al., 2016). They have, however, been applied in this work due to their simplicity as part of the work in paper i.

2.1.1.3 Posterior Predictive Distribution

The posterior distribution $p(\theta|D)$ captures uncertainty in the model parameters. This uncertainty can estimate the probability of observing some new data d_{new} through the posterior predictive distribution, which is the marginalized probability distribution of d_{new} written as

$$p(d_{\text{new}}|D) = \int p(d_{\text{new}}|\theta)p(\theta|D)d\theta, \tag{2.16}$$

where the distribution $p(\cdot|\theta)$ remains the same as the likelihood used to compute the posterior distribution. The posterior predictive distribution is used in the work of paper iii to evaluate new data as it arrives in the online setting.

2.1.1.4 Model Selection

Bayesian inference can also be used to perform model selection between competing hypotheses about the data generating process. If we have a discrete number M of models, each denoted $p(D|m_i)$ for $i = \{1, \ldots, M\}$, and a prior distribution $p(m)$ over the models, then Bayes' theorem gives us their posterior probability as

$$p(m_j|D) = \frac{p(D|m_j)p(m_j)}{\sum_{i=1}^{M} P(D|m_i)p(m_i)}. \tag{2.17}$$

From the posterior distribution we identify the most probable hypothesis by computing

$$j^\star = \underset{i\in\{1..M\}}{\arg\max}\ \ p(m_i|D). \tag{2.18}$$

This way of performing model selection is an essential part of the Bayesian Hierarchical Clustering method presented in section 4.1.1.

2.1.1.5 Automatic Relevance Determination

Automatic Relevance Determination (ARD)(MacKay, 1992; Neal, 1996), also known as sparse Bayesian learning(Tipping, 2001), is a choice of a prior able to learn sparse models. Given K parameters $\mathbf{a} = \{a_1, \ldots, a_K\}$ distributed as

$$\mathbf{a} \sim \mathcal{N}(\mathbf{0}, \mathbf{P}^{-1}), \tag{2.19}$$

with $\mathbf{P} = diag(\boldsymbol{\alpha})$ where $\boldsymbol{\alpha} = \{\alpha_1, \ldots, \alpha_K\}$ is the diagonal of the precision matrix for the multivariate normal distribution. These hyperparameters are then estimated by computing $\hat{\boldsymbol{\alpha}} = \arg\max_{\boldsymbol{\alpha}} p(\mathbf{a}|D, \boldsymbol{\alpha})$. Increasingly large values of α_k will progressively concentrate the prior around the zero mean effectively turning off components. Using variational Bayes, p is the variational distribution q for the parameters $\mathbf{a}$. This is an example of the expectation-maximization (EM) algorithm(Bishop, 2006). An ARD prior is specified for the probabilistic PARAFAC2 models in paper i.

2.2 Unsupervised Learning

When the answer to our question is not observable or difficult to obtain, unsupervised learning might be able to help us. Unsupervised learning methods are used to analyze data with little to no additional input from a user. In contrast, supervised learning methods require some annotation of the data observations such as class labels or target values of a dependent variable of interest. These annotations provide the learning algorithm with the expected output based on the input. Unsupervised methods have to define the desired output through some notion of similarity or a loss function. In a probabilistic setting where the data might be assumed to be independently and identically distributed according to $p(x)$, it can be meaningful to estimate such distributions.

2.2.1 Decomposition methods

The first class of unsupervised learning algorithms that will be described is commonly known as decomposition methods. These methods can estimate a factorization of a data set specified by the applied model. Below is a description of some of the most important models for two-way and three-way data.

2.2.1.1 Two-Way Decompositions

The well-known PCA model is useful for two-way data as it models patterns with a bilinear relationship(Kroonenberg, 2007). To do the same thing on three-way data, some considerations have to be taken as it can not be directly applied for this type of data.

Given a data matrix $\mathbf{X}$ with I samples along the first axis and J features along the second axis, PCA decomposes $\mathbf{X}$ into two matrices and an additive error matrix: orthonormal matrix $\mathbf{A}$ of the size $I \times M$ with loadings for each row; orthogonal matrix $\mathbf{B}$ of the size $J \times M$ with loadings for each column; matrix $\mathbf{E}$ of size $I \times J$ with the residuals. If matrix $\mathbf{B}$ is further decomposed into the product of $\mathbf{D}$ — a diagonal matrix of size $M \times M$ containing the standard deviations of the columns of $\mathbf{B}$ — and $\mathbf{F}$ of size $J \times M$ — equal to $\mathbf{B}$ normalized along the columns — the decomposition becomes the singular value decomposition (SVD). The number of components M can equal to the number of features J or lower. These decompositions can be written as

$$\mathbf{X} = \mathbf{A}\mathbf{B}^\intercal + \mathbf{E} = \mathbf{A}\mathbf{D}\mathbf{F}^\intercal + \mathbf{E}. \tag{2.20}$$

A three-component SVD decomposition is visualized in Figure 2.1. On the right-hand side, each component has been marked by a color. The contribution of each component m is the outer product of column m in $\mathbf{A}$ and $\mathbf{F}$ scaled with d_{mm}: $\mathbf{X}_m = d_{mm}\mathbf{a}_m\mathbf{f}_m^\intercal$. The reconstruction of $\mathbf{X}$ is the sum of these M rank-one arrays.

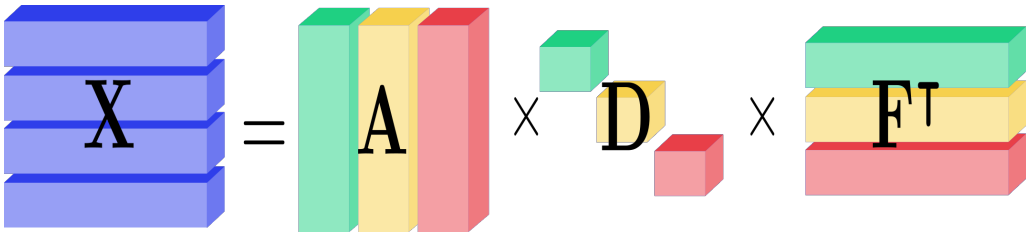


Figure 2.1: A data matrix with 4 rows/features for J occasions along with the individual matrices of a 3 component SVD. On the right-hand side, the elements contributing to each component have been marked by a color.

2.2.1.2 From Two-Way to Three-Way Data Analysis

Generalizations of data arrays arranged as vectors or matrices have been referred to as multiway, multimode or higher-order data/arrays/tensors. In this work, we adopt the wording *multiway arrays* with *mode* referring to some dimension in the data. These multiway arrays allow for a more complex structure than vectors or matrices as they can capture the relationship of measurements simultaneously across repeated experiments or conditions or both for multiple objects. Multiway analysis also comes with the advantage of not requiring the same set of constraints to obtain unique solutions or suffering from rotational indeterminacy(Kroonenberg, 2007; Mørup, 2011; Bro, 2006).

A data set of I observations on J features is often considered as a collection of vectors — written as $D = \{\mathbf{x}_i\}_{i=1,\ldots,I},\ \mathbf{x}_i \in \mathbb{R}^J$. Concatenating these vectors forms a matrix: $\mathbf{X} = \left[\mathbf{x}_1 \ldots \mathbf{x}_I\right]^\intercal$; concatenating these matrices along the third mode forms a three-way array: $\boldsymbol{\mathcal{X}} = \left[\mathbf{X}_1 \ldots \mathbf{X}_K\right]$, where the matrices in the three-way array are referred to as frontal slices. These arrangements of the data are visualized in Figure 2.2. This generalizes to even more ways by further concatenating three-way arrays forming four-way arrays and so on. The number of ways/modes depends on

the data design; here we focus on three-way profile data where measurements are performed simultaneously for all features for all objects at all occasions — a so-called fully crossed design. Further discussion of data designs and terminology can be found in chapter 3 of (Kroonenberg, 2007).

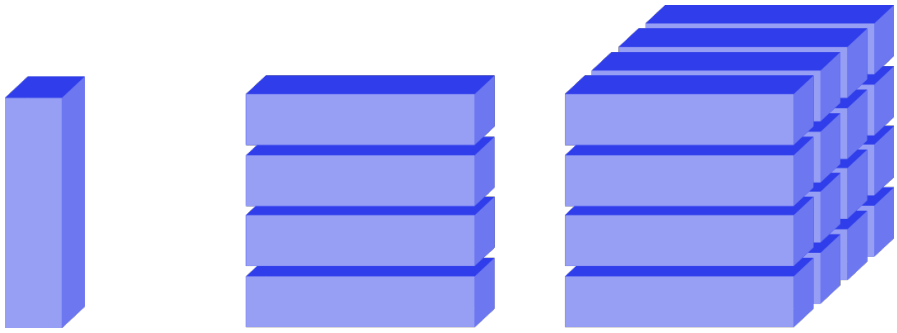


Figure 2.2: Profile data as represented by a vector (one-way), a matrix (two-way), or a data-cube (three-way) — from left to right.

With these structures in mind, we continue on to describe decomposition models for three-way data.

2.2.1.3 Three-Way Decompositions

Attempts to apply the two-way SVD to three-way data have been made by performing an independent SVD on each frontal slice followed by some estimation of similarity between the solutions. Another approach could be to transform the multiway array into a two-way array by flattening it. This is often done by either taking the sum or average over enough ways until only two remain. More direct approaches can, however, be taken when dealing with three-way data (Kroonenberg, 2007), which will be explored in the following. Multiway data poses additional challenges to pre-processing in comparison to two-way data, and the demand for tools able to handle multiway data is growing (Esslinger, Riedl, & Fauhl-Hassek, 2014).

Two of the most important multiway models(Mørup, 2011) are the *TUCKER* and *Canonical Decomposition (CANDECOMP)*(Carroll & Chang, 1970)/*Parallel Factor Analysis (PARAFAC)* (R. a. Harshman, 1970) models. The TUCKER model(Tucker, 1966) is most flexible as it allows any two components across the modes to have linear interactions and, at the same time, for a different number of components for each way in the data. It can fully decompose the data array while its components are, however, not unique similar to two-way PCA(Kroonenberg, 2007). The second model — referred to as PARAFAC in this work as the focus of attention in the following will be on the extension named PARAFAC2. It is a special case of the TUCKER model with less flexibility as it requires the same number of components for each way and only allows for linear interactions between components of the same indices across the ways. The benefit of these limitations results in unique solutions that are more interpretable, but at the risk of encountering degenerate solutions with highly collinear components for data arrays not

fully explained by the more simple interactions between the components.

The component loadings of the PARAFAC model are shared across the first and second way, while they are different across the third way, i.e. for each frontal slice. The PARAFAC model for each frontal slice can be written as

$$\mathbf{X}_k = \mathbf{A}\mathbf{D}_k\mathbf{F}^\intercal + \mathbf{E}. \tag{2.21}$$

Compared to performing a SVD on each frontal slice, the PARAFAC model identifies loadings of the components of the first and second way simultaneously for all frontal slices only varied by the loadings of the third way expressed by the diagonal matrix $\mathbf{D}_k$ for the k'th frontal slice. The resulting components of the decomposition of a multiway array $\mathcal{X}$ of the size $I \times J \times K$ using three components are illustrated in Figure 2.3.

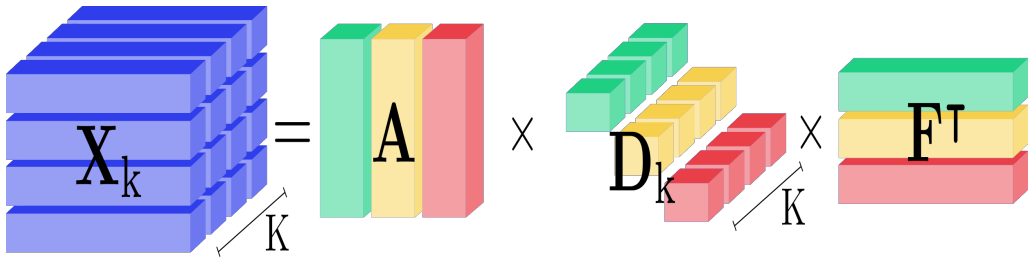


Figure 2.3: A data tensor with 4 rows/features (I) across J occasions repeated 4 times (K) along with the individual matrices of a 3 component PARAFAC decomposition. On the right-hand side, the elements contributing to each component have been marked by a color.

The PARAFAC2 model is an extension of the PARAFAC model which relaxes the assumption that the component loadings are shared across frontal slices in two of the ways to only one. Here we assume that the components of the first way are shared while components of the second way have loadings for each frontal slice similar to the third-way components. Writing this decomposition for each frontal slice gives

$$\mathbf{X}_k = \mathbf{A}\mathbf{D}_k\mathbf{F}_k^\intercal + \mathbf{E}. \tag{2.22}$$

This relaxation alone leads to non-unique solutions, but (R. A. Harshman, 1972) suggested putting an invariance constraint on the loadings of $\mathbf{F}_k$, which can lead to unique solutions if the correct number of components is used. This constraint says that the cross products of these matrices should be constant over k, i.e. $\mathbf{F}_i^\intercal\mathbf{F}_i = \mathbf{F}_j^\intercal\mathbf{F}_j$ for all pairs $i, j = 1 \ldots K$. This constraint together with (2.22) is the PARAFAC2 model. A three-component PARAFAC2 is visualized in Figure 2.4. In the next chapter, we will see how to estimate this model and perform model selection.

2.2.2 Clustering

Clustering analysis is the second class of unsupervised learning algorithms considered in this work. These algorithms cover important unsupervised learning methods able to automat-

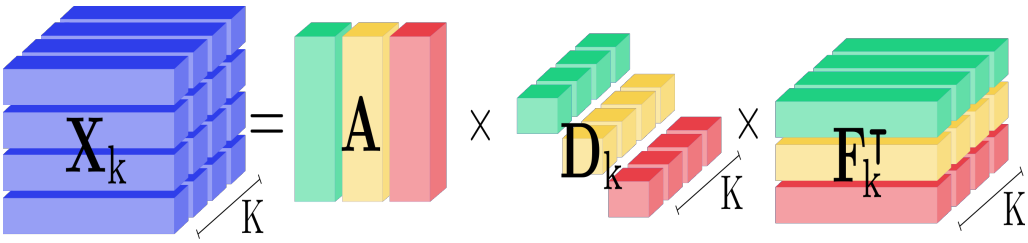


Figure 2.4: A data tensor with 4 rows/features (I) across J occasions repeated 4 times (K) along with the individual matrices of a 3 component PARAFAC2 decomposition. On the right-hand side, the elements contributing to each component have been marked by a color.

ically identify groups of similar data points independent of any labeling information(Jain, Narasimha Murty, & Flynn, 1999; Mittal, Goyal, Hemanth, & Sethi, 2019). Clustering is widely used including for tasks found in bioinformatics(MacCuish & MacCuish, 2010), community detection(Blundell & Teh, 2013; Jorgensen, Morup, Schmidt, & Herlau, 2016), image processing(Coleman & Andrews, 1979; Ahmed, 2015) and text mining(Srivastava & Sahami, 2009). A common notion of a high-quality clustering is when most of the grouped objects are more similar than when compared to objects of different groups. While many different methods for performing clustering exists, we here focus on the important type of approach known as hierarchical clustering.

2.2.2.1 Hierarchical Clustering

Many clustering algorithms are dependent on the user to select a number of clusters to be used(Hastie, Tibshirani, & Friedman, 2009). The most suited number of clusters can be identified by evaluating models fitted using some range of the number of clusters. This can be done qualitatively by an expert or quantified by some measure of fit for how meaningful the clusters are. However, defining what constitutes meaningful clusters is no easy task(Hennig, 2015). Contrarily, hierarchical clustering does not require the user to specify the number of clusters as these methods do not estimate a partition of the data, but a full hierarchy of each data point. Each level of the hierarchy is a particular partitioning of the data. This hierarchy can be further analyzed to determine a hard partitioning of the data by selecting one of these levels. Hierarchical approaches are either agglomerative or divisive. Agglomerative methods build the hierarchy bottom-up by merging a pair of nodes from the previous level. This starts with each node being a single data point and ends with a single node of all the data. Divisive methods start from the top and create new clusters by splitting an existing one into two groups. While these algorithms date back to the 1970s, hierarchical algorithms are still being actively analyzed and develop (Murtagh & Contreras, 2012; Charikar, Chatziafratis, Niazadeh, & Yaroslavtsev, 2019; Kobren, Monath, Krishnamurthy, & McCallum, 2017a).

Hierarchical agglomerative clustering (HAC)(Sneath, Sokal, & Others, 1973) uses the bottom-up approach based on a measure of dissimilarity. The dissimilarity of two clusters C_1 and C_2 is computed using a pairwise comparison between the elements of either cluster using a linkage

function. One such function is known as single linkage which is based on the minimum distance between any pairing of points belonging to separate clusters. Said another way, it compares the nearest-neighbors of separate clusters. This is computed as

$$L_{\text{single}}(C_1, C_2) = \min_{n_i \in C_1, n_j \in C2} d(n_i, n_j).$$

Another example of linkage function is the complete linkage. It computes the distance of the most dissimilar pair to be used for merging pairs. It is computed as

$$L_{\text{complete}}(C_1, C_2) = \max_{n_i \in C_1, n_j \in C2} d(n_i, n_j).$$

A linkage function operating somewhere between the single and complete linkage definitions is the group average linkage. This uses the average dissimilarity between the clusters and is computed as

$$L_{\text{group average}}(C_1, C_2) = \frac{1}{N_{C_1} N_{C_2}} \sum_{n_i \in C_1} \sum_{n_j \in C_2} d(n_i, n_j),$$

where N_{C_1} and N_{C_2} is the number of data points in each cluster. These three linkage functions are the most common ones and they each have different properties. Single linkage will usually combine data points through chains of similar points. This might create clusters with very dissimilar points as they only need to be connected through a chain of pairwise similar data points. Complete linkage will tend to create clusters with very similar data points as all of them are required to be close for a merge to happen. However, it might generate clusters with data points that are closer to data points belonging to other clusters than its own. The group average linkage will produce clusters somewhere in between, but will be affected by the numerical scale of the dissimilarities between the data points as a simple monotone transformation of the dissimilarities can change its result(Hastie et al., 2009).

The hierarchy of data points is commonly visualized as a dendrogram as shown later in Figure 4.1. This detailed graphical representation of the hierarchy that is easy to interpret has been important for the popularity of hierarchical clustering algorithms(Hastie et al., 2009).

2.2.2.2 Online Clustering

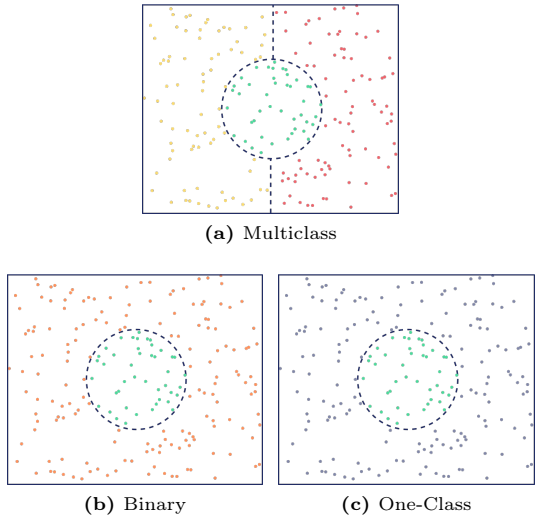
Methods able to learn in an online fashion can receive input from a data stream continuously while updating their model as new data arrives. Data is usually considered arriving one data point at a time, but methods might also perform batch updates simply by waiting until some number of data points have been collected. Online clustering methods can be applied for modern tasks on large data sets(Menon et al., 2019; Guha, Meyerson, Mishra, Motwani, & O'Callaghan, 2003; Guha & Mishra, 2016). Important challenges in this setting include limiting the memory footprint, having efficient updates and adjusting the model complexity to the observed data.

In the online setting — especially for monitoring and sensing systems — novelty and outlier detection are important tasks to be able to perform(Garcia, Poel, Kok, & de Carvalho, 2019; Sadik & Gruenwald, 2011). This could both be for critical systems that are required to behave in a certain way, or systems with challenges such as concept drift where the distribution of the

classes themselves might change over time, or new classes appear as time goes by. Hierarchical clustering methods are good candidates for online methods as the hierarchy provides a tree data structure for efficiently searching for similar objects and performing updates. Later, in relation to paper iii, we will look at specific methods able to perform online clustering.

2.3 One-class Classification

In contrast to binary and multiclass classification — which aim to classify objects into two or more classes respectively — one-class classification(Moya & Hush, 1996) tries to distinguish a single class from all others as illustrated in Figure 2.5. One-class classification can be performed on the target class alone, but can alternatively incorporate information from other classes. (Rodionova, Oliveri, & Pomerantsev, 2016) distinguishes between these settings and denotes the former, which exclusively models the target class, the *rigorous* approach and the latter, incorporating exterior information, the *compliant* approach. The rigorous approach is inherently a harder problem as no counter-examples of the target class are given, so determining a suitable decision boundary can only be done using the sample distribution of the target class. One-class classification is also referred to as class modeling, outlier detection, anomaly detection or novelty detection(Oliveri et al., 2014; Zimek & Schubert, 2017; Pimentel, Clifton, Clifton, & Tarassenko, 2014).



(a) Multiclass

(b) Binary **(c)** One-Class

Figure 2.5: Different types of classification problems. The colors represent different classes, while the gray points outside the circle in (c) have not been observed.

One-class classification has applications for monitoring and critical systems found for example in security, control and healthcare systems(Pimentel et al., 2014). It has also been recommended for food authentication(Oliveri & Downey, 2012), because it can be difficult to obtain counter-examples and impossible to exhaustively collect all types of counter-examples. Consequently,

the one-class classification models have some properties that are well suited for the task of authentication. Class modeling aims to identify a subspace of the input space belonging to the target class while everything outside this subspace will be considered to be of the non-target classes. In a setting with multiple classes of authentic classes, one can fit a one-class classification model to each of them, and thus describe a number of classes that may or may not overlap potentially leading to multiple assignments. The system will output either that the test sample belongs to one or more authentic classes, or it does not. This is not the case for conventional binary or multiclass classification where two or more classes are described; here the assignments are most often mutually exclusive and any data point will be recognized as one of the known classes.

Chapter 3

Probabilistic Multiway Modeling for Chemometrics

The PARAFAC2 model described in the previous chapter is important for the field of chemometrics. In this chapter, we describe the work of the first two papers. Here, a probabilistic formulation of the PARAFAC2 model together with experiments on chromatographic data and one-class classification tasks are presented. Several methods for estimating the PARAFAC2 decomposition model using alternating least squares solutions exist, and with some recent advances introducing more efficient computations and nonnegativity. Some of the benefits of using a probabilistic PARAFAC2 include new ways to perform model selection and more flexible noise models, as will be described in greater detail shortly.

The PARAFAC2 model is an extension of the PARAFAC model which was based on an idea known as parallel proportional profiles first explained by (Cattell, 1944), which states that if a factor is an underlying true component with organic structure then its loadings will be shared under different experimental conditions only varying in their magnitude. Such components differ from the more mathematical abstract factors identified by optimizing for some target objective like maximizing their variance. While the three-way PARAFAC model assumes such proportional components in two of its modes, this has only to be true for one of the modes in the three-way PARAFAC2 model. In this work, this is assumed to be the first mode, while the second model only assumes a constant correlation between the components. This allows for greater flexibility as the model can describe shifts in the components as long as these shifts are similar in size(Bro, Andersson, & Kiers, 1999). These properties make the PARAFAC2 well suited for chromatographic data as the data might be shifted between runs on different samples(Group, Science, & Veterinary, 1999; Amigo, Skov, Bro, Coello, & Maspoch, 2008).

Formulating and computing multiway models using probabilistic methods is nothing new. The TUCKER model has been formulated as a probabilistic model for continuous(Chu & Ghahra-

mani, 2009; Mørup & Hansen, 2009), count(Schein, Zhou, Blei, & Wallach, 2016; Han & Dunson, 2018), and categorical data(Yang & Dunson, 2016). A probabilistic version of the PARAFAC model was first introduced in the works (Nielsen, 2004). A recent effort in gathering these models — and more — in the probabilistic tensor decomposition toolbox is described in (Hinrich, Madsen, & Mørup, 2020).

3.1 Chemometrics

Chemometrics is a data-first approach trying to acquire new knowledge by analyzing data inexpensively generated by chemical instrumentation(Lavine & Workman, 2002). The field of chemometrics has been one of the main drivers for the development of multiway analysis(Kroonenberg, 2007). This comes naturally as applications in chemistry provide many data sources able to generate multiway data. As described by (Bro, 2006), these sources include fluorescence spectroscopy, chromatography, flow injection, magnetic resonance (MR), and electroencephalography (EEG) with applications for calibration, multivariate statistical process control, metabonomics, environmental analysis, kinetics, sensory analysis and classification.

3.1.1 Gas Chromatography-Mass Spectrometry

Some details of gas chromatography-mass spectrometry (GC-MS) data are presented here to exemplify how real data look like and relate to the model structures described in the previous chapter. GC-MS measurements are an important separation technique to identify chemical components(Eric Stauffer, Julia A. Dolan, Reta Newman, 2008) able to provide multiway data which can be processed using chemometrics. A mass spectrometer measures ion counts within a specified range of mass-to-charge ratio on substances separated using a gas chromatograph on some samples of interest. An example of such data is presented as a single interval of a GC-MS data set in Figure 3.1. The left-most figure shows the individual curves of the data from the individual mass-to-charge ratio sensors, but we can think of the GC-MS measurements as a data matrix as visualized in the surface plot or top-down view in the middle and right-most figures respectively.

A multiway array is obtained by performing GC-MS measures on multiple samples. This is the case shown left-most block in Figure 3.2. Now, using a PARAFAC2 model with 3 components, the multiway array would be decomposed into the matrices $\boldsymbol{A}$, $\boldsymbol{C}$, $\boldsymbol{F}$ and $\boldsymbol{P}_k$ for each sample k as shown in Figure 3.2 similar to Figure 2.4. Here, the k'th row of $\boldsymbol{C}$ is equal to the diagonal of $\boldsymbol{D}_k$.

To process GC-MS data one must deal with artifacts from the chromatographic measurements such as baseline drifts, peak shift, low signal-to-noise ratios and overlapping or co-elution of the components(Amigo et al., 2010; Yi et al., 2016). These are often dealt with individually by specialized software resulting in a complicated and slow manual process(Vestner et al., 2016; Yi et al., 2016). The PARAFAC2 model can handle these challenges as it models the noise and can deconvolute and integrate the peaks of the components when retention time shifts exist in the data(Group et al., 1999; Amigo et al., 2008).

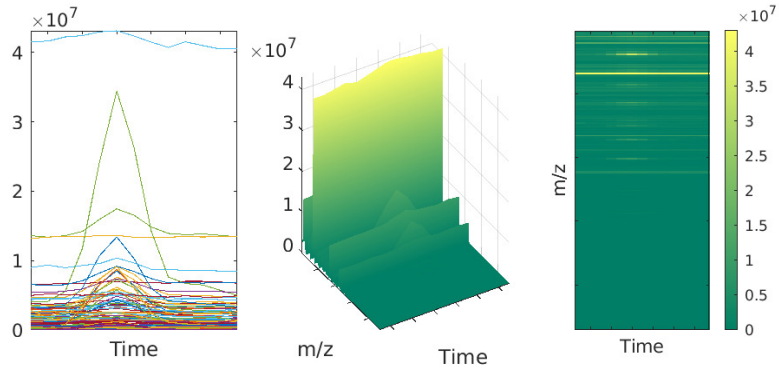


Figure 3.1: A single frontal slice / sample of the tobacco data visualized with different perspectives — from left to right: curves / side-view; surface / 3-d; heatmap / top-down.

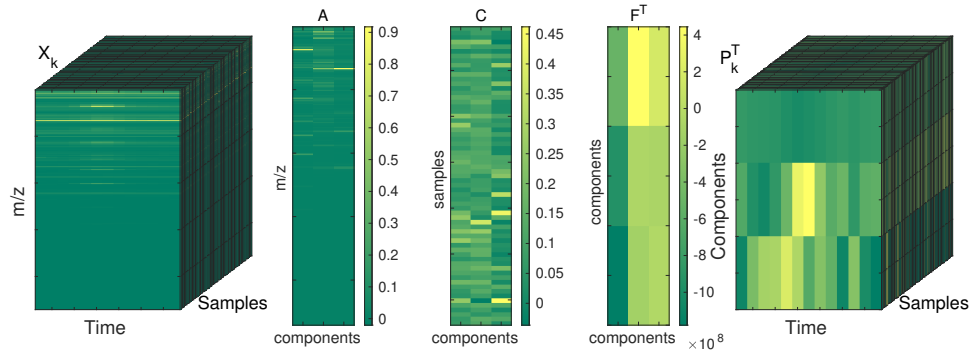


Figure 3.2: Multiple samples measured using GC-MS represented as a multiway array and its corresponding 3-component PARAFAC2 matrices.

3.2 PARAFAC2

Before going into the details of the probabilistic PARAFAC2 models, we take a look at existing methods for estimating the model and performing model selection.

3.2.1 Alternating Least Squares Solution

Estimating the PARAFAC2 model using an alternating least squares algorithm(Kiers, Ten Berge, & Bro, 1999) can be done by solving the following optimization problem

$$\min \sum_k \|\boldsymbol{X}_k - \boldsymbol{A}\boldsymbol{D}_k\boldsymbol{F}^\intercal\boldsymbol{P}_k^\top\|^2 \quad \text{s.t. } \boldsymbol{P}_k^\top\boldsymbol{P}_k = \mathbf{I}. \tag{3.1}$$

This formulation of the model structure uses the observation that to satisfy the invariance constraint on $\boldsymbol{F}_k$, it is necessary and sufficient to have $\boldsymbol{F}_k = \boldsymbol{P}_k\boldsymbol{F}$ for an orthonormal matrix $\boldsymbol{P}_k$. It can be shown that the optimal $\boldsymbol{P}_k$ for (3.1) is equal to the $\boldsymbol{P}_k$ which maximizes the trace

$$\mathrm{Tr}(\boldsymbol{F}\boldsymbol{D}_k\boldsymbol{A}^\top\boldsymbol{X}_k\boldsymbol{P}_k) \tag{3.2}$$

This is due to the orthogonality constraint as only the linear term depends upon $\boldsymbol{P}_k$ after expanding the squared residuals. This is solved by computing the SVD decomposition

$$\boldsymbol{U}_k\boldsymbol{S}_k\boldsymbol{V}_k^\top = \boldsymbol{F}\boldsymbol{D}_k\boldsymbol{A}^\top\boldsymbol{X}_{\boldsymbol{k}}.$$

where the product of $\boldsymbol{V}_k$ and $\boldsymbol{U}_k$ gives the optimal $\boldsymbol{P}_k$(Green, 1952; Kiers et al., 1999)

$$\boldsymbol{P}_k = \boldsymbol{V}_k\boldsymbol{U}_k^\top \tag{3.3}$$

After having computed the optimal $\boldsymbol{P}_k$ for a given $\boldsymbol{A}$, $\boldsymbol{D}_k$ and $\boldsymbol{F}$, the least-squares problem can be reduced to that of the PARAFAC model by projecting the k frontal slices $\boldsymbol{X}_k$ of the data tensor onto $\boldsymbol{P}_k$, like so

$$\sum_k \|\boldsymbol{X}_k\boldsymbol{P}_k - \boldsymbol{A}\boldsymbol{D}_k\boldsymbol{F}^\intercal\|^2. \tag{3.4}$$

which means we can find the optimal $\boldsymbol{A}$, $\boldsymbol{D}_k$ and $\boldsymbol{F}$ by computing a PARAFAC model for $\boldsymbol{X}_k\boldsymbol{P}_k$. Estimating these parameters for the PARAFAC model is explained in (Bro, 1997).

This algorithm for estimating the PARAFAC2 model was the first approach directly fitting the data. Its predecessor proposed by (R. A. Harshman, 1972) estimated a derived model based on the cross-product of the PARAFAC2 structure. Some extensions have more recently been introduced to improve upon the direct fitting algorithm. These include a geometric search algorithm making the alternating least squares algorithm more efficient(Tian, Wu, Min, & Bro, 2018), a more efficient computation of the projection in (3.4)(Perros et al., 2017), and the incorporation of a semi-algebraic approach to more efficiently compute the underlying PARAFAC model(Cheng & Haardt, 2019). Also, a nonnegative version has been developed(Cohen & Bro, 2018).

3.2.2 Model Selection

The PARAFAC2 relies on the model order specification made by the user. Existing approaches rely on the evaluation of several diagnostics either by the user or automatically by a system trained in a supervised manner to do so. We take a look at two important ones.

The explained variance quantifies how much of the variation present in the data is explained by the model(Kiers et al., 1999). This is computed as

$$\mathrm{R2} = 1 - \frac{\sum_k \|\boldsymbol{X}_k - \boldsymbol{A}\boldsymbol{D}_k\boldsymbol{F}^\top\boldsymbol{P}_k^\top\|^2}{\sum_k \|\boldsymbol{X}_k\|^2}. \tag{3.5}$$

The explained variance will increase as the model overfits, and hence it is unable to detect when no more parameters should be introduced. However, it can indicate whether all of the signal is explained by the model if the user has an idea of the expected noise level.

The core consistency diagnostic (CCD) is originally proposed for the PARAFAC model(Bro & Kiers, 2003). However, it is found to be useful for the PARAFAC2 model as well(Kamstrup-Nielsen, Johnsen, & Bro, 2013a), but not recommended being a standalone measure of model complexity. It uses the fact that the PARAFAC model is a constrained Tucker model with a superdiagonal unit core array $\mathcal{I}$. The CCD quantifies how much a core array $\mathcal{G}$ in an unconstrained TUCKER model, with component loadings equal to the PARAFAC model, deviates from a superdiagonal unit core array. This is written as,

$$\text{CCD} = 100 \left(1 - \frac{||\mathcal{G} - \mathcal{I}||_{\mathcal{F}}^2}{||\mathcal{I}||_{\mathcal{F}}^2}\right)$$

where $||\cdot||_{\mathcal{F}}$ denotes the Frobenius norm. The connection to the PARAFAC2 model is based on the projection in (3.4), as this allows for the same approach in estimating a core array in the TUCKER model.

A method for automatically determining the model complexity based on a classification model fitted to 102 diagnostics — including the R2 and CCD — used as features was proposed by (Johnsen, Amigo, Skov, & Bro, 2014). A software package called PARADISe for performing model selection manually based on diagnostics such as the R2 and CCD, but also the loadings, is made publicly available(Johnsen, Skou, Khakimov, & Bro, 2017).

3.2.3 Probabilistic PARAFAC2 (paper i)

The work on the Probabilistic PARAFAC2 described below is presented in [i/(Jørgensen et al., 2018)], which is attached in the appendix. It has also been summarized in a shorter workshop paper(Jørgensen et al., 2019) for the *Machine Learning and Physical Sciences* workshop at the 33rd conference on Neural Information Processing Systems (NeurIPS) 2019, which is also attached in the appendix. The specification and most of the implementation of the probabilistic PARAFAC2 model were carried out as part of my masters' thesis, which lay much of the foundation for the paper included here. The body of the paper includes text originally written for my masters' thesis. Specifically, these are parts related to the specification of the model and related methods used to develop the probabilistic PARAFAC2, but most of the original text has been heavily revised several times with many improvements based on feedback given by co-authors and reviewers from previous submissions. All the data and results presented were obtained as part of my PhD studies.

A probabilistic framework for the PARAFAC2 model using variational inference was developed. The framework opens up for additional modeling of the noise, improves robustness to misspecification of the model order through ARD priors and quantifies the model fit through a lower bound on the marginal likelihood of the data.

The full specification of the generative model for the PARAFAC2 decomposition with M components as formulated in (3.1) with all the diagonals of the diagonal matrices $\boldsymbol{D}_k$ having been

collected as rows in the matrix $\boldsymbol{C}$ is

$$
\begin{aligned}
\boldsymbol{a}_{i\cdot} &\sim \mathcal{N}(\mathbf{0}, \mathbf{I}_M), \\
\boldsymbol{f}_{m\cdot} &\sim \mathcal{N}(\mathbf{0}, \mathbf{I}_M), \\
\boldsymbol{c}_{k\cdot} &\sim \mathcal{N}(\mathbf{0}, \mathrm{diag}(\boldsymbol{\alpha}^{-1})), \\
\text{i)} \quad \boldsymbol{P}_k &\sim \mathrm{vMF}\left(\mathbf{0}\right), \\
\text{ii)} \quad \boldsymbol{P}_k &\sim \mathcal{MN}\left(\mathbf{0}, \mathbf{I}_J, \mathbf{I}_M\right), \\
\tau_k &\sim \mathrm{Gamma}(a_{\tau_k}, b_{\tau_k}), \\
\boldsymbol{X}_k &\sim \mathcal{N}(\boldsymbol{A}\boldsymbol{D}_k\boldsymbol{F}^\top\boldsymbol{P}_k^\top, \tau_k^{-1}\mathbf{I}_J).
\end{aligned}
$$

Here each row in $\boldsymbol{A}$ and $\boldsymbol{F}$ is normal distributed with zero mean and an identity matrix as covariance. The rows in $\boldsymbol{C}$ are also normal distributed with a zero mean but uses a length scale parameter vector $\boldsymbol{\alpha}$ along the diagonal of an isotropic covariance matrix. These parameters form the ARD prior which enables the inference of sparsity in the model as described in the theory chapter. Two different distributions for the prior of the orthonormal matrices $\boldsymbol{P}_k$ were proposed. The first being the von Mises-Fisher matrix distribution (vMF). This distribution has support on the Stiefel manifold consisting of orthonormal matrices. The second is based on a matrix normal distribution, where the mean parameter of its corresponding variational factor is constrained to be orthonormal ($c\mathcal{MN}$). A closed-form solution to estimating this parameter by maximizing the ELBO is found to be similar to the approach for estimating the optimal $\boldsymbol{P}_k$ in the direct fitting algorithm. The precision of the noise τ_k follows a Gamma distribution with parameters $a_{\tau k}$ and $b_{\tau k}$. Finally, the likelihood of the data follows a normal distribution with the mean given by the PARAFAC2 decomposition and a diagonal precision matrix with values according to τ_k. These k precision parameters can be inferred individually to model heteroscedastic noise across the frontal slices or assumed to be equal to model homoscedastic noise.

These distributions are the likelihood and priors in the probabilistic model leading to the variational updates derived by (2.15) using the mean-field variational family. More details on the exact expressions of the updates can be found in the paper. We investigated the probabilistic PARAFAC2 in several ways using this variational framework as described in the following.

The data used in the experiments include both synthetic and real data. The synthetic data was generated based on the PARAFAC2 structure to verify that the proposed methods would be able to perform as well as expected on a suitable and known ground truth. The approach taken was inspired by the one presented in the work of (Kiers et al., 1999). Each of the matrices $\boldsymbol{A}, \boldsymbol{F}, \boldsymbol{C}$ and $\boldsymbol{P}_k$ were generated followed by computing the frontal slices $\boldsymbol{X}_k$ for a chosen number K. $\boldsymbol{A}$ was generated from a multivariate normal distribution; $\boldsymbol{F}$ based on a Cholesky factorization of a matrix with 1 in its diagonal and values of 0.4 in its off-diagonal elements; $\boldsymbol{C}$ was generated from a uniform distribution in the range $[0, 30]$; and finally, the orthonormal $\boldsymbol{P}_k$ matrices were estimated using the default orthonormalization function found within MATLAB applied to a random set of vectors from a multivariate normal distribution. Specifically, we generated data sets with dimensions of $50 \times 50 \times 10$ using 4 underlying components. Furthermore, both homoscedastic and heteroscedastic noise was generated for each data set at signal-to-noise ratios within the range $[-20, 10]$ discretized with a step size of 2. Each configuration was repeated 10 times resulting in 320 data sets for the synthetic experiments. The real data consisted of 3 chromatographic data sets of varying complexity: fluorescence measurements of amino acids

with the 3 constituents tyrosine, tryptophan, and phenylalanine(Bro, 1998; Kiers, 1998). It is a small data set with only 5 samples measured with 201 emission and 61 excitation wavelengths. It has previously been studied to determine the usefulness of the CCD for the PARAFAC2 model in(Kamstrup-Nielsen, Johnsen, & Bro, 2013b). The remaining 2 real data sets consist of GC-MS measures on wine and tobacco samples respectively. Both have previously been analyzed by experts on such data providing a good estimate of the most suitable number of components. The wine data(Skov, Ballabio, & Bro, 2008) consists of 44 samples measured on a mass range m/z $5-204$ for elution times $4.5903-4.7525$ min. The tobacco data(Tian et al., 2018) consists of 65 samples measured on a mass range of m/z $50-350.0$ for elution times $4.95-5.03$ min.

The main outcomes of this work are described and discussed in the following paragraphs including the most important figures from the paper.

Investigation of the evidence lower bound (ELBO) for model selection We compared the ELBO to the explained variance, R2, and the CCD on both the synthetic and real data. These results are visualized in Figure 3.3. For the synthetic data, we saw for all the probabilistic models that the ELBO reached a plateau after using the correct or higher number of components. The ELBO as a function of the model order followed the pattern of the R2 curve closely, but instead of plateauing the R2 improved slightly as excessive components were added. The CCD decreased slightly for the model using one additional component than the ground truth but saw a drastic decrease for any higher number of components strongly indicating the wrong model order.

The results on the real data were not as clear. The ELBO increased faster as the number of components increased until the number estimated by the experts in previous work was employed than after adding excess components. The amino acid and tobacco data had a stronger indication of this, but it might also have been more evident for the wine data if more components were considered as it was also recommended requiring more components than the other two data sets.

Comparison of the noise models Throughout the experiments the two noise models of either homoscedastic or heteroscedastic noise was applied. These are denoted by a Δ and Ω respectively in the figures. As was expected for the simulated studies, we saw that the heteroscedastic noise model performed better than the homoscedastic noise model on data with heteroscedastic noise, and very little difference between them on data with homoscedastic noise. These results were expected since the heteroscedastic noise model can describe homoscedastic noise by having equal variance across its parameters, but the homoscedastic noise model can not describe heteroscedastic noise. This is evident from the difference between (a) and (b) in Figure 3.3.

One of the simulated studies was on the model fit performance as a function of the signal-to-noise ratio (SNR). These results are shown in Figure 3.4. The fit was computed on the data without added noise to investigate how well the underlying structure was recovered. Here we saw the effect of the noise models compared to the direct fitting method. The small differences might be attributed to the different methods for orthogonality. The vMF model is more constrained than the $c\mathcal{MN}$model which can explain its superior recovery of the signal. The higher flexibility of

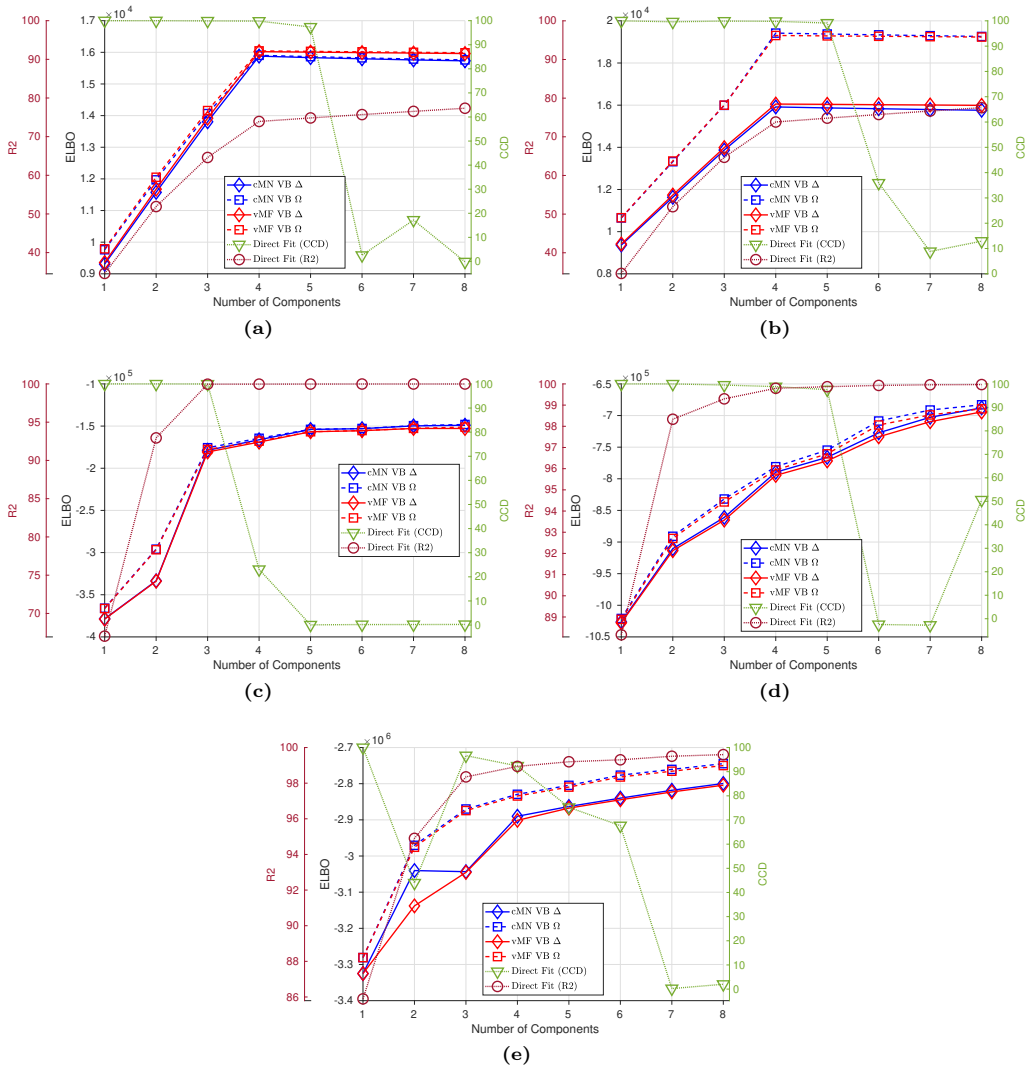


Figure 3.3: The R2, CCD and the ELBO as a function of model order from 1 to 8 components on synthetic data with homoscedastic noise (a) and heteroscedastic noise (b) and the real amino acid (c), wine (d), and tobacco (d) data sets for the different PARAFAC2 models. In the legend, Δ implies a homoscedastic noise model and Ω a heteroscedastic noise model. The ELBO strongly indicates the correct model order for the synthetic data. Figure from paper i.

the $c\mathcal{MN}$ model compared to both the vMF and direct fitting models — as the third-mode components are only orthogonal in their expectation — causes this model to perform worst using

the correct number of components. The difference in performance was more prominent when the models were overspecified as the probabilistic models can automatically turn off the excessive components through their ARD prior. On the real data, we did not see much of a difference for the amino acid and wine data sets. These data sets did not have indications of heteroscedastic noise, so this was expected. The tobacco data were included as part of the analysis since it had some indication of containing heteroscedastic noise which the results agreed with as the heteroscedastic models performed better as seen in (e) in Figure 3.3.

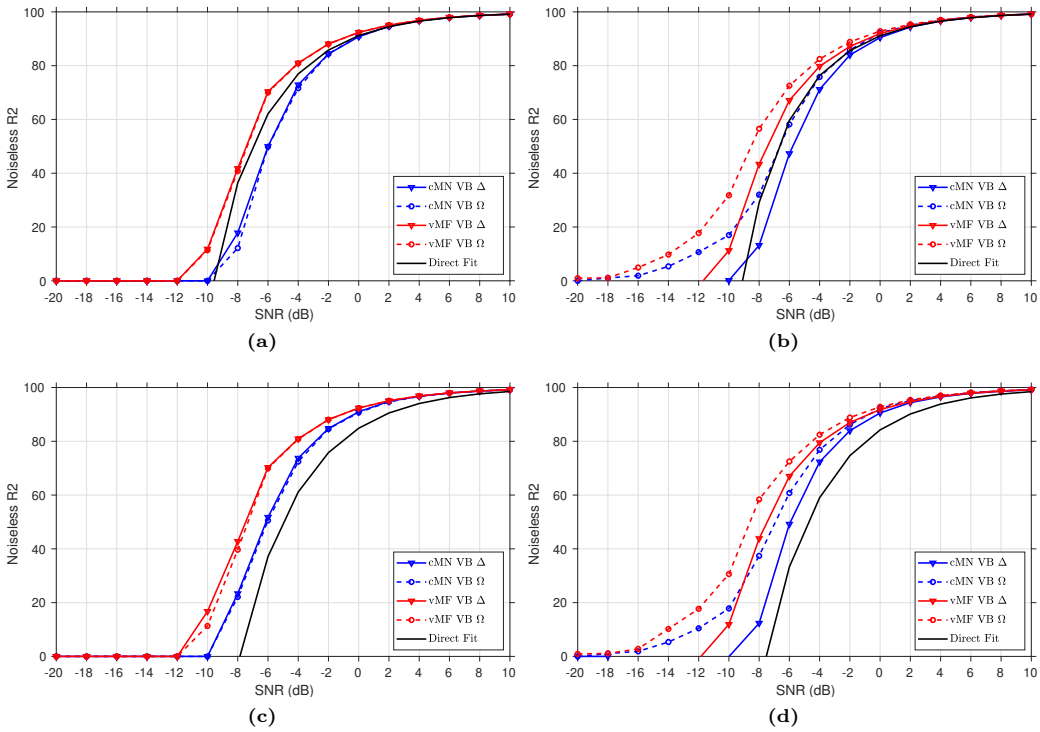


Figure 3.4: Recovery of the underlying signal in synthetic data with varying levels of homoscedastic (a,c) and heteroscedastic (b,d) added noise as measured by noiseless R2. Both for the conventional PARAFAC2 and probabilistic PARAFAC2 models fitted with the true number of components in (a,b), with M = 4, and with an overspecified number of components in (c,d), with M = 6. In the legend Δ indicates a homoscedastic noise model and Ω indicates a heteroscedastic noise model. Figure from paper i.

Investigation of the different methods to handle the orthogonality constraint The main difference between the two models handling the orthogonality constraint is that the vMF model only supports orthonormal matrices while the $c\mathcal{MN}$model results in less constrained

matrices as the orthogonality constraint is on its mean parameter only. This allows the realizations from the $c\mathcal{MN}$distribution to not be orthonormal. The difference between the two models across the experiments was small, and most emphasized from the SNR experiments in Figure 3.4 with the more constrained vMF being more robust to noise.

Evaluation of the ability of the automatic relevance determination priors to simplify the model order The ARD prior of the probabilistic models increases their robustness to model misspecification throughout the experiments. In the simulated studies, it seems to be turning off any unnecessary component as the ELBO reaches a plateau after including the correct number of components or more, and the recovering of the true signal in the SNR experiments only slightly changes from using the exact number of components to an excessive amount. The results on the real data suggest improved robustness to model order misspecification as well. The inspection of the elution profiles and the correlation of the components show that the true components might still be split across multiple components when the model is overspecified as illustrated by Figure 3.5 on the tobacco data set. More similar results on the other data sets can be found in the paper.

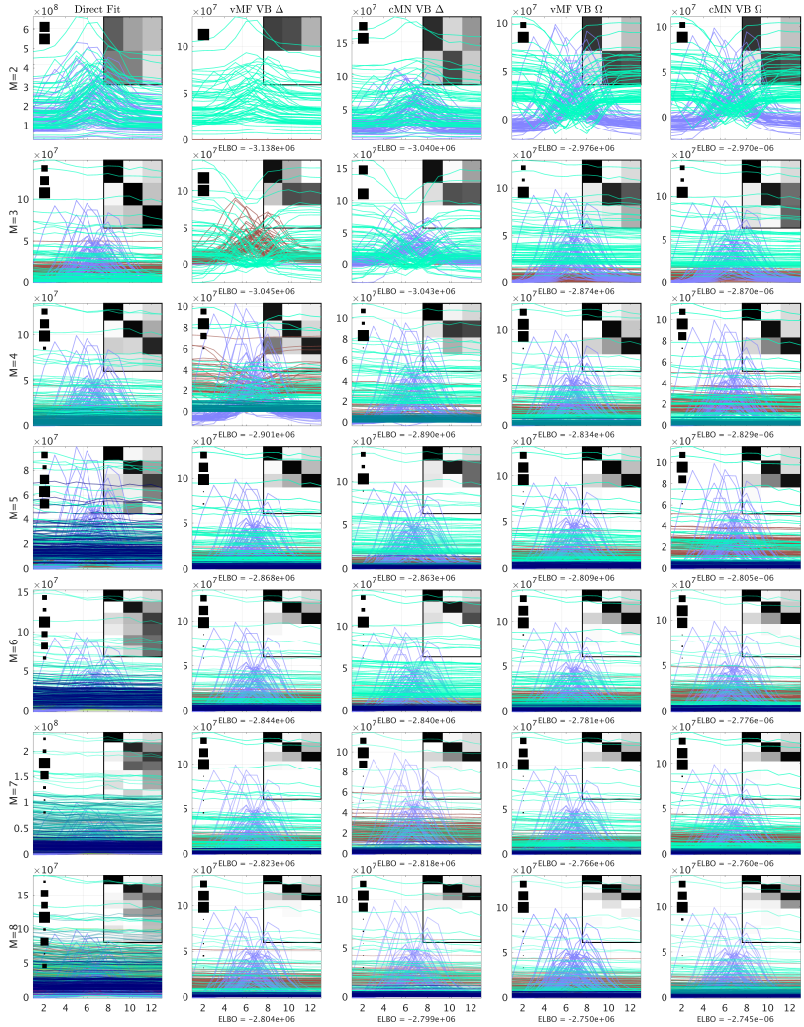


Figure 3.5: The elution profiles of the GC-MS-TOBAC data given by the conventional PARAFAC2 and probabilistic PARAFAC2 models. From top to bottom the profiles consist of 2 to 8 components. For each model, the background heatmap visualizes the correlation between the data reconstruction for each identified component and the componentwise data reconstruction of the conventional PARAFAC2 model with 3 components (expert conclusion). Furthermore, to the left, a Hinton diagram indicates the relative squared Frobenius norm of the componentwise data reconstructions to their sum. In the column headers, Δ indicates a homoscedastic noise model and Ω indicates a heteroscedastic noise model. Figure from paper i.

3.3 Probabilistic PARAFAC2 for Downstream Tasks

A decomposition method like PARAFAC2 can be used as either a preprocessing step or by itself for downstream tasks such as binary or one-class classification to perform anomaly detection(Fanaee-T & Gama, 2016). The preprocessing consists of extracting features for further modeling with multivariate methods like PCA or partial least squares discriminant analysis (PLS-DA), while the one-class classification can be computed based on some quantity estimating the similarity between a target class and test samples. Both of these approaches are exemplified by analyzing food data in the following.

3.3.1 Food Authentication

Food authentication is the task of verifying that a food product complies with its description, which is a concern for the whole supply chain from producers to consumers, but also regulators tasked with ensuring food safety(Danezis, Tsagkaris, Camin, Brusic, & Georgiou, 2016).

Chromatography is an important analytical technique able to separate chemical compounds in food matrices(Cserháti, Forgács, Deyl, & Miksik, 2005). Chromatographic methods can extract chemical fingerprints able to authenticate food based on identifying small analytical differences in patterns or unique compounds. High-resolution techniques include gas and liquid chromatography coupled with mass spectrometry (GS/LC-MS)(Danezis et al., 2016). PARAFAC2 applied to chromatographic data is a strong combination for performing a non-targeted comprehensive analysis of a food product(Wilde, 2019). A non-targeted analysis attempts to capture all the available information in a data set, whereas a targeted approach evaluates specific marker compounds using a control limit(Esslinger et al., 2014).

The PARAFAC2 model based on the direct fitting algorithm has already been applied as a preprocessing tool for tasks related to food authentication, but the probabilistic PARAFAC2 models potentially provide more robustness as they model the uncertainty of parameters. The probabilistic models also provide model selection by adapting the model order automatically, and the option for using their approximated probability distributions for making decisions. All of this will be considered in the following.

3.3.2 Extracting Features Using PARAFAC2

The loadings of the sample mode are the features commonly extracted as illustrated in Figure 3.6. These are also referred to as scores as they express the activation or concentration — depending on the context — of the components. These are computed with a PARAFAC2 model fitted to all samples, while any available labeling information is first used in the downstream multivariate model. These scores are then used as input for some two-way model to either visualize or classify the samples.

Several works have performed feature extraction using the PARAFAC2 model estimated with the direct fitting algorithm including (Amigo et al., 2010, 2008; Toraman et al., 2018; Ebrahimi

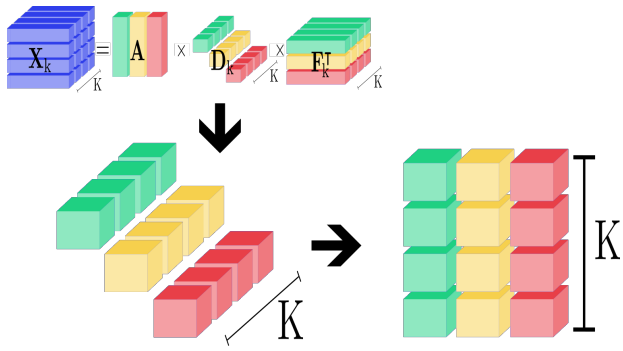


Figure 3.6: Extraction of the concentration levels using the PARAFAC2 decomposition resulting in a matrix of K rows represented by M components (columns).

& Hibbert, 2008; Sales Martínez, Portolés Nicolau, Johnsen, Danielsen, & Beltrán Arandes, 2019; Amante et al., 2019). Most of these either use these features for an explorative analysis using PCA or for performing classification with PLS-DA. The same approach has been used with the PARAFAC model(Lenhardt, Zekovic, Tatjana, & Dramicanin, 2018).

3.3.3 One-class Classification Using PARAFAC2

Instead of fitting an additional multivariate model to features computed for both training and test samples with a PARAFAC2 model, it is possible to take an approach where test samples are evaluated directly by a PARAFAC2 model estimated only on the authentic samples. Three different approaches to this have previously been investigated to perform fault detection. The first uses the control limits on the residuals of the test samples normalized by the number of time instances based on a chi-square distribution (Wise, Gallagher, & Martin, 2001). The same authors also propose to place control limits on the activation of the components — the same scores of the PARAFAC2 model as commonly used for feature extraction — assuming they are normally distributed. The third approach is similar to the second but proposes to estimate sparse components and to only using the most relevant components of the first mode to evaluate test samples (Luo, Chen, Bao, & Tong, 2019).

3.3.4 Probabilistic PARAFAC2 for Food Authentication (paper ii)

In [ii/(Jørgensen & Mørup, 2020)] we investigate the use of the probabilistic PARAFAC2 models for authentication with a focus on food GC-MS data. The authentication can be carried out similar to previous approaches using the concentration levels — the mean of the variational distributions — as features for binary or one-class classification, or by using the probabilistic framework directly for one-class classification.

For this comparison, we consider two GC-MS data sets. The first consisting of 44 wine samples

originating from the 4 different regions Argentina, Chile, Australia and South Africa measured with mass-to-charge ratios of 5 to 204 at 2700 elution time points (Skov et al., 2008). The resulting dimensions are thus $200 \times 2700 \times 44$. The second data set consists of 79 rice samples of 4 different grain cultivars (C. Hu et al., 2016; Sirén, Fischer, & Vestner, 2019). These were measured with a mass-to-charge ratio between 50 to 600 scanned twice per second for a total of 20809 scans. After preprocessing the data using the code made available by (Sirén et al., 2019), the data ended with dimensions $530 \times 20809 \times 79$.

These two data sets are preprocessed either by flattening the data or using the PARAFAC2 model estimated using the direct fitting algorithm or the probabilistic framework. The flattening is both done for the full data set and the same intervals as the PARAFAC2 models are fitted to. These intervals are identified by using an automated segmentation algorithm made available by (Sirén et al., 2019) to split the data. These features are then fitted using either a logistic regression (LR) or support-vector machine (SVM) to perform binary classification, or a one-class SVM to obtain a one-class classification model. The one-class SVM is compared to test samples evaluated based on the ELBO and KL divergence between the distributions of the concentrations of the authentic and test samples as explained in the following.

A tool widely used for fitting the PARAFAC2 models is the PARADISe software (Johnsen et al., 2017). This tool makes it more accessible to perform the model selection by evaluating several diagnostics as mentioned in section 3.2.2. However, the evaluation process is still very manual. As the probabilistic PARAFAC2 models indicated improved robustness to model misspecification, and they provide a quantification of the model fit through the ELBO, we wanted to investigate an approach for taking advantage of these properties as an alternative to the manual process. In a manual process, the user has to determine which model order is best and which of the components correspond to chemical meaningful components. More recent work tried to make this easier by identifying suggestions for which peaks correspond to chemical components and which components are either baselines or otherwise not meaningful using deep learning(Risum & Bro, 2019).

We propose an iterative algorithm using the ELBO of the probabilistic models. The first iteration starts by fitting a 1-component PARAFAC2 model to the data whose ELBO is compared to a PARAFAC2 model with 1 additional component. If the ELBO of this 2-component PARAFAC2 model is lower, the 1-component PARAFAC2 is accepted as the best fit. Otherwise, a new iteration compares the ELBO of the 2-component PARAFAC2 to a 3-component PARAFAC2 following the same procedure. It terminates either when a smaller ELBO is encountered or after reaching the maximum allowed number of components specified by the user.

After having identified a suitable PARAFAC2 model described by its variational distribution $q_{\text{train}}(\boldsymbol{\theta})$ for the authentic samples, we propose to evaluate a test sample $\mathbf{X}_{\text{test}}$ by estimating the variational factors $q_{\text{test}}(\mathbf{C})$ and $q_{\text{test}}(\mathcal{P})$ for the test sample using the variational factors $q_{\text{train}}(\mathbf{A})$ and $q_{\text{train}}(\mathbf{F})$. This is analogous to (Wise et al., 2001) using the direct fitting algorithm for fault detection. The full resulting variational distribution of the test samples is

$$q_{\text{test}}(\boldsymbol{\theta}) = q_{\text{train}}(\mathbf{A}, \mathbf{F}) q_{\text{test}}(\mathbf{c}, \mathcal{P}) q_{\text{noise model}}(\boldsymbol{\tau}). \tag{3.6}$$

The variational factor for the precision $\boldsymbol{\tau}$ depends on the noise model. In case of a heteroscedastic noise model, it is also fitted to the test sample; otherwise, with a homoscedastic noise model it

is equal to $q_{\text{train}}(\boldsymbol{\tau})$.

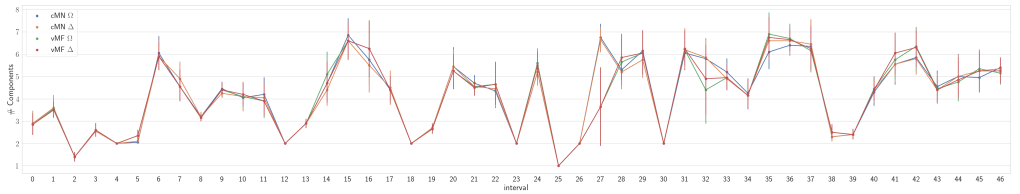
Equipped with this variational distribution of a test sample, we propose two approaches for obtaining a similarity score quantifying how likely it is that the sample is authentic. The first is simply using the ELBO of (3.6). The ELBO is a bound and thus not justified in theory to use for model selection, it can work as intended empirically (Blei et al., 2016). The second approach computes average KL divergence between the variational factors $q_{\text{test}}(\mathbf{c})$ and $q_{\text{train}}(\boldsymbol{C}) = \prod_k q^{(k)}_{\text{train}}(\mathbf{c})$

$$\text{KL}_{\text{avg}}(q_{\text{train}}(\mathbf{C}), q_{\text{test}}(\mathbf{c})) = \frac{1}{K}\sum_k \text{KL}\big[q^{(k)}_{\text{train}}(\mathbf{c})\big\|q_{\text{test}}(\mathbf{c})\big]. \tag{3.7}$$

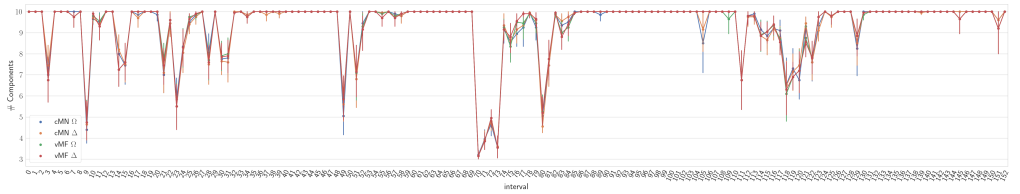
This is inspired by using the concentration levels as features for further modeling, but instead uses the full information captured by the variational distributions of them without the need for additional modeling steps. Both of these approaches where held up against using R2 based on the direct fitting algorithm as a heuristic for one-class classification. Using the R2 is similar to previous approaches using residuals.

The experiments are set up to consider each class in turn as the authentic samples for both the binary classification models and the one-class classification models. This results in 8 different authentic tasks across the 2 data sets. The performances are evaluated by cross-validated area-under-the-curve (AUC) of precision-recall (PR) curves. These are also normalized (N) by the non-achievable region (Boyd, Santos Costa, Davis, & Page, 2012), and compared to the expected performance of a random classifier referred to as the baseline performance. We also report the AUC of the receiver operating characteristics (ROC) curves for completeness. Below we discuss the results of these experiments.

An algorithm to fit the probabilistic PARAFAC2 using model selection based on the ELBO We describe and use an algorithm to automatically identify the number of components from the change in the ELBO. The algorithm incrementally adds more components until either a maximum number of components or a drop in the ELBO is encountered. As mentioned above this is not guaranteed to determine the correct number of components, but it can work in practice as shown by the simulated studies in [i/(Jørgensen et al., 2018)], where the maximum ELBO is reached for the true number of components. We also see that for the wine data many of the intervals use less than the maximum number of components as visualized in Figure 3.7. For the rice data, a large number of intervals use the maximum number of 10 components as also seen in the figure. While not all the intervals are resolved well for the data in this work, we do still get better results on the downstream tasks using this algorithm with the probabilistic PARAFAC2 models than using the flattened data. Future work should compare it to a more manual approach using an expert estimate of the number of components. The algorithm can also potentially be improved by using less strict stopping criteria than a decrease in ELBO such as a minimum required increase in the ELBO, or some approach for estimating the similarity between the components of the models to evaluate whether the increase in model order has been identified new components or overidentified existing ones by splitting them. Such a similarity could be based on the correlation between the components or the KL divergence.



(a)



(b)

Figure 3.7: Figure (a) and (b) show the number of components chosen by the probabilistic models estimated by the incremental algorithm described previously for each interval of the (a) wine and (b) rice data sets respectively. The probabilistic PARAFAC2 models are specified as vMF: von Mises-Fisher matrix distribution on $\boldsymbol{P}_k$; $c\mathcal{MN}$: constrained multivariate normal distribution on $\boldsymbol{P}_k$; Δ: homoscedastic noise model; Ω: heteroscedastic noise model. Figure from paper ii.

An evaluation of using the probabilistic PARAFAC2 as a feature extractor Analogous to the approaches performing feature extraction using the direct fitting PARAFAC2 mentioned above, we applied the probabilistic PARAFAC2 models to do the same thing. Our results on the wine and rice data sets shown in Figure 3.8 and Figure 3.9 suggest that the probabilistic models have the potential to extract features leading to better performance for the downstream task such as binary classification or one-class classification. Generally, they achieve better performance than the flattened data and with less variance compared to the features of the direct fitting algorithm. These results can have been influenced by the PARAFAC2 models being identified in local maxima, but the risk of attaining such solutions was reduced by using the best out of ten initializations for each model. Also, the direct fitting algorithm was set to use a model order equal to the minimum number of components used by any of the probabilistic models as a solution to the otherwise difficult task of model selection. This could have lead to suboptimal solutions for the direct fitting algorithm, but at the same time we choose the number of components for the probabilistic models based on the ELBO, which has no guarantee to be optimal either. It seemed like in [i/(Jørgensen et al., 2018)] that determining the model order through the ELBO would include too many components unless the PARAFAC2 assumptions were fulfilled. Nonetheless, the results did suggest that the probabilistic framework extract more robust features for the downstream tasks. We also compared our approach to one of simply flattening the data by summing over one of the modes. These were almost consistently outperformed by the PARAFAC2 extracted features except for the Flat_{ep} on the intervals of the rice data using the one-class SVM model.

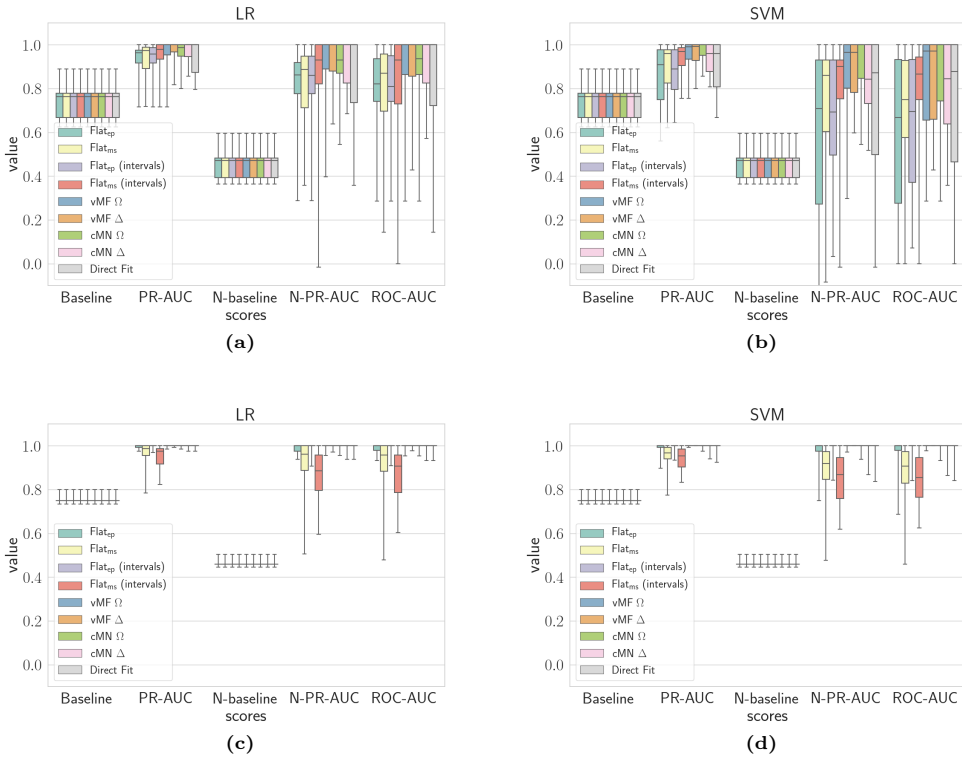


Figure 3.8: The top figures (a,b) show box plots of the cross-validated baseline, PR-AUC, N-baseline, N-PR-AUC and ROC-AUC of the LR and SVM models respectively on the different features extracted from the wine data. The bottom figures (c,d) report the same on the different features extracted from the rice data. The top and bottom of the whiskers are the minimum and maximum values in the results respectively. The legend shows flatten data as Flat_{ep}: total ion current (TIC) chromatogram; Flat_{ms}: summed over retention times; *intervals* indicates the data was flattened per interval - and the features of the probabilistic PARAFAC2 models as: vMF: von Mises-Fisher matrix distribution on $\boldsymbol{P}_k$; $c\mathcal{MN}$: constrained multivariate normal distribution on $\boldsymbol{P}_k$; Δ: homoscedastic noise model; Ω: heteroscedastic noise model - and finally the features of the direct fit PARAFAC2 algorithm. Figure from paper ii.

An approach for using the probabilistic PARAFAC2 as a one-class model The probabilistic PARAFAC2 framework provides approximated distributions of the model parameters, which can be used for further modeling and decision-making. We proposed to evaluate test samples on the training data components by only fitting the concentration matrix and the elution profiles on the test samples while keeping the training data components of the mass spectrum and elution profile correlations fixed. These decompositions for the test samples were compared

to the ones of the training samples to quantify the similarity between them. We used either the ELBO or computed an average over the KL divergence between the approximated distribution of the concentration levels of the test samples and those of each of the training samples. We compared this to using the direct fitting algorithm and estimating the explained variance for the test samples. These results can be seen in Figure 3.9 where the scores were aggregated across the intervals by summing them. Such an approach requires no additional modeling, which might be preferable for a more simplified workflow. Looking at the individual performance of the intervals, however, indicates that some of them might be better explained by the probabilistic PARAFAC2 models. This could become less of an issue if a better algorithm for automatically choosing the model order is employed, but could also be addressed by filtering out intervals that the PARAFAC2 model is unable to fit well. The ELBO show improved performance in comparison to using R2 with the direct fitting algorithm.

Some considerations for using Precision-Recall curves on small data Evaluating the performance of the models on the data under consideration, we end up with few positive samples and many negative samples in the test set — especially in the one-class classification. It is described by (Boyd et al., 2012; Davis & Goadrich, 2006) that the precision-recall (PR) curves are more informative than the receiver operating characteristic (ROC) curves for evaluating binary classifiers on imbalanced datasets, as PR curves may reveal bad performers when the ROC curves do not. (Saito & Rehmsmeier, 2015) discuss the unachievable region of PR curves, and recommend using the normalized *area under the curve* (AUC) of the PR curves when performing cross-validation on unbalanced data. We argue that it is advantageous to evaluate the models using the larger class as the positive samples, and it becomes especially important to take the unachievable region into account in this case. By doing both of those, we show visually that the area of the different classifiers becomes more evenly distributed. The PR-AUC scores also become more expressive by denoting the larger class — here the outliers — as the positive samples, as all of them are included in the computations of the evaluation. By using the inliers, or authentic samples, as the positive class, only the false positives of the outliers would be taken into account.

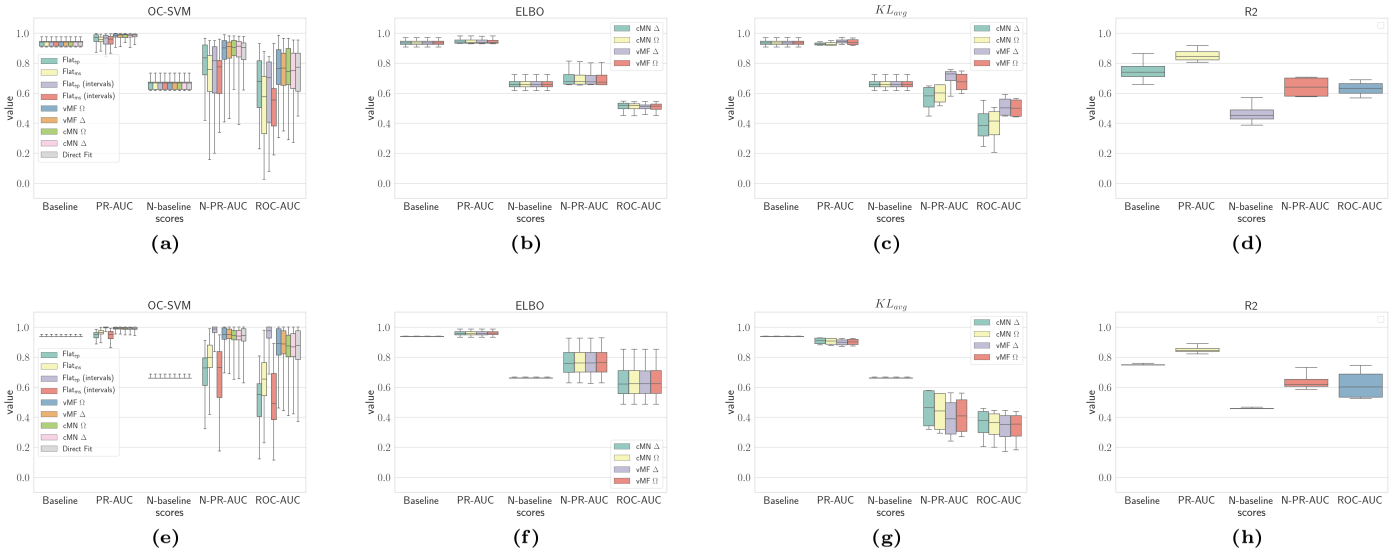


Figure 3.9: The figures (a,b,c,d) show box plots of the cross-validated baseline, PR-AUC, N-baseline, N-PR-AUC and ROC-AUC of the OC-SVM using the previously extracted features, as well as the ELBO computed as (3.6), the $\mathrm{KL}_{\mathrm{avg}}$ computed as (3.7) and R2 estimated using the different PARAFAC2 models evaluated on the wine data. The box plots are showing the values summed across all the intervals of the data set. The top and bottom of the whiskers are the minimum and maximum values in the results respectively. The bottom figures (e,f,g,h) report the same on the rice data. Legend lists the same features as in the previous figures for the OC-SVM models and the different probabilistic PARAFAC2 models for the ELBO and $\mathrm{KL}_{\mathrm{avg}}$. R2 is estimated based on the direct fit PARAFAC2. Figure from paper ii.

Chapter 4

Hierarchical Clustering Using Bayesian Methods for Online Learning

When performing authentication the user usually knows what to expect, but for other monitoring or explorative tasks that might not be the case. With no available labels or exact knowledge of what to look for, the number of classes and their patterns are undetermined. To gain a better understanding of such data one analysis it using clustering methods to discover groups from the patterns within the data.

4.1 Probabilistic Hierarchical Clustering

Hierarchical clustering usually takes an agglomerative approach which builds the clustering tree bottom-up. The tree is often binary and built by merging data points using some notion of similarity between them, which is commonly measured by some heuristic based on distances as previously described in section 2.2.2.1. An alternative to these would be to employ probabilistic methods, where a probability distribution over full hierarchies is specified(Aldous, 1996; Blundell, Teh, & Heller, 2012; Y. Hu, Ying, Daume, & Ying, 2013; Knowles & Ghahramani, 2015; Neal, 2003). Instead of learning a distribution over hierarchies, another probabilistic hierarchical clustering method (K. A. Heller & Ghahramani, 2005) models the data using a mixture model that adheres to a hierarchy. This hierarchy is built greedily bottom-up like HAC but uses the probability of merging pairs of nodes instead of distance measures. We describe this approach below.

4.1.1 Bayesian Hierarchical Clustering

The Bayesian Hierarchical Clustering (BHC) method introduced by (K. A. Heller & Ghahramani, 2005) uses probabilistic model comparisons to determine which nodes to pair and which parts of the hierarchy to consider as individual clusters. This model was shown to find hierarchies with a high level of dendrogram purities[1], and a good structure among the clusters, especially at the top levels. Two randomized algorithms were also introduced by the same authors(K. Heller & Ghahramani, 2005) to improve computational efficiency.

The algorithm builds a binary tree bottom-up by merging subsets of the data points which have the highest posterior probability of merging given as

$$r_k = \frac{\pi_k p(\mathcal{D}_k|\mathcal{H}_1^k)}{\pi_k p(\mathcal{D}_k|\mathcal{H}_1^k) + (1-\pi_k)p(\mathcal{D}_k|\mathcal{H}_2^k)} \tag{4.1}$$

These subsets $\mathcal{D}_k \subset \mathcal{D}$ consist of either a single data point or the data points at the leaves of a tree $\mathcal{T}_k$. The full data is the set of observations $\mathcal{D} = \{\boldsymbol{X}^{(1)}, \ldots, \boldsymbol{X}^{(n)}\}$. The i'th merge will result in a new root node of a tree $\mathcal{T}_i$ with the data points $d_i = d_l \cup d_r$ at its leaves as given by its left and right children.

$\mathcal{H}_1^k$ is the hypothesis of merging computed based on a specified probabilistic model $p(\boldsymbol{X}|\theta)$ and its prior $p(\theta|\beta)$. Assuming the data is identically and independently distributed its marginal distribution is

$$p(\mathcal{D}_k|\mathcal{H}_1^k) = \int \prod_{\boldsymbol{X}\in\mathcal{D}} p(\boldsymbol{X}|\theta)p(\theta|\beta)d\theta, \tag{4.2}$$

where it is assumed in the following that a conjugate prior is employed to make the computations efficient. The alternative hypothesis being $\mathcal{H}_2^k$ expresses the probability of any partition consistent with tree structures without merging the considered pair as

$$p(\mathcal{D}_k|\mathcal{H}_2^k) = p(\mathcal{D}_L|\mathcal{T}_L)p(\mathcal{D}_R|\mathcal{T}_R), \tag{4.3}$$

where $p(\mathcal{D}_k|\mathcal{T}_k) = \pi_k p(\mathcal{D}_k|\mathcal{H}_1^k) + (1-\pi_k)p(\mathcal{D}_k|\mathcal{H}_2^k)$ is the probability of the partitions of $\mathcal{D}_k$ as found in the tree $\mathcal{T}_k$. π_k is prior of the merging hypothesis $\mathcal{H}_1^k$ given as $\pi_k = \frac{\alpha\Gamma(n_k)}{d_k}$. Here $d_k = \alpha\Gamma(n_k) + d_{L_k}d_{R_k}$ and n_k is the number of data points in $\mathcal{T}_k$. d_{L_k} and d_{R_k} are the d_i of the left and right child of $\mathcal{T}_k$ respectively. α is the hyperparameter of the model, where the d_i of each leaf node is equal to this α along with a merging prior set as $\pi_i = 1$. The final result is that all nodes have been greedily merged into a single binary tree which requires the computation and comparison of $O(n^2)$ potential merges. A natural choice of where to cut the final tree to obtain a flat clustering would be $r_k < 0.5$. This is merges with a posterior probability lower than the posterior probability of the alternative hypothesis.

To improve the quadratic complexity, two randomized algorithms were proposed by the original authors(K. Heller & Ghahramani, 2005). The first randomly subsamples m data points from the data $\mathcal{D}$ before fitting a BHC model on them using the exact algorithm. The other $n-m$ data points are filtered through the top level of this estimated BHC model. Each of these data points is grouped with either the left or right subtree of the root based on the posterior merging

[1] see [iii/(Jørgensen, Hansen, Heskes, & Krijthe, 2020)] for a definition of this quantity.

probabilities. Now the algorithm performs a recursion on these two groups by applying the randomized algorithm to both sets. This procedure continues until the full tree is identified. The second algorithm also randomly subsamples m data points and uses these as the initial set of clusters. The other $n - m$ data points are then assigned to their most probable cluster. This clustering is refined with k steps of an EM algorithm. These m clusters can now be used as input to the full algorithm building a BHC model with the clusters as leaves instead of individual data points. Another approach for performing an initial clustering before fitting a BHC model to them was proposed by (Y. Xu, Heller, & Ghahramani, 2009). They use a Bayes K-means algorithm to do so. The randomized algorithms have a computational complexity of $O(n \log n)$ and $O(n)$ respectively. The second algorithm assumes a small number of clusters and a low number of subsamples drawn per step to obtain its linear scaling. (Darkins et al., 2013) investigates the trade-off between how well the first randomized algorithm approximates the exact algorithm and the run-time as a function of m and n.

The BHC model also allows for computing the predictive posterior probability of a new observation $\boldsymbol{X}_{t+1}$ under the model. The posterior probability of node k in the tree being a cluster is $\omega_k = r_k \prod_{i \in \mathcal{P}_k} (1 - r_i)$, where $\mathcal{P}_k$ is the set of nodes along the path from the parent of node k to the root node. Furthermore, the k'th node has the predictive distribution $p(\boldsymbol{X}_{t+1}|\mathcal{D}_k) = \int p(\boldsymbol{X}_{t+1}|\theta)p(\theta|\mathcal{D}_k, \beta)d\theta$. The posterior predictive distribution of the full tree becomes

$$p(\boldsymbol{X}_{t+1}|\mathcal{D}) = \sum_{k \in \mathcal{N}} \omega_k p(\boldsymbol{X}_{t+1}|\mathcal{D}_k), \tag{4.4}$$

where $\mathcal{N}$ is the set of all nodes in the tree. Computing this has a computational complexity of $O(n)$.

4.2 Online Hierarchical Clustering

Existing algorithms for online hierarchical clustering use ideas similar to the ones naturally arising from taking a probabilistic approach as we will see next. Hierarchical clustering can not only be updated online efficiently but also handle changing or new concepts. A new observation can be inserted into the hierarchy by identifying the subtree most similar to it from the top down. In case of a changed concept — when the underlying process generating a specific cluster has changed — the observation should ideally share ancestors with the data of the original concept, while if it is an entirely new concept, it should be placed high in the hierarchy having few ancestors. The issue of changing concepts is commonly referred to as concept drift, while the introduction of new classes is called novelties(Garcia et al., 2019; Widmer & Kubat, 1996; Pimentel et al., 2014).

Two important approaches include the Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH)(Zhang, Ramakrishnan, & Livny, 1996) and Purity Enhancing Rotations for Cluster Hierarchies (PERCH)(Kobren, Monath, Krishnamurthy, & McCallum, 2017b). These use ideas for controlling the memory footprint and updating costs by summarizing the information at all levels of the hierarchy. BIRCH uses sufficient statistics based on a chosen notion of distance as summaries. It routes the data points into the leaves while controlling the balance of the tree by limiting the number of branches. This ensures the updates are efficient. PERCH

uses bounding boxes to approximate the observed data throughout the hierarchy. It also keeps a balanced tree but does so by using rotations after inserting new data inspired by self-balancing binary trees(Knuth, 1998). When the tree becomes too large to fit in the available memory it can employ a collapsed mode by summarizing the node whose maximum distance between children is minimal using its sufficient statistics. Both of these models operate in Euclidean space. BIRCH was developed 25 years ago, and PERCH was shown to empirically outperform BIRCH(Kobren et al., 2017b).

PERCH optimizes its hierarchy with respect to an optimal flat clustering. Another approach is taken by (Menon et al., 2019) who proposes two incremental algorithms for hierarchical clustering. The first called Online Top Down (OTD) clustering provides a guarantee with respect to an optimal hierarchy, and the second called Online HAC (OHAC) approximates the conventional HAC structure. At the arrival of the $(k+1)$'th data point $\mathcal{D}_{k+1}$, OTD compares the similarity between the leaves of the current tree $\mathcal{T}_k$ to the similarity between the same leaves and $\mathcal{D}_{k+1}$. The root of $\mathcal{T}_k$ becomes siblings with $\mathcal{D}_{k+1}$ with their parent being the root of the new tree $\mathcal{T}_{k+1}$ if the former is larger. Otherwise, it recursively makes this comparison into the subtree $\mathcal{T}_i$ with the leaves most similar to $\mathcal{D}_{k+1}$. This can be computed in $O(d)$, with d being the depth of $\mathcal{T}_k$, with a linkage function computed in constant time. OHAC is proposed to be a tradeoff between computing a complete HAC for all the data at every new arrival and forcing the new data point to be placed naively next to its nearest neighbor among the leaves. Instead, it splits the hierarchy along the path from the nearest neighbor leaf to the root of the tree and uses the children of this path along with the new data point as input to HAC. The authors conclude that OHAC approximates the result of HAC well, but can be applied to much larger data as its complexity is linear with the number of data points.

4.3 Online Bayesian Hierarchical Clustering (paper iii)

In the paper [iii/(Jørgensen et al., 2020)] we developed an online algorithm for BHC to be able to take advantage of the improved clustering quality suggested by the original results(K. A. Heller & Ghahramani, 2005) and the probabilistic nature of the model to evaluate new data points in an online setting. Furthermore, we wanted to use inherent probabilities to determine how to collapse the hierarchy to limit its size.

It is assumed the data is generated by a stream delivering one data point at a time. We denote an existing BHC tree as $\mathcal{T}_t$ with t denoting the t'th iteration of data arrival. The first two data points are simply merged resulting in a tree $\mathcal{T}_2$ with 2 leaves. As the third and any subsequent data point arrives the algorithm is tasked with inserting it into the existing tree structure. An insertion requires determining in which part of the tree to place the new data point followed by breaking up and rebuilding that part. The part for insertion is determined by performing a binary search from the top of the tree for the leaf with the highest posterior predictive probability given by (4.4) computed only for the nodes of each subtree considered. After having identified such a leaf, the path from this leaf to the root of the existing tree $\mathcal{T}_t$ is deleted. The new tree $\mathcal{T}_{t+1}$ is then built from all the children of the deleted path and the new data point using the exact offline BHC algorithm.

This algorithm makes it possible to update an existing BHC tree given a new data point with

a complexity similar to the offline algorithm, but this is only true in the worst case. Updating shallow trees that are fairly well balanced in terms of the number of leaves of each subtree can be much faster. In such cases, the reconstruction of the BHC tree is performed on a number of inputs potentially much smaller than the total number of data points. So the complexity is $O(|leaves(\mathcal{T}_t)|^2)$ in the worst case with $|leaves(\mathcal{T}_t)|$ being the number of leaves in $\mathcal{T}_t$.

To improve the scaling of the algorithm and to limit its memory footprint, we further proposed to collapse subtrees of the BHC tree based on either of two criteria. The first (C1) collapses the subtree with the maximum r_k when the number of leaves in $\mathcal{T}_t$ reaches a hard limit ℓ_{max}. The second (C2) would collapse any subtrees with a r_k larger than a specified threshold $T_{collapse}$. (C1) limits the worst-case to be $O(\ell_{max}^2)$ for any data set. (C2) potentially decreases the number of leaves dramatically if most of the subtrees have a large r_k. Ideally, the number of leaves would match the most suitable number of clusters $m^\star$ for the given data resulting in the updates taking $O(m^{\star 2})$.

An example run of the algorithm can be seen in Figure 4.1, where the model is being fitted to a synthetic data set consisting of 8 clusters. The clusters are adapted as more as more observations become available.

We compared the online algorithm to the offline algorithm as well as the first randomized algorithm on synthetic data and a real data set. The likelihood function was a multivariate normal distribution with a conjugate prior set to a normal-inverse-Wishart throughout the experiments. The synthetic data are generated using a categorical distribution drawn from a symmetric Dirichlet distribution with hyperparameters $\alpha = \mathbf{1}_m$, where m is a specified number of clusters, and each component is drawn from a 2-dimensional normal-inverse-Wishart distribution with hyperparameters μ equal to $(0, 0)$, diagonal scale matrix Ψ equal to 50 along the diagonal, degrees of freedom ν equal to 10 and κ equal to 0.005. 10 realizations of data with 200 samples for m equal to $[1, 2, \ldots, 5, 10, 15, \ldots, 50]$ were generated resulting in 140 data sets. The real data was prepared by (Darkins et al., 2013), where they draw several realizations from a BHC model using Gaussian processes on a 169-gene subset of the cell cycle gene expression data of (Cho et al., 1998). This data have 999 samples with 13 clusters. The algorithms were evaluated on their run-times, the identified number of clusters, and the quality of the clusterings. The quality was quantified using log marginal likelihood of the BHC trees, the adjusted Rand index (ARI) and dendrogram purity (DP)(K. A. Heller & Ghahramani, 2005). The online and randomized algorithms were run for 10 different orderings of data sets.

The first experiment compared the offline and online algorithms. These were fitted to the synthetic data using the same values of the hyperparameters as those that generated them. The online model was not collapsed to ensure obtaining its best results. These results can be seen in Figure 4.4. To evaluate the gene expression data, the online algorithm was executed using either (C1) or (C2) for collapsing subtrees and compared to both the offline and randomized algorithms. In Figure 4.3 this comparison is shown for the different quality measures, the run-time and the identified number of clusters as a function of the parameters ℓ_{max}, $T_{collapse}$ and m controlling how well each type of algorithm could approximate the exact offline algorithm. Furthermore, we investigated the influence of the hyperparameters on the log marginal likelihood for both the offline, randomized and online algorithms as seen in Figure 4.2. The parameters of the approximate algorithms — the collapsed online and randomized algorithms — were kept fixed with $\ell_{max} = 200$, $T_{collapse} = 0.99$ and $m = 300$. These relatively high values were

used to obtain good approximations. The following settings of the hyperparameters were used; $\alpha \in [0.5, 1, 5, 10]$, $F_\psi \in [0.1, 0.5, 1, 2]$, $\nu \in [-1, 1, 5, 10, 20]$ and $\kappa \in [0.1, 1]$. The mean μ_0 was computed using the data. The diagonal of Ψ were computed to be a factor of the diagonal of the covariance matrix of the data by the parameter denoted F_ψ. The results of these experiments are described and discussed in the following.

An online algorithm for building a Bayesian Hierarchical Clustering model Before this work no online algorithm for the BHC model existed, but only offline versions in form of a greedy algorithm or a more efficient randomized algorithm. We propose an algorithm which given an existing BHC tree can insert a new data point into the hierarchy. Furthermore, we proposed to collapse subtrees to limit the size of the model for increased scalability and a limited memory footprint. Our results seen in Figure 4.2, Figure 4.3 and Figure 4.4 show that the online algorithm can achieve performance similar to the greedy offline algorithm; that the relationship between the performance and log marginal likelihood is more similar between the greedy offline and online algorithms than the randomized algorithm; that the log marginal likelihood of the models estimated using the online algorithm can be better than those of the greedy offline algorithm as the number of clusters increases. Future work should include a theoretical understanding of the relationship between the models and their performance.

Methods for increasing scalability and limiting the memory footprint of the model A central concern for both online algorithms and hierarchical clustering algorithms are their memory footprint. The BIRCH and PERCH models address this using ideas based on summarizing sets of data points by their mean and variance, or by describing a set of data points by the boundaries of their locations. These ideas naturally translate to the probabilistic models by using sufficient statistics, which under a Gaussian model would be the mean and variance. A decision on which part of the hierarchy to summarize can also be made based on the merging probabilities to ensure a memory limited is satisfied, or as a trade-off between computational efficiency and lower performance. This is done by describing a subtree in the hierarchy only by its sufficient statistics and discarding the data points belonging to that subtree. We proposed two ways of making this decision. The first approach puts an upper limit on the number of leaf nodes in the tree and summarizing the two leaf nodes with the highest probability of merging when this limit is reached. This is deterministic and therefore can be set to match the available hardware resources. The second approach merges any subtree with a merging probability larger than some threshold. The results, as shown in Figure 4.3, of the first method were very promising, as they show good performance at reduced run times. The second approach did show potential, however, its performance was very sensitive to the threshold, but this also comes from the hyperparameters influencing the merging probabilities. Automatic tuning of the hyperparameters could potentially lead to better results for this approach.

Improved scalability while more closely approximating the offline greedy algorithm We compared the online algorithm to the randomized algorithm that was proposed by the original authors(K. Heller & Ghahramani, 2005) as seen in Figure 4.3. The run time of both the online and the randomized algorithms are affected by their respective parameters controlling the amount of data considered at any time. Our results showed that the online algorithm

required less run time to attain better approximations to the exact algorithm compared to the randomized algorithm — both in terms of estimating an equal amount of clusters, achieving similar log marginal likelihoods and performance as measured by the DP and ARI.

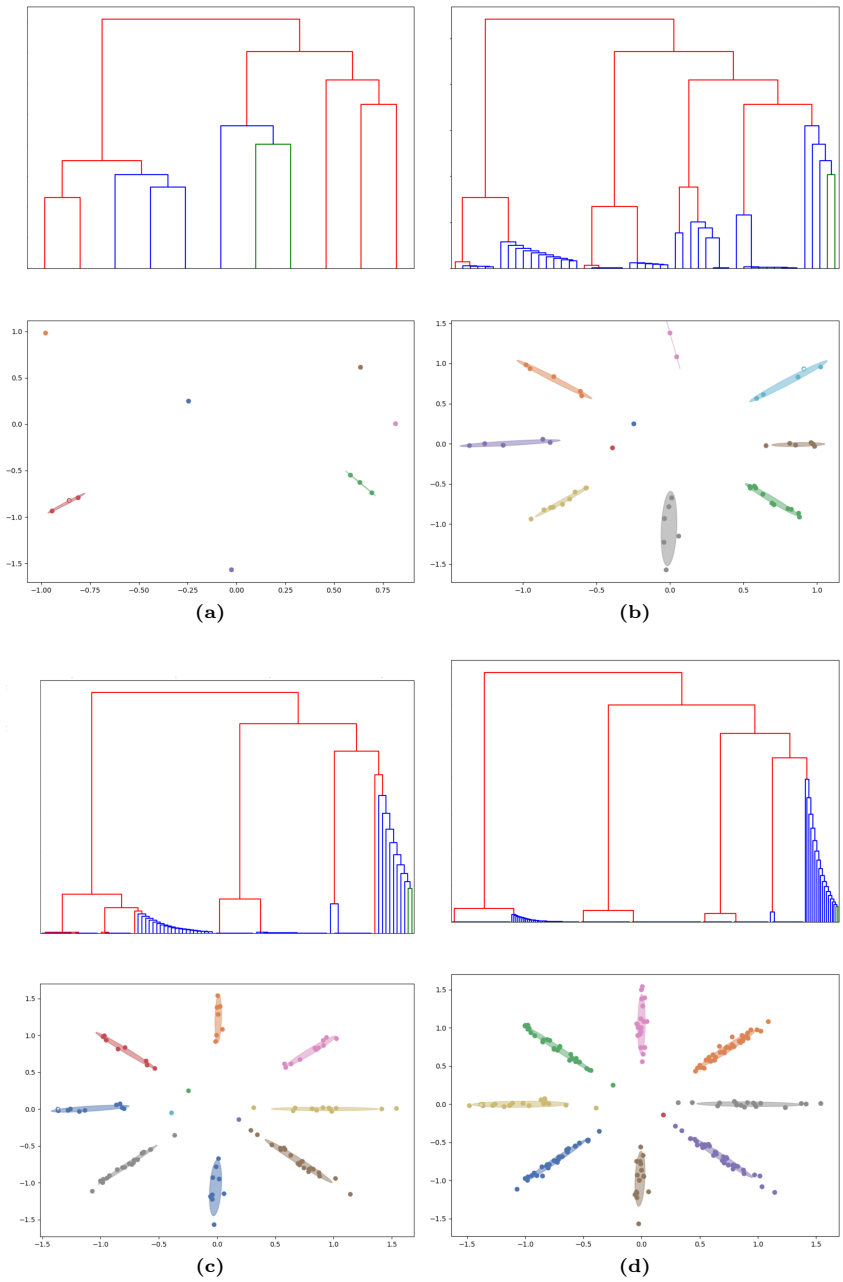


Figure 4.1: An example of the online BHC model on a toy data set consisting of 8 classes. The hierarchy and flat clustering of the model is shown after (a) 10 observations; (b) 50 observations; (c) 100 observations; (d) 200 observations.

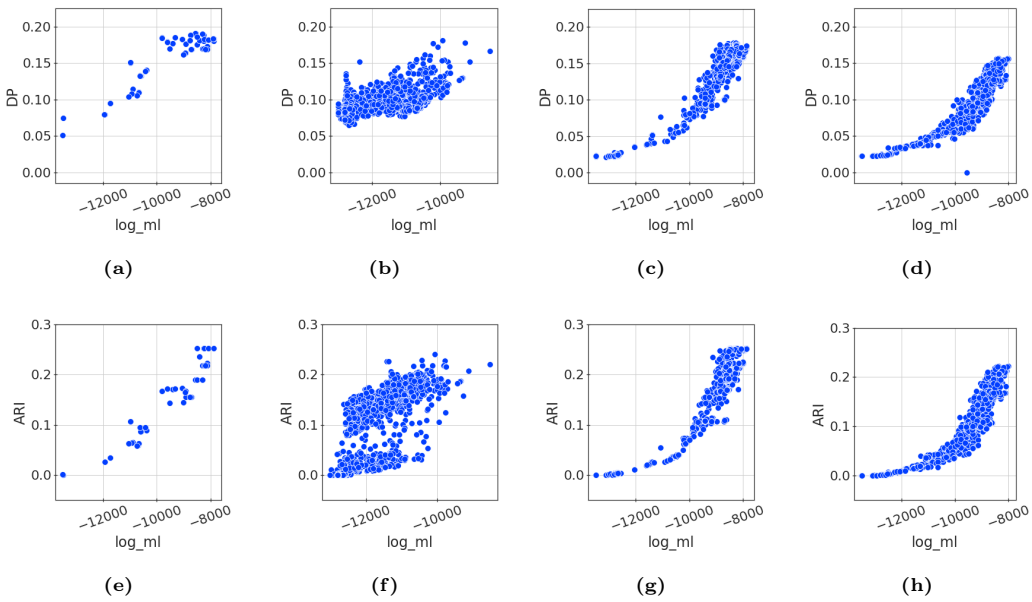


Figure 4.2: Resulting scores of DP and ARI for the offline algorithm (a,e), the randomized algorithm (b,f), the online algorithm using (C1) (c,g) and the online algorithm using (C2) (d,h) for varying settings of hyperparameters. Figure from paper iii.

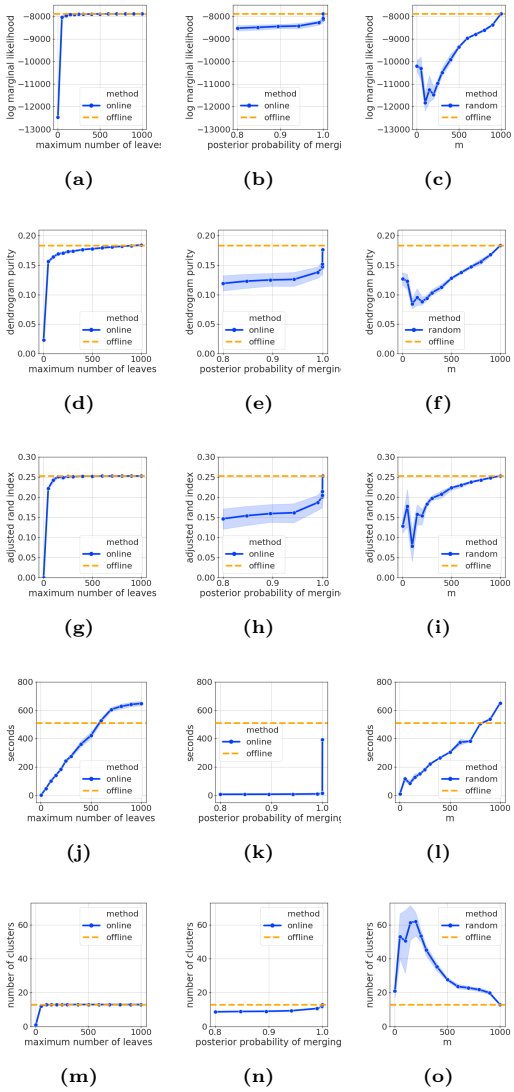
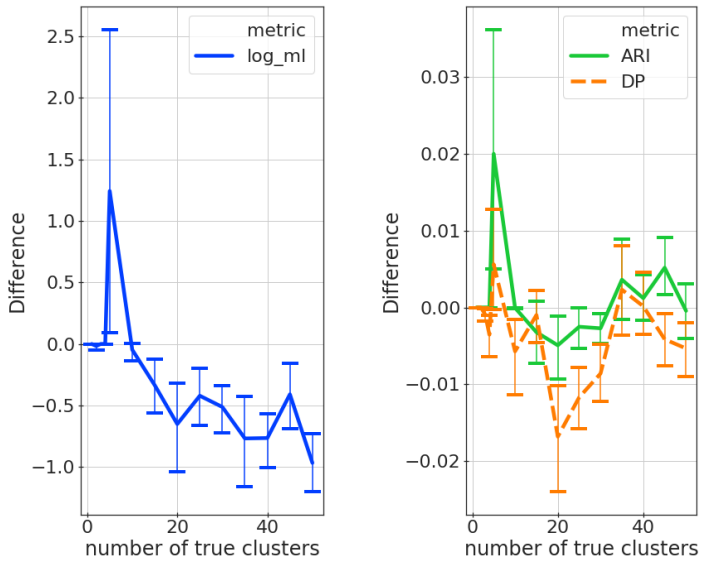


Figure 4.3: Comparison of the online algorithms to the randomized and offline algorithm showing from top to bottom: log marginal likelihood, DP, ARI, run time in seconds and number of clusters as suggested by the BHC model. Figure from paper iii.



(a) (b)

Figure 4.4: Difference between the offline and online BHC algorithms as measured by the log marginal likelihood in (a), and the scores ARI and DP in (b) on the synthetic GMM data. Figure from paper iii.

Chapter 5

Conclusion

With this thesis we set out to investigate and develop unsupervised methods using Bayesian inference for multiway modeling and clustering, specifically, the PARAFAC2 model and the BHC model. Below we answer the research question previously stated in the introduction followed by some remarks on future perspectives.

What are the merits of the probabilistic PARAFAC2 as opposed to conventional PARAFAC2 modeling not accounting for uncertainty?
We investigated the use of the probabilistic PARAFAC2, specifically on synthetic and chromatographic data, and saw that it has improved robustness to model misspecification and in settings of low SNR compared to the conventional direct fitting algorithm. Furthermore, the probabilistic framework enables new approaches to model selection including a new view of the model fit given by the ELBO, and the use of an ARD prior turning off excess components. Two formulations of the orthogonality constraint with different degrees of flexibility are specified in the framework. The vMF model is more constrained than the $c\mathcal{MN}$model which results in better recovery of an underlying PARAFAC2 structure as shown using simulated data. Finally, the probabilistic formulation included the option of employing a heteroscedastic noise model, which was found to make an important difference when dealing with data affected by this type of noise. The benefits of the probabilistic model are not limited to chromatographic data, but potentially any data type that can be described by the PARAFAC2 structure.

How does the probabilistic PARAFAC2 model compare to the conventional PARAFAC2 and methods for flattening multiway data as a preprocessing step for downstream tasks?
The PARAFAC2 model can transform multiway data into a matrix ready to be analyzed by

multivariate methods downstream. More sophisficated approaches to preprocessing and transforming the raw data to be used in downstream tasks than simply flattening it often require a trained end user to be effective. Our results showed that the probabilistic framework extracted features with higher performance and robustness for binary and one-class classification than those extracted using the conventional direct fitting algorithm. These two sets of features also generally performed better than simply flattening the multiway data by summing over one of the modes. The probabilistic PARAFAC2 combines the well-suited PARAFAC2 structure with the strong inference machinery of Bayesian statistics to push forward towards automating the process of processing multiway data for tasks like food authentictaion.

How can the probabilistic PARAFAC2 model be used for one-class classification in the context of food authentication?
The approximated distributions provided by the probabilistic PARAFAC2 framework were used to evaluate test samples directly. This was done using either the ELBO of the variational distribution on the test samples using the shared parameters obtained on training data, or an average KL divergence between concentration levels of the test and training samples. The latter was inspired of the approach used when performing feature extraction. The results showed some potential to perform one-class classification, but did not achieve the same level of performance as fitting a multivariate model to the extracted features. However, the ELBO performed better than using the explained variance based on the direct fitting algorithm similar which is similar to previous approaches based on the residuals.

How can Bayesian Hierarchical Clustering be advanced to an online setting where data arrives sequentially?
We developed an online algorithm to learn the BHC model, which not only allows for it to be used for data streams, but also showed an improved trade-off between its scalability and the resulting approximations to the greedy offline algorithm when compared to the randomized algorithm. An existing tree could be updated using the predictive posterior distribution of the model to determine the point of insertion by only partially rebuilding the hierarchical structure. The hyperparameters also affected the resulting log marginal likelihood of the online algorithm with greater similarity to that of the offline algorithm compared to the randomized algorithm. The new online algorithm makes all of the benefits of the BHC model available to be apply to data streams and larger data sets. This includes using different probabilistic models for different types of data and processes, providing a principled estimate of the number of clusters and creating quality clusterings.

How can an online Bayesian Hierarchical Clustering be scaled?
The hierarchical clustering tree provided by BHC leads to efficient updates and summaries of the data through sufficient statistics similar to the approaches taken by existing online hierarchical clustering algorithms such as PERCH, OTD and OHAC. Subtrees of high certainty within the hierarchical structure can be collapsed as needed to not exceed a specified available memory. Collapsing subtrees in the model not only reduces its memory usage but also makes updates more efficient as fewer levels in the hierarchy of clusters have to be investigated.

5.1 Future Work

The Probabilistic PARAFAC2 will hopefully find many applications given its added benefits compared to the conventional PARAFAC2. Specifically, the update of the prior based on the $c\mathcal{MN}$distribution to handle orthogonality was derived as part of the work, and thus, it has the potential to be used as part of other models requiring orthogonality on its mean parameters. Furthermore, a comparison between the probabilistic PARAFAC2 and a state-of-the-art manual process to perform the same preprocessing for downstream tasks would be of great interest to fully determine the power of the automated process enabled by the probabilistic framework. This requires an expert able to perform such an analysis. Additional work is needed to fully explore the potential of performing one-class classification based on the probabilistic PARAFAC2 for analyzing a complete multiway chromatographic data set. The intervals that are troublesome for the PARAFAC2 model could be identified and filtered out, which might be possible evaluating multiple fits of the PARAFAC2 model using a split-half analysis(Bro et al., 1999), or some further modeling as done using deep learning to identify meaningful components in (Risum & Bro, 2019).

Our experiments using the OBHC model empirically demonstrate the potential of the algorithm, but future work should include building a theoretical understanding of the relationship between the offline and online models. It could also be to understand how this online algorithm relates to online algorithms for learning Dirichlet process mixture models(Hughes & Sudderth, 2013; Lin, 2013), as the offline algorithm already approximates such models. As the model uses a tree structure, it also has the potential to be further optimized by performing concurrent updates(Kung & Lehman, 1980), which might make it viable for very large data sets. Testing the OBHC on large data streams would then be very interesting including a comparison to the existing algorithms able to handle such streams.

Through the development and investigation of these algorithms for multiway modeling and clustering using Bayesian statistics, we hope that they will find many applications and with their innovations assist in solving problems that were out of the scope of this work. These general methods have the potential to impact many areas of societal importance by automating aspects of the tasks that would usually require an expert to perform the decision-making.

Bibliography

Acar, E., & Yener, B. (2009). Unsupervised Multiway Data Analysis: A Literature Survey. *IEEE Trans. Knowl. Data Eng.*, *21*(1), 6–20.

Ahmed, N. (2015, November). Recent review on image clustering. *IET Image Proc.*, *9*(11), 1020–1032.

Aldous, D. (1996). Probability Distributions on Cladograms. In *Random Discrete Structures* (pp. 1–18). Springer New York.

Amante, E., Salomone, A., Alladio, E., Vincenti, M., Porpiglia, F., & Bro, R. (2019, August). Untargeted Metabolomic Profile for the Detection of Prostate Carcinoma-Preliminary Results from PARAFAC2 and PLS-DA Models. *Molecules*, *24*(17).

Amigo, J. M., Popielarz, M. J., Callejón, R. M., Morales, M. L., Troncoso, A. M., Petersen, M. A., & Toldam-Andersen, T. B. (2010). Comprehensive analysis of chromatographic data by using PARAFAC2 and principal components analysis. *J. Chromatogr. A*, *1217*(26), 4422–4429.

Amigo, J. M., Skov, T., Bro, R., Coello, J., & Maspoch, S. (2008). Solving GC-MS problems with PARAFAC2. *TrAC - Trends in Analytical Chemistry*, *27*(8), 714–725.

Barber, D., & van Laar, P. d. (2011, May). Variational Cumulant Expansions for Intractable Distributions.

Bayes, T. (1764). An essay towards solving a problem in the doctrine of chances. *Philosophical Transactions of the Royal Society of London*(53), 370–418.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*.

Blei, D. M., Kucukelbir, A., & McAuliffe, J. D. (2016, January). Variational Inference: A Review for Statisticians. , 1–33.

Blundell, C., & Teh, Y. W. (2013). Bayesian hierarchical community discovery. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems* (Vol. 26). Curran Associates, Inc. Retrieved from `https://proceedings.neurips.cc/paper/2013/file/35cf8659cfcb13224cbd47863a34fc58-Paper.pdf`

Blundell, C., Teh, Y. W., & Heller, K. A. (2012, March). Bayesian Rose Trees.

Boyd, K., Santos Costa, V., Davis, J., & Page, C. D. (2012, December). Unachievable Region in Precision-Recall Space and Its Effect on Empirical Evaluation. *Proc. Int. Conf. Mach.*

Learn., *2012*, 349.

Bro, R. (1997). PARAFAC. Tutorial and applications. *Chemometrics Intellig. Lab. Syst.*, *38*(2), 149–171.

Bro, R. (1998). Multi-way Analysis in the Food Industry - Models, Algorithms, and Applications. In *MRI, EPG and EMA," Proc ICSLP 2000.*

Bro, R. (2006, December). Review on Multiway Analysis in Chemistry—2000–2005. *Crit. Rev. Anal. Chem.*, *36*(3-4), 279–293.

Bro, R., Andersson, C. A., & Kiers, H. A. L. (1999). PARAFAC2—Part II. Modeling chromatographic data with retention time shifts. *J. Chemom.*, *13*(3-4), 295–309.

Bro, R., & Kiers, H. A. L. (2003). A new efficient method for determining the number of components in PARAFAC models. *J. Chemom.*, *17*(5), 274–286.

Carroll, J. D., & Chang, J. J. (1970). Analysis of individual differences in multidimensional scaling via an n-way generalization of "Eckart-Young" decomposition. *Psychometrika*, *35*(3), 283–319.

Cattell, R. B. (1944). Parallel Proportional Profiles" and Other Principles for Determining The Choice of Factors by Rotation. , *9*(4), 267–283.

Charikar, M., Chatziafratis, V., Niazadeh, R., & Yaroslavtsev, G. (2019). Hierarchical Clustering for Euclidean Data. In K. Chaudhuri & M. Sugiyama (Eds.), *Proceedings of Machine Learning Research* (Vol. 89, pp. 2721–2730). PMLR.

Cheng, Y., & Haardt, M. (2019, April). Enhanced Direct Fitting Algorithms for PARAFAC2 With Algebraic Ingredients. *IEEE Signal Process. Lett.*, *26*(4), 533–537.

Cho, R. J., Campbell, M. J., Winzeler, E. A., Steinmetz, L., Conway, A., Wodicka, L., ... Davis, R. W. (1998, July). A genome-wide transcriptional analysis of the mitotic cell cycle. *Mol. Cell*, *2*(1), 65–73.

Chu, W., & Ghahramani, Z. (2009). Probabilistic Models for Incomplete Multi-dimensional Arrays. In D. van Dyk & M. Welling (Eds.), *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics* (Vol. 5, pp. 89–96). Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA: PMLR.

Cohen, J. E., & Bro, R. (2018, February). Nonnegative PARAFAC2: a flexible coupling approach.

Coleman, G. B., & Andrews, H. C. (1979, May). Image segmentation by clustering. *Proc. IEEE*, *67*(5), 773–785.

Cox, R. T. (1946, January). Probability, frequency and reasonable expectation. *Am. J. Phys.*, *14*(1), 1–13.

Cserháti, T., Forgács, E., Deyl, Z., & Miksik, I. (2005, April). Chromatography in authenticity and traceability tests of vegetable oils and dairy products: a review. *Biomed. Chromatogr.*, *19*(3), 183–190.

Danezis, G. P., Tsagkaris, A. S., Camin, F., Brusic, V., & Georgiou, C. A. (2016, December). Food authentication: Techniques, trends & emerging approaches. *Trends Analyt. Chem.*, *85*, 123–132.

Darkins, R., Cooke, E. J., Ghahramani, Z., Kirk, P. D. W., Wild, D. L., & Savage, R. S. (2013, April). Accelerating Bayesian hierarchical clustering of time series data with a randomised algorithm. *PLoS One*, *8*(4), e59795.

Davis, J., & Goadrich, M. (2006). *The relationship between Precision-Recall and ROC curves.*

Ebrahimi, D., & Hibbert, D. B. (2008, July). Identification of sources of diesel oil spills using parallel factor analysis: a bridge between American Society for Testing and Materials and Nordtest methods. *J. Chromatogr. A*, *1198-1199*, 181–187.

Elkan, C., & Noto, K. (2008, August). Learning classifiers from only positive and unlabeled data. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 213–220). New York, NY, USA: Association for Computing Machinery.

Eric Stauffer, Julia A. Dolan, Reta Newman. (2008). CHAPTER 8 - Gas Chromatography and Gas Chromatography—Mass Spectrometry. In Eric Stauffer, Julia A. Dolan, Reta Newman (Ed.), *Fire Debris Analysis* (pp. 235–293). Burlington: Academic Press.

Esslinger, S., Riedl, J., & Fauhl-Hassek, C. (2014, June). Potential and limitations of non-targeted fingerprinting for authentication of food in official control. *Food Res. Int.*, *60*, 189–204.

Fanaee-T, H., & Gama, J. (2016, April). Tensor-based anomaly detection: An interdisciplinary survey. *Knowledge-Based Systems*, *98*, 130–147.

Fink, D. (1997). A compendium of conjugate priors. *See http://www. people. cornell. edu/pages/df36*.

Garcia, K. D., Poel, M., Kok, J. N., & de Carvalho, A. C. P. L. F. (2019). Online Clustering for Novelty Detection and Concept Drift in Data Streams. In *Progress in Artificial Intelligence* (pp. 448–459). Springer International Publishing.

Gelman, A. (2008, September). Objections to Bayesian statistics. *Bayesian Anal.*, *3*(3), 445–449.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2013). *Bayesian Data Analysis, Third Edition*. CRC Press.

Ghahramani, Z. (2004). Unsupervised Learning. In O. Bousquet, U. von Luxburg, & G. Rätsch (Eds.), *Advanced Lectures on Machine Learning: ML Summer Schools 2003, Canberra, Australia, February 2 - 14, 2003, Tübingen, Germany, August 4 - 16, 2003, Revised Lectures* (pp. 72–112). Berlin, Heidelberg: Springer Berlin Heidelberg.

Ghahramani, Z. (2015, May). Probabilistic machine learning and artificial intelligence. *Nature*, *521*(7553), 452–459.

Green, B. F. (1952). The Orthogonal Approximation of An Oblique Staructre in Factor Analysis. *Psychometrika*, *17*(4), 429–440.

Group, C., Science, F., & Veterinary, T. R. (1999). Parafac2—part ii. modeling chromatographic data with retention time shifts. , *309*(July 1998), 295–309.

Guha, S., Meyerson, A., Mishra, N., Motwani, R., & O'Callaghan, L. (2003, May). Clustering data streams: Theory and practice. *IEEE Trans. Knowl. Data Eng.*, *15*(3), 515–528.

Guha, S., & Mishra, N. (2016). Clustering Data Streams. In M. Garofalakis, J. Gehrke, & R. Rastogi (Eds.), *Data Stream Management: Processing High-Speed Data Streams* (pp. 169–187). Berlin, Heidelberg: Springer Berlin Heidelberg.

Han, S., & Dunson, D. B. (2018, March). Multiresolution Tensor Decomposition for Multiple Spatial Passing Networks.

Harshman, R. a. (1970). Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multimodal factor analysis. *UCLA Working Papers in Phonetics*, *16*(10), 1–84.

Harshman, R. A. (1972). PARAFAC2: Mathematical and technical notes. *UCLA Working Papers in Phonetics*, *22*(10), 30–44.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning. *Elements*, *1*, 337–387.

Heller, K., & Ghahramani, Z. (2005). Randomized algorithms for fast Bayesian hierarchical clustering. In *PASCAL Workshop on Statistics and Optimization of Clustering* (Vol. 25, pp. 1–22).

Heller, K. A., & Ghahramani, Z. (2005). Bayesian Hierarchical Clustering. In *Proceedings of the 22Nd International Conference on Machine Learning* (pp. 297–304). New York, NY, USA: ACM.

Hennig, C. (2015, October). What are the true clusters? *Pattern Recognit. Lett.*, *64*, 53–62.

Hinrich, J. L., Madsen, K. H., & Mørup, M. (2020, June). The probabilistic tensor decomposition toolbox. *Mach. Learn.: Sci. Technol.*, *1*(2), 025011.

Hodge, V. J., & Austin, J. (2004, October). A survey of outlier detection methodologies. *Artificial Intelligence Review*, 41.

Hu, C., Tohge, T., Chan, S.-A., Song, Y., Rao, J., Cui, B., … Shi, J. (2016, February). Identification of Conserved and Diverse Metabolic Shifts during Rice Grain Development. *Sci. Rep.*, *6*, 20942.

Hu, Y., Ying, J. L., Daume, H., III, & Ying, Z. I. (2013). Binary to Bushy: Bayesian Hierarchical Clustering with the Beta Coalescent. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, & K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems* (Vol. 26). Curran Associates, Inc.

Hughes, M. C., & Sudderth, E. (2013). Memoized Online Variational Inference for Dirichlet Process Mixture Models. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, & K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems 26* (pp. 1133–1141). Curran Associates, Inc.

Jain, A. K., Narasimha Murty, M., & Flynn, P. J. (1999, September). Data clustering: a review. ACM Comput Surv. *ACM Computing Surveys*, *31*(3), 264–323.

Jaynes, E. T. (2012). *Probability theory: The logic of science* (G. L. Bretthorst, Ed.). Cambridge, England: Cambridge University Press.

Johnsen, L. G., Amigo, J. M., Skov, T., & Bro, R. (2014). Automated resolution of overlapping peaks in chromatographic data. *J. Chemom.*, *28*(2), 71–82.

Johnsen, L. G., Skou, P. B., Khakimov, B., & Bro, R. (2017, June). Gas chromatography - mass spectrometry data processing made easy. *J. Chromatogr. A*, *1503*, 57–64.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., & Saul, L. K. (1999, November). An Introduction to Variational Methods for Graphical Models. *Mach. Learn.*, *37*(2), 183–233.

Jorgensen, P. H., Morup, M., Schmidt, M. N., & Herlau, T. (2016, September). Bayesian latent feature modeling for modeling bipartite networks with overlapping groups. In *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE.

Jørgensen, P. J. H., Hansen, L. K., Heskes, T., & Krijthe, J. H. (2020). *Online Bayesian Hierarchical Clustering.*

Jørgensen, P. J. H., & Mørup, M. (2020). *Classification with the Probabilistic PARAFAC2 for Food Authentication.*

Jørgensen, P. J. H., Nielsen, S. F. V., Hinrich, J. L., Schmidt, M. N., Madsen, K. H., & Mørup, M. (2018, June). Probabilistic PARAFAC2.

Jørgensen, P. J. H., Nielsen, S. F. V., Hinrich, J. L., Schmidt, M. N., Madsen, K. H., & Mørup, M. (2019). Analysis of Chromatographic Data using the Probabilistic PARAFAC2. In *Proceedings of Second Workshop on Machine Learning and the Physical Sciences.*

Kamstrup-Nielsen, M. H., Johnsen, L. G., & Bro, R. (2013a). Core consistency diagnostic in PARAFAC2. *J. Chemom.*, *27*(5), 99–105.

Kamstrup-Nielsen, M. H., Johnsen, L. G., & Bro, R. (2013b). Core consistency diagnostic in PARAFAC2. *J. Chemom.*, *27*(5), 99–105.

Kiers, H. A. L. (1998, May). A three-step algorithm for CANDECOMP/PARAFAC analysis of large data sets with multicollinearity. *J. Chemom.*, *12*(3), 155–171.

Kiers, H. A. L., Ten Berge, J. M. F., & Bro, R. (1999). PARAFAC2 — Part I . A Direct Fitting Algorithm for the PARAFAC2 Model. *J. Chemom.*, *13*, 275–294.

Knowles, D. A., & Ghahramani, Z. (2015, February). Pitman Yor Diffusion Trees for Bayesian Hierarchical Clustering. *IEEE Trans. Pattern Anal. Mach. Intell.*, *37*(2), 271–289.

Knuth, D. E. (1998). *The Art of Computer Programming: Volume 3: Sorting and Searching*. Addison-Wesley Professional.

Kobren, A., Monath, N., Krishnamurthy, A., & McCallum, A. (2017a). A Hierarchical Algorithm for Extreme Clustering. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 255–264). New York, NY, USA: ACM.

Kobren, A., Monath, N., Krishnamurthy, A., & McCallum, A. (2017b, August). A Hierarchical Algorithm for Extreme Clustering. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 255–264). New York, NY, USA: Association for Computing Machinery.

Kroonenberg, P. M. (2007). *Applied Multiway Data Analysis*.

Kung, H. T., & Lehman, P. L. (1980). Concurrent Manipulation of Binary Search Trees. *No.*, *3*, 354–382.

Lavine, B. K., & Workman, J. (2002). Chemometrics. *Anal. Chem.*, *74*(12), 2763–2769.

Leisink, M. A., & Kappen, H. J. (2001, September). A tighter bound for graphical models. *Neural Comput.*, *13*(9), 2149–2171.

Lenhardt, L., Zekovic, I., Tatjana, D., & Dramicanin, M. D. (2018, January). Modeling Food Fluorescence with PARAFAC: From Basics to Medical Applications. In *Calcium-Binding Proteins of the EF-Hand Superfamily* (pp. 161–197). unknown.

Lin, D. (2013). Online Learning of Nonparametric Mixture Models via Sequential Variational Approximation. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, & K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems 26* (pp. 395–403). Curran Associates, Inc.

Luo, L., Chen, Y., Bao, S., & Tong, C. (2019, December). Sparse PARAFAC2 decomposition: Application to fault detection and diagnosis in batch processes. *Chemometrics Intellig. Lab. Syst.*, *195*, 103893.

MacCuish, J. D., & MacCuish, N. E. (2010). *Clustering in Bioinformatics and Drug Discovery*. CRC Press.

MacKay, D. J. C. (1992, May). A practical Bayesian framework for backpropagation networks. *Neural Comput.*, *4*(3), 448–472.

Menon, A. K., Rajagopalan, A., Sumengen, B., Citovsky, G., Cao, Q., & Kumar, S. (2019, September). Online Hierarchical Clustering Approximations.

Minter, T. C. (1975). Single-class classification. In *LARS Symposia* (p. 54). docs.lib.purdue.edu.

Mittal, M., Goyal, L. M., Hemanth, D. J., & Sethi, J. K. (2019, May). Clustering approaches for high-dimensional databases: A review. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, *9*(3), e1300.

Mørup, M. (2011). Applications of tensor (multiway array) factorizations and decompositions in data mining. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, *1*(1), 24–40.

Mørup, M., & Hansen, L. K. (2009). Automatic relevance determination for multi-way models. *Journal of Chemometrics: A Journal of the Chemometrics Society*, *23*(7-8), 352–363.

Moya, M. M., & Hush, D. R. (1996). Network Constraints and Multi-objective Optimization for One-class Classification. *Neural Netw.*, *9*(3), 463–474.

Murphy, K. P. (2007). Conjugate Bayesian analysis of the Gaussian distribution. *DEF*, *1*(2σ2), 16.

Murphy, K. P. (2012). *Machine learning: A probabilistic perspective*. The MIT Press.

Murphy, K. P. (2021). *Probabilistic machine learning: An introduction*. MIT Press. Retrieved from `probml.ai`

Murtagh, F., & Contreras, P. (2012, January). Algorithms for hierarchical clustering: an overview. *WIREs Data Mining Knowl Discov*, *2*(1), 86–97.

Neal, R. M. (1996). *Bayesian Learning for Neural Networks*. Springer New York.

Neal, R. M. (2003). Density modeling and clustering using Dirichlet diffusion trees. *Bayesian statistics*, *7*, 619–629.

Nielsen, F. B. (2004). *Variational approach to factor analysis and related models* (Unpublished master's thesis).

Oliveri, P., & Downey, G. (2012, May). Multivariate class modeling for the verification of food-authenticity claims. *Trends Analyt. Chem.*, *35*, 74–86.

Oliveri, P., López, M. I., Casolino, M. C., Ruisánchez, I., Callao, M. P., Medini, L., & Lanteri, S. (2014, December). Partial least squares density modeling (PLS-DM) - a new class-modeling strategy applied to the authentication of olives in brine by near-infrared spectroscopy. *Anal. Chim. Acta*, *851*, 30–36.

Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.*, *22*(10), 1345–1359.

Patricia, N., & Caputo, B. (2014, June). Learning to Learn, from Transfer Learning to Domain Adaptation: A Unifying Perspective. In *2014 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1442–1449).

Perros, I., Papalexakis, E. E., Wang, F., Vuduc, R., Searles, E., Thompson, M., & Sun, J. (2017, August). SPARTan: Scalable PARAFAC2 for Large & Sparse Data. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 375–384). New York, NY, USA: Association for Computing Machinery.

Pimentel, M. A. F., Clifton, D. A., Clifton, L., & Tarassenko, L. (2014, June). A review of novelty detection. *Signal Processing*, *99*, 215–249.

Raiffa, H., & Schlaifer, R. (1961). *Applied statistical decision theory [by] howard raiffa and robert schlaifer* [Book]. Division of Research, Graduate School of Business Adminitration, Harvard University Boston.

Risum, A. B., & Bro, R. (2019, November). Using deep learning to evaluate peaks in chromatographic data. *Talanta*, *204*, 255–260.

Rodionova, O. Y., Oliveri, P., & Pomerantsev, A. L. (2016, December). Rigorous and compliant approaches to one-class classification. *Chemometrics Intellig. Lab. Syst.*, *159*, 89–96.

Sadik, S., & Gruenwald, L. (2011, September). Online outlier detection for data streams. In *Proceedings of the 15th Symposium on International Database Engineering & Applications* (pp. 88–96). ACM.

Saito, T., & Rehmsmeier, M. (2015, March). The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLoS One*, *10*(3), e0118432.

Sales Martínez, C., Portolés Nicolau, T., Johnsen, L. G., Danielsen, M., & Beltrán Arandes, J. (2019). Olive oil quality classification and measurement of its organoleptic attributes by untargeted GC–MS and multivariate statistical-based approach.

Schein, A., Zhou, M., Blei, D. M., & Wallach, H. (2016, June). Bayesian Poisson Tucker decomposition for learning the structure of international relations.

Settles, B. (2009). *Active learning literature survey* (Tech. Rep.). University of Wisconsin-Madison Department of Computer Sciences.

Sirén, K., Fischer, U., & Vestner, J. (2019, March). Automated supervised learning pipeline for non-targeted GC-MS data analysis. *Analytica Chimica Acta: X*, *1*, 100005.

Skov, T., Ballabio, D., & Bro, R. (2008, May). Multiblock variance partitioning: A new approach for comparing variation in multiple data blocks. *Anal. Chim. Acta*, *615*(1), 18–29.

Sneath, P. H. A., Sokal, R. R., & Others. (1973). *Numerical taxonomy. The principles and practice of numerical classification.* cabdirect.org.

Srivastava, A. N., & Sahami, M. (2009). *Text Mining: Classification, Clustering, and Applications.* CRC Press.

Tax, D. M. J. (2002). One-class classification: Concept learning in the absence of counter-examples.

Tian, K., Wu, L., Min, S., & Bro, R. (2018, August). Geometric search: A new approach for fitting PARAFAC2 models on GC-MS data. *Talanta*, *185*, 378–386.

Tipping, M. E. (2001). *Sparse Bayesian learning and the relevance vector machine.* `https://www.jmlr.org/papers/volume1/tipping01a/tipping01a.pdf`. (Accessed: 2021-3-2)

Tipping, M. E. (2004). Bayesian Inference: An Introduction to Principles and Practice in Machine Learning. In O. Bousquet, U. von Luxburg, & G. Rätsch (Eds.), *Advanced Lectures on Machine Learning: ML Summer Schools 2003, Canberra, Australia, February 2 - 14, 2003, Tübingen, Germany, August 4 - 16, 2003, Revised Lectures* (pp. 41–62). Berlin, Heidelberg: Springer Berlin Heidelberg.

Toraman, H. E., Abrahamsson, V., Vanholme, R., Van Acker, R., Ronsse, F., Pilate, G., ... Marin, G. B. (2018, January). Application of Py-GC/MS coupled with PARAFAC2 and PLS-DA to study fast pyrolysis of genetically engineered poplars. *J. Anal. Appl. Pyrolysis*, *129*, 101–111.

Tucker, L. R. (1966). Some mathematical notes on three-mode factor analysis. *Psychometrika*, *31*(3), 279–311.

van Engelen, J. E., & Hoos, H. H. (2020, February). A survey on semi-supervised learning. *Mach. Learn.*, *109*(2), 373–440.

Van Horn, K. S. (2003, September). Constructing a logic of plausible inference: a guide to Cox's theorem. *Int. J. Approx. Reason.*, *34*(1), 3–24.

Vestner, J., de Revel, G., Krieger-Weber, S., Rauhut, D., du Toit, M., & de Villiers, A. (2016, March). Toward automated chromatographic fingerprinting: A non-alignment approach to gas chromatography mass spectrometry data. *Anal. Chim. Acta*, *911*, 42–58.

Widmer, G., & Kubat, M. (1996, April). Learning in the Presence of Concept Drift and Hidden Contexts. *Mach. Learn.*, *23*(1), 69–101.

Wilde, A. (2019). *Detection of food fraud in high value products - exemplary authentication studies on vanilla, black pepper and bergamot oil* (Unpublished doctoral dissertation).

Wise, B. M., Gallagher, N. B., & Martin, E. B. (2001, May). Application of PARAFAC2 to fault detection and diagnosis in semiconductor etch. *J. Chemom.*, *15*(4), 285–298.

Xu, R., & Wunsch, D., 2nd. (2005, May). Survey of clustering algorithms. *IEEE Trans. Neural Netw.*, *16*(3), 645–678.

Xu, Y., Heller, K., & Ghahramani, Z. (2009). Tree-Based Inference for Dirichlet Process Mixtures. In D. van Dyk & M. Welling (Eds.), *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics* (Vol. 5, pp. 623–630). Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA: PMLR.

Yang, Y., & Dunson, D. B. (2016, August). Bayesian Conditional Tensor Factorizations for High-Dimensional Classification. *J. Am. Stat. Assoc.*, *111*(514), 656–669.

Yi, L., Dong, N., Yun, Y., Deng, B., Ren, D., Liu, S., & Liang, Y. (2016, March). Chemometric methods in data processing of mass spectrometry-based metabolomics: A review. *Anal.*

Chim. Acta, *914*, 17–34.

Zhang, T., Ramakrishnan, R., & Livny, M. (1996). BIRCH: an efficient data clustering method for very large databases. *ACM Sigmod Record*.

Zimek, A., & Schubert, E. (2017). Outlier Detection. In L. Liu & M. T. Özsu (Eds.), *Encyclopedia of Database Systems* (pp. 1–5). New York, NY: Springer New York.

Papers

Probabilistic PARAFAC2

Philip J. H. Jørgensen, Søren F. V. Nielsen, Jesper L. Hinrich, Mikkel N. Schmidt, Kristoffer H. Madsen, and Morten Mørup

Abstract—The PARAFAC2 is a multimodal factor analysis model suitable for analyzing multi-way data when one of the modes has incomparable observation units, for example because of differences in signal sampling or batch sizes. A fully probabilistic treatment of the PARAFAC2 is desirable in order to improve robustness to noise and provide a principled approach for determining the number of factors, but challenging because the factor loadings are constrained to be orthogonal. We develop two probabilistic formulations of the PARAFAC2 along with variational procedures for inference: In the first approach, the mean values of the factor loadings are orthogonal leading to closed form variational updates, and in the second, the factor loadings themselves are orthogonal using a matrix Von Mises-Fisher distribution. We contrast our probabilistic formulations to the conventional direct fitting algorithm based on maximum likelihood on synthetic data and real fluorescence spectroscopy and gas chromatography-mass spectrometry data showing that the probabilistic formulations are more robust to noise and model order misspecification. The probabilistic PARAFAC2 thus forms a promising framework for modeling multi-way data accounting for uncertainty.

Index Terms—Tensor decomposition, multi-way modeling, variational inference, orthogonality constraint

1 Introduction

Tensor decompositions are multi-way generalizations of matrix decompositions such as principal component analysis (PCA): A matrix is a second order array with two modes, rows and columns, while a data cube is a third order array with the third mode referred to as slabs. When multi-way data has inherent multi-linear structure, the advantage of tensor decomposition methods is that they capture this intrinsic information and often provide a unique representation without needing further constraints such as sparsity or statistical independence.

Tensor factorization originated within the field of psychometrics [1], [2], and has proved widely useful in other fields such as chemometrics [3] for example to model the relationship between excitation and emission spectra of samples of different mixed compounds obtained by fluorescence spectroscopy [4]. Tensor decomposition is today encountered in practically all fields of research including signal processing, neuroimaging, and information retrieval (see also [5], [6]).

The two most prominent tensor decompositions are i) the Tucker model [7], where the so-called core array accounts for all multi-linear interactions between the components of each mode, and ii) the CandeComp/PARAFAC (CP) model [1], [2], [8], where interactions are restricted to be between components of identical indices across modes, corresponding to a Tucker model with a diagonal core array. Both models can be considered generalizations of PCA to higher order arrays, with the Tucker model being more flexible at the expense of reduced interpretability. The CP model has been widely used primarily due to its ease of interpretation and its uniqueness [6].

In the CP model the components are assumed identical across measurements, varying only in their scaling. In many situations this is too restrictive, for example when signal sampling or batch sizes vary across a mode. In chemometrics, violation of the CP structure can be caused by retention time shifts [9], whereas in neuroimaging violation can be caused by subject and trial variability [6]. To handle variability while preserving the uniqueness of the representation, the PARAFAC2 model was proposed [2]. It admits individual loading matrices for each entry in a mode while preserving uniqueness properties of the decomposition by imposing consistency of the Gram matrix (i.e. the loading matrix left multiplied by its transpose) [10]–[12]. It has since been applied within diverse application domains such as in chemometrics for handling variations in elution profiles due to retention shifts in chromatography [9], for monitoring and fault detection facing unequal batch lengths in chemical processes [13], in neuroimaging to analyze latency changes in frequency resolved evoked EEG potentials [14], to extract common connectivity profiles in multi-subject fMRI data accounting for individual variability [15], for cross-language information retrieval [16], and for music and image tagging [17], [18]. Recently, efforts have been made to scale the PARAFAC2 model to large-scale data [19], [20], enchance the conventional direct fitting algorithm [21], and a nonnegative version have been developed [22].

Traditionally, tensor decompositions have been based on maximum likelihood inference using alternating least squares estimation in which the components of a mode are estimated while keeping the components of other modes fixed. Initial probabilistic approaches defined probability distributions over the component matrices and the core array, but relied on maximum likelihood estimates for determining a solution. However, the Bayesian approach

- *P. J. H. Jørgensen, J. L. Hinrich, M. N. Schmidt, K. H. Madsen and M. Mørup are with Department of Applied Mathematics and Computer Science, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark*
- *K. H. Madsen is also with Danish Research Centre for Magnetic Resonance, Centre for Functional and Diagnostic Imaging and Research, Copenhagen University Hospital Hvidovre, 2650 Hvidovre, Denmark*
- *S. F. V. Nielsen is with with Sennheiser Communications*

This work is supported by Innovation Fund Denmark through the Danish Center for Big Data Analytics and Innovation (DABAI).

presented here makes inference with respect to the posterior distributions of the model parameters, and can thus be used to asses uncertainty in the parameters and noise estimates. Recently, the TUCKER and CP models have been formulated in a probabilistic setting, using either Markov Chain Monte Carlo (MCMC) sampling [23]–[25] or variational inference [26]–[29]. The CP and Tucker models have been extended to model sparsity [25], [30], [31], non-negativity [32] and non-linearity [23], [33] in component loadings. Heteroscedastic noise modeling has been discussed in the context of the CP model [31], [34] and Tucker model [35], the latter also providing a generalization of tensor decomposition to exponential family distributions. A CP model where a subset of the component matrices are orthogonal matrices was recently explored using the von-Mises-Fisher distribution [36], their approach was not fully Bayesian, as they used MAP estimates for the orthogonal matrices, neither did they explore other orthogonal formulation or the PARAFAC2 structure.

Benefits of probabilistic modeling include the ability to account for uncertainty and noise while providing tools for model order selection. Whereas probabilistic modeling can be directly applied to the CP and TUCKER models extending probabilistic PCA [37], a probabilistic treatment of the PARAFAC2 model faces the following two key challenges, i) the ability to impose orthogonality on variational factors (necessary for imposing the PARAFAC2 structure), and ii) handling the coupling of these orthogonal components. In this paper we address these challenges and derive the probabilistic PARAFAC2 model. We investigate two different formulations of the orthogonality constraint and demonstrate how the orthogonality of variational factors as in the least squares estimation for conventional PARAFAC2 can be obtained in closed form using the singular value decomposition. We exploit how the probabilistic framework admits model order quantification by the evaluation of model evidence and automatic relevance determination. We contrast our probabilistic formulation to conventional maximum likelihood estimation on synthetic data as well as flourescence spectroscopy and gas chromatography-mass spectrometry data highlighting the utility of the probabilistic formulation facing noise and model order misspecification.

2 METHODS

The three-way CP model can be formulated as a series of coupled matrix decompositions,

$$\boldsymbol{X}_k = \boldsymbol{A}\boldsymbol{D}_k\boldsymbol{F}^\top + \boldsymbol{E}_k$$

where $\boldsymbol{X}_k \in \mathbb{R}^{I\times J}$ is the k'th slab of the three-way array $\mathcal{X}$ with dimensions $I \times J \times K$. Let M be the number of components in the model, then the matrix $\boldsymbol{A}$ with dimensions $I \times M$ contains loadings for the first mode and $\boldsymbol{F}$ with dimensions $J \times M$ contains loadings for the second mode. The matrices $\boldsymbol{D}_k$, $k = 1, \ldots, K$, are diagonal with dimensions $M \times M$ and contain loadings for the third mode. These are usually written as a single matrix $\boldsymbol{C} \in \mathbb{R}^{K\times M}$ where the k'th row contains the diagonal of $\boldsymbol{D}_k$. $\boldsymbol{E}_k$ denotes the residuals for the k'th slab with dimensions $I \times J$. Notice that the structure of the first and second mode are invariant across the third mode in this model.

The PARAFAC2 model extends the CP structure by letting a mode have individual factors $\boldsymbol{F}_k$ for each slab. The extension allows for a varying number of observations in the chosen mode. This model would be as flexible as PCA on the concatenated data $[\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots \boldsymbol{X}_K]$ if not for the additional constraint that each Gram matrix of $\boldsymbol{F}_k$ be identical, $\boldsymbol{F}_k^\top\boldsymbol{F}_k = \boldsymbol{\Psi}$ which is a necessary constraint in order to obtain unique solutions [38]. The three-way PARAFAC2 model can thus be written as,

$$\boldsymbol{X}_k = \boldsymbol{A}\boldsymbol{D}_k\boldsymbol{F}_k^\top + \boldsymbol{E}_k \quad \text{s.t. } \boldsymbol{F}_k^\top\boldsymbol{F}_k = \boldsymbol{\Psi}.$$

Modeling Ψ explicitly can be difficult, but it is necessary and sufficient [12] to have $\boldsymbol{F}_k = \boldsymbol{P}_k\boldsymbol{F}$, with $\boldsymbol{P}_k$ being a columnwise orthogonal $J \times M$ matrix, and $\boldsymbol{F}$ a $M \times M$ matrix, and the model can thus be written as

$$\boldsymbol{X}_k = \boldsymbol{A}\boldsymbol{D}_k\boldsymbol{F}^\top\boldsymbol{P}_k^\top + \boldsymbol{E}_k \text{ s.t. } \boldsymbol{P}_k^\top\boldsymbol{P}_k = \mathbf{I}. \tag{1}$$

In the following, we describe the conventional direct fitting algorithm [12] for parameter estimation in the PARAFAC2 model, before we introduce the probabilistic model formulation.

2.1 Direct Fitting Algorithm

The parameters in the PARAFAC2 model in (1) can be estimated using the alternating least squares algorithm [12], minimizing the constrained least squares objective function,

$$\min \sum_k \|\boldsymbol{X}_k - \boldsymbol{A}\boldsymbol{D}_k\boldsymbol{F}^\top\boldsymbol{P}_k^\top\|^2 \quad \text{s.t. } \boldsymbol{P}_k^\top\boldsymbol{P}_k = \mathbf{I}.$$

For fixed $\boldsymbol{A}$, $\boldsymbol{D}_k$, and $\boldsymbol{F}$, the $\boldsymbol{P}_k$ that minimizes the k'th term in the objective function is equal to the $\boldsymbol{P}_k$ that maximizes

$$\text{Tr}(\boldsymbol{F}\boldsymbol{D}_k\boldsymbol{A}^\top\boldsymbol{X}_k\boldsymbol{P}_k) \tag{2}$$

and can be computed as [12], [39]

$$\boldsymbol{P}_k = \boldsymbol{V}_k\boldsymbol{U}_k^\top \tag{3}$$

where $\boldsymbol{V}_k$ and $\boldsymbol{U}_k$ comes from the singular value decomposition (SVD) decomposition

$$\boldsymbol{U}_k\boldsymbol{S}_k\boldsymbol{V}_k^\top = \boldsymbol{F}\boldsymbol{D}_k\boldsymbol{A}^\top\boldsymbol{X}_{\boldsymbol{k}}.$$

Upon fitting $\boldsymbol{P}_k$ each slab $\boldsymbol{X}_k$ of the tensor can be projected onto $\boldsymbol{P}_k$ thereby leaving the remaining parameters to be fitted as a CP model minimizing

$$\sum_k \|\boldsymbol{X}_k\boldsymbol{P}_k - \boldsymbol{A}\boldsymbol{D}_k\boldsymbol{F}^\top\|^2. \tag{4}$$

A solution to (4) is well explained by Bro in [40]. A well-known issue with maximum likelihood methods is that it can lead to overfitting due to noise and a lack of uncertainty in the model parameters resulting in robustness issues which we attempt to provide a solution for by advancing the PARAFAC2 model to a fully Bayesian setting.

2.1.1 Model Selection

A general problem for latent variable methods is how to choose the model order. A popular heuristic would be how well the model fits the data given as

$$\text{R2} = 1 - \frac{\sum_k \|\boldsymbol{X}_k - \boldsymbol{A}\boldsymbol{D}_k\boldsymbol{F}^\top\boldsymbol{P}_k^\top\|^2}{\sum_k \|\boldsymbol{X}_k\|^2}. \tag{5}$$

However, this measure will simply increase until the model incorporates enough parameters to completely fit the data, thus eventually leading to overfitting. The model selection criterion would only be based on the expected noise level.

Another popular heuristic is the core consistency diagnostic (CCD) originally developed for the CP model [41], but shown useful for the PARAFAC2 model as well [42]. It is based on the observation that the PARAFAC model can be seen as a constrained Tucker model, where the core array is enforced to be a superdiagonal array of ones. The principle behind CCD is to measure how much the PARAFAC model violates this assumption of a superdiagonal core array of ones by re-estimating the core array of the PARAFAC model to fit the Tucker model, denoted $\mathcal{G}$, while keeping the loadings fixed and then calculating the CCD according to,

$$\text{CCD} = 100\left(1 - \frac{||\mathcal{G} - \mathcal{I}||^2_{\mathcal{F}}}{||\mathcal{I}||^2_{\mathcal{F}}}\right)$$

in which $\mathcal{I}$ is the superdiagonal core array and $||\cdot||_{\mathcal{F}}$ denotes the Frobenius norm. The PARAFAC2 model can be written as a PARAFAC model for each slab as in (4), and thus the core array can be estimated in the same way as for the standard PARAFAC model. This approach have been evaluated on synthetic as well real data sets by [42] where the conclusion is even though the CCD is found to be an useful parameter for determining model order, it is not recommended to be used without considering other diagnostic measures like the residuals and the loadings.

2.2 Variational Bayesian Inference

In Bayesian modeling, the posterior distribution of the parameters $\boldsymbol{\theta}$ is computed by conditioning on the observed data $\boldsymbol{X}$ using Bayes' rule, $p(\boldsymbol{\theta}|\boldsymbol{X}) = p(\boldsymbol{X}|\boldsymbol{\theta})p(\boldsymbol{\theta})/p(\boldsymbol{X})$. It is given by the product of the likelihood $p(\boldsymbol{X}|\boldsymbol{\theta})$ and the prior probability of the parameters $p(\boldsymbol{\theta})$, divided by the probability of the observed data $p(\boldsymbol{X})$ under the model, also known as the marginal likelihood. Evaluating the marginal likelihood is in general intractable, and instead a variational approximation can be found by fitting a distribution $q(\boldsymbol{\theta})$ to the posterior [43] minimizing the Kullback-Leibler (KL) divergence, given by

$$q^\star(\boldsymbol{\theta}) = \arg\min \text{KL}\big[q(\boldsymbol{\theta})\big\|p(\boldsymbol{\theta}|\boldsymbol{X})\big].$$

Minimizing the KL divergence is solved by maximizing a related quantity, the evidence lower bound (ELBO).

$$\text{ELBO}(q(\boldsymbol{\theta})) = \mathbb{E}[\log p(\boldsymbol{\theta}, \boldsymbol{X})] - \mathbb{E}[\log q(\boldsymbol{\theta})].$$

A common choice is a variational distribution that factorizes over the parameters, known as a mean-field approximation, $q(\boldsymbol{\theta}) = \prod_j q_j(\boldsymbol{\theta}_j)$. The optimal variational distribution can then be found by iterative updates of the form

$$q_j(\theta_j) \propto \exp\left(\mathbb{E}_{-j}[\log p(\theta_j, \boldsymbol{\theta}_{-j}, \boldsymbol{X})]\right) \tag{6}$$

where $\mathbb{E}_{-j}[\cdot]$ denotes the expectation over the variational distribution except q_j. For a comprehensive overview of variational inference see for example [44], [45].

2.3 Probabilistic PARAFAC2

We propose a probabilistic PARAFAC2 using the formulation in (1). The constraint $\boldsymbol{P}_k^\top\boldsymbol{P}_k = \boldsymbol{I}_M$ has the probabilistic interpretation i) $\mathbb{E}[\boldsymbol{P}_k^\top\boldsymbol{P}_k] = \boldsymbol{I}_M$, but one could also consider to model ii) $\mathbb{E}[\boldsymbol{P}_k]^\top\mathbb{E}[\boldsymbol{P}_k] = \boldsymbol{I}_M$. The main motivation for the latter approach being the interpretation of the orthogonal factor is identical to that of the maximum likelihood estimation, however the resulting components are no longer themselves restricted to the set of orthogonal matrices (the Stiefel manifold). As such, the model becomes more flexible as only the mean parameters of the variational approximation are constrained to be orthogonal and not the expectation of their inner product as required for every realization of the underlying distribution to conform to the PARAFAC2 model. We include the latter model formulation as it provides simple closed form updates similar to the conventional direct fitting PARAFAC2 algorithm as shown below. The updates are derived by constraining the mean of a matrix normal ($\mathcal{MN}$) distribution within the variational approximation to the Stiefel manifold, whereas the former formulation based on [46] uses a matrix von Mises-Fisher (vMF) distribution which has support on the Stiefel manifold only. We thus have the generative models i) and ii),

$$\begin{aligned}
&\boldsymbol{a}_{i\cdot} \sim \mathcal{N}(\boldsymbol{0}, \mathbf{I}_M)\\
&\boldsymbol{f}_{m\cdot} \sim \mathcal{N}(\boldsymbol{0}, \mathbf{I}_M)\\
&\boldsymbol{c}_{k\cdot} \sim \mathcal{N}(\boldsymbol{0}, \text{diag}(\boldsymbol{\alpha}^{-1}))\\
\text{i)}\quad &\boldsymbol{P}_k \sim \text{vMF}\,(\boldsymbol{0})\\
\text{ii)}\quad &\boldsymbol{P}_k \sim \mathcal{MN}\,(\boldsymbol{0}, \mathbf{I}_J, \mathbf{I}_M)\\
&\tau_k \sim \text{Gamma}(a_{\tau_k}, b_{\tau_k})\\
&\boldsymbol{X}_k \sim \mathcal{N}(\boldsymbol{A}\boldsymbol{D}_k\boldsymbol{F}^\top\boldsymbol{P}_k^\top, \tau_k^{-1}\mathbf{I}_J)
\end{aligned}$$

where $\boldsymbol{a}_{i\cdot}$ denotes the ith row of the matrix $\boldsymbol{A}$ etc. In the above formulation, $\boldsymbol{\alpha}$ defines the length scale of each component and can thus be used for automatic relevance determination by turning off excess components by concentrating their distributions at zero when α_m is large [44]. We further allow the noise to vary across slabs thereby accounting for potential different levels of the noise (i.e., assuming heteroscedastic noise) across slabs.

2.4 Variational Update Rules

The inference is based on the following factorized distribution,

$$\text{q}(\boldsymbol{\theta}) = q(\boldsymbol{A})q(\boldsymbol{C})\prod_m q(\boldsymbol{f}_{m\cdot})\prod_k q(\boldsymbol{P}_k)q(\tau_k)$$

leading to the following ELBO,

$$\begin{aligned}\text{ELBO}(\mathrm{q}(\boldsymbol{\theta})) &= \mathbb{E}[\log p(\mathcal{X}, \boldsymbol{\theta})] - \mathbb{E}[\log \mathrm{q}(\boldsymbol{\theta})] \\ &= \mathbb{E}[\log \mathrm{p}(\mathcal{X} \mid \boldsymbol{A}, \boldsymbol{C}, \boldsymbol{F}, \mathcal{P}, \boldsymbol{\tau})] + \mathbb{E}[\log \mathrm{p}(\boldsymbol{A})] \\ &\quad + \mathbb{E}[\log \mathrm{p}(\boldsymbol{C} \mid \boldsymbol{\alpha})] + \mathbb{E}[\log \mathrm{p}(\boldsymbol{F})] \\ &\quad + \mathbb{E}[\log \mathrm{p}(\mathcal{P})] + \mathbb{E}[\log \mathrm{p}(\boldsymbol{\tau})] \\ &\quad + \mathrm{h}(\mathrm{q}(\boldsymbol{A})) + \mathrm{h}(\mathrm{q}(\boldsymbol{C})) + \mathrm{h}(\mathrm{q}(\boldsymbol{F})) \\ &\quad + \mathrm{h}(\mathrm{q}(\mathcal{P})) + \mathrm{h}(\mathrm{q}(\boldsymbol{\tau})). \end{aligned} \tag{7}$$

Expanding the variational factors as given by (6), the resulting variational distributions and update rules are given in TABLE 1. The update for the factor matrix $\boldsymbol{F}$ is non-trivial, and to obtain a closed-form solution we employ a component-wise updating scheme inspired by the non-negative matrix factorization literature [47]–[49]. For each latent parameter we use (6) and moment matching to determine the optimal variational distributions.

2.4.1 Von Mises-Fisher Loading

In the von Mises-Fisher model for the loading $\boldsymbol{P}_k$, the variational distribution is given by

$$\text{vMF}(\boldsymbol{P}_k|\boldsymbol{B}_{\boldsymbol{P}_k}) = \kappa(J, \boldsymbol{B}_{\boldsymbol{P}_k}^\top \boldsymbol{B}_{\boldsymbol{P}_k})^{-1} \exp\big(\text{tr}[\boldsymbol{B}_{\boldsymbol{P}_k}^\top \boldsymbol{P}_k]\big)$$

which is defined on the Stiefel manifold, $\boldsymbol{P}_k^\top \boldsymbol{P}_k = \boldsymbol{I}$. The normalization constant is given by $\kappa = {}_0F_1\left(\frac{1}{2}J, \frac{1}{4}\boldsymbol{B}_{\boldsymbol{P}_k}^\top \boldsymbol{B}_{\boldsymbol{P}_k}\right)\text{v}_{J,M}$ where $\text{v}_{J,M}$ is the volume of the J-dimensional Stiefel manifold described by M components [50].

The hypergeometric function with matrix argument can be calculated more efficiently using the SVD of $\boldsymbol{B}_{\boldsymbol{P}_k} = \boldsymbol{U}_k \boldsymbol{S}_k \boldsymbol{V}_k^\top$, since ${}_0F_1\left(\frac{1}{2}J, \frac{1}{4}\boldsymbol{B}_{\boldsymbol{P}_k}^\top \boldsymbol{B}_{\boldsymbol{P}_k}\right) = {}_0F_1\left(\frac{1}{2}J, \frac{1}{4}\boldsymbol{S}_k^2\right)$ [50].

Computing expectations over the vMF matrix distribution requires evaluating the hypergeometric function and can be done as described by [46].[†1] Note, that it follows from the vMF matrix distribution that $\mathbb{E}[\boldsymbol{P}_k^\top \boldsymbol{P}_k] = \mathbf{I}$, but in general $\mathbb{E}[\boldsymbol{P}_k^\top]\mathbb{E}[\boldsymbol{P}_k] \neq \mathbf{I}$. However, if an orthogonal summary representation is desired one can inspect the mode of the vMF given by $\mathbf{U}_k \mathbf{V}_k^\top$.

2.4.2 Constrained Matrix Normal Loading

In the constrained matrix normal ($c\mathcal{MN}$) model for the variational factor of the loadings $\boldsymbol{P}_k$, instead of using the free form variational approach, we maximize (7) as a function of the mean parameter $\boldsymbol{M}_{\boldsymbol{P}_k}$ subject to the orthogonality constraint $\boldsymbol{M}_{\boldsymbol{P}_k} \boldsymbol{M}_{\boldsymbol{P}_k}^\top = \mathbf{I}_M$.

The constraint consequently causes (7) to be constant except for the linear term of the expected log of the probability density function of the data. The reason for this is that all other terms do not depend on $\boldsymbol{M}_{\boldsymbol{P}_k}$ or only on the matrix product $\boldsymbol{M}_{\boldsymbol{P}_k} \boldsymbol{M}_{\boldsymbol{P}_k}^\top$, which is equivalent to the identity matrix, resulting in the optimization problem

$$\arg\max_{\boldsymbol{M}_{\boldsymbol{P}_k}} \text{ELBO}(\boldsymbol{M}_{\boldsymbol{P}_k}) \text{ s. t. } \boldsymbol{M}_{\boldsymbol{P}_k} {\boldsymbol{M}_{\boldsymbol{P}_k}}^\top = \mathbf{I}$$

where

$$\text{ELBO}(\boldsymbol{M}_{\boldsymbol{P}_k}) = \sum_k \mathbb{E}[\tau_k] \text{Tr}(\mathbb{E}[\boldsymbol{F}]\mathbb{E}[\boldsymbol{D}_k]\mathbb{E}[\boldsymbol{A}^\top]\boldsymbol{X}_k \boldsymbol{M}_{\boldsymbol{P}_k}) + c.$$

This is equal to (2) except for a scalar leading to the same solution as for the maximum likelihood estimation method as given in (3). More details on identifying the expression above are given in the supplemental material. The variance parameter $\boldsymbol{\Sigma}_{\boldsymbol{P}_k}$ in the variational distribution follows from moment matching using (6).

2.4.3 The F Matrix

The updates for $\boldsymbol{f}_{m\cdot}$ are non-trivial due to an intercomponent dependency. The quadratic term in (6) for $\boldsymbol{F}$ is

$$\begin{aligned}&\mathbb{E}_{-\boldsymbol{F}}[\boldsymbol{a}_{i\cdot} \boldsymbol{D}_k \boldsymbol{F}^\top {\boldsymbol{P}_k}^\top \boldsymbol{P}_k \boldsymbol{F} \boldsymbol{D}_k {\boldsymbol{a}_{i\cdot}}^\top] \\ &= \mathbb{E}_{-\boldsymbol{F}}[\text{Tr}(\boldsymbol{F} \boldsymbol{D}_k {\boldsymbol{a}_{i\cdot}}^\top \boldsymbol{a}_{i\cdot} \boldsymbol{D}_k \boldsymbol{F}^\top {\boldsymbol{P}_k}^\top \boldsymbol{P}_k)] \\ &= \text{Tr}(\boldsymbol{F} \mathbb{E}_{-\boldsymbol{F}}[\boldsymbol{D}_k {\boldsymbol{a}_{i\cdot}}^\top \boldsymbol{a}_{i\cdot} \boldsymbol{D}_k] \boldsymbol{F}^\top \mathbb{E}_{-\boldsymbol{F}}[{\boldsymbol{P}_k}^\top \boldsymbol{P}_k]) \\ &= \sum_{mm\prime} (\boldsymbol{F} \mathbb{E}[\boldsymbol{D}_k {\boldsymbol{a}_{i\cdot}}^\top \boldsymbol{a}_{i\cdot} \boldsymbol{D}_k] \boldsymbol{F}^\top)_{mm\prime} (\mathbb{E}[{\boldsymbol{P}_k}^\top \boldsymbol{P}_k])_{mm\prime} \\ &= \sum_{mm\prime} \boldsymbol{f}_{m\cdot} \mathbb{E}[\boldsymbol{D}_k {\boldsymbol{a}_{i\cdot}}^\top \boldsymbol{a}_{i\cdot} \boldsymbol{D}_k] \mathbf{f}_{m\prime\cdot}^\text{T} \mathbb{E}[\mathbf{p}_{\cdot mk}^\text{T} \mathbf{p}_{\cdot m\prime k}] \\ &= \sum_m \boldsymbol{f}_{m\cdot} \mathbb{E}[\boldsymbol{D}_k {\boldsymbol{a}_{i\cdot}}^\top \boldsymbol{a}_{i\cdot} \boldsymbol{D}_k] \mathbb{E}[\mathbf{p}_{\cdot mk}^\text{T} \mathbf{p}_{\cdot mk}] \mathbf{f}_{m\cdot}^\text{T} \\ &\quad + 2 \sum_m \sum_{m\prime \setminus m} \boldsymbol{f}_{m\cdot} \mathbb{E}[\boldsymbol{D}_k {\boldsymbol{a}_{i\cdot}}^\top \boldsymbol{a}_{i\cdot} \boldsymbol{D}_k] \mathbb{E}[\mathbf{p}_{\cdot mk}^\text{T} \mathbf{p}_{\cdot m\prime k}] \mathbf{f}_{m\prime\cdot}^\text{T}.\end{aligned}$$

where we see that the quadratic term separates into a quadratic and linear part revealing the linear intercomponent dependency.

2.4.4 Non-trivial expectations

An overview of all the factors and their updates are given in TABLE 1. Below we detail some non-trivial expectations and the necessary steps to compute them. The first group of expectations deals with having the diagonal matrix $\boldsymbol{D}_k$ left and right multiplied with an inner term. The first case is the following expectation

$$\mathbb{E}[\boldsymbol{D}_k {\boldsymbol{a}_{i\cdot}}^\top \boldsymbol{a}_{i\cdot} \boldsymbol{D}_k]$$

which is equivalent to the Hadamard product of the outer-product of the diagonal of the surrounding matrix with itself and the inner matrix, so we can separate the expectation into two parts

$$\mathbb{E}[\boldsymbol{D}_k {\boldsymbol{a}_{i\cdot}}^\top \boldsymbol{a}_{i\cdot} \boldsymbol{D}_k] = \mathbb{E}[{\boldsymbol{c}_{k\cdot}}^\top \boldsymbol{c}_{k\cdot}] \circ \mathbb{E}[{\boldsymbol{a}_{i\cdot}}^\top \boldsymbol{a}_{i\cdot}]$$

where $\boldsymbol{c}_{k\cdot}$ is the vector containing the diagonal elements of $\boldsymbol{D}_k$. The same rule applies for the following expectation

$$\mathbb{E}[\boldsymbol{D}_k \boldsymbol{F}^\top {\boldsymbol{P}_k}^\top \boldsymbol{P}_k \boldsymbol{F} \boldsymbol{D}_k] = \mathbb{E}[{\boldsymbol{c}_{k\cdot}}^\top \boldsymbol{c}_{k\cdot}] \circ \mathbb{E}[\boldsymbol{F}^\top {\boldsymbol{P}_k}^\top \boldsymbol{P}_k \boldsymbol{F}]$$

where the second expectation becomes trivial when using the vMF prior (ii) as the matrix product ${\boldsymbol{P}_k}^\top \boldsymbol{P}_k$ is the identity matrix. However, when using the matrix normal distribution (i) we get

$$\mathbb{E}[{\boldsymbol{P}_k}^\top \boldsymbol{P}_k] = \text{Tr}(\boldsymbol{\Sigma}_{\boldsymbol{P}_k}) + \mathbf{I}_M$$

1. † Source code for approximating the hypergeometric function is available online http://staff.utia.cz/smidl/files/mat/OVPCA.zip (24 Feb 2017). This code was used with default settings and without modifications in the experiments.

TABLE 1
Overview of All the Variational Factors and Their Updates.

Variational factor	Update
$\mathrm{q}(\boldsymbol{A}) \sim \prod_i \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{a}_{i\cdot}}, \boldsymbol{\Sigma}_{\boldsymbol{a}_{i\cdot}})$	$\boldsymbol{\Sigma}_{\boldsymbol{a}_{i\cdot}} = \left(\mathbf{I}_M + \sum_k \mathbb{E}[\tau_k]\mathbb{E}[\boldsymbol{D}_k \boldsymbol{F}^\top \boldsymbol{P}_k^\top \boldsymbol{P}_k \boldsymbol{F} \boldsymbol{D}_k]\right)^{-1}$ $\boldsymbol{\mu}_{\boldsymbol{a}_{i\cdot}} = \boldsymbol{\Sigma}_{\boldsymbol{a}_{i\cdot}} \sum_k \mathbb{E}[\tau_k]\mathbb{E}[\boldsymbol{D}_k \boldsymbol{F}^\top \boldsymbol{P}_k^\top \boldsymbol{x}_{i\cdot k}^\top]$
$\mathrm{q}(\boldsymbol{C}) \sim \prod \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{c}_{k\cdot}}, \boldsymbol{\Sigma}_{\boldsymbol{c}_{k\cdot}})$	$\boldsymbol{\Sigma}_{\boldsymbol{c}_{k\cdot}} = \left(\mathrm{diag}(\boldsymbol{\alpha}) + \mathbb{E}[\tau_k]\mathbb{E}[\boldsymbol{F}^\top \boldsymbol{P}_k^\top \boldsymbol{P}_k \boldsymbol{F}] \circ \mathbb{E}[\boldsymbol{A}^\top \boldsymbol{A}]\right)^{-1}$ $\boldsymbol{\mu}_{\boldsymbol{c}_{k\cdot}} = \boldsymbol{\Sigma}_{\boldsymbol{c}_{k\cdot}} \mathbb{E}[\tau_k] \mathrm{diag}(\mathbb{E}[\boldsymbol{F}^\top]\mathbb{E}[\boldsymbol{P}_k^\top]\boldsymbol{X}_k^\top \mathbb{E}[\boldsymbol{A}])$
$\mathrm{q}(\boldsymbol{F}) \sim \prod_m \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{f}_{m\cdot}}, \boldsymbol{\Sigma}_{\boldsymbol{f}_{m\cdot}})$	$\boldsymbol{\Sigma}_{\boldsymbol{f}_{\boldsymbol{m}\cdot}} = \left(\sum_k \mathbb{E}[\tau_k]\mathbb{E}[\boldsymbol{D}_k \boldsymbol{A}^\top \boldsymbol{A} \boldsymbol{D}_k]\mathbb{E}[\boldsymbol{p}_{\cdot mk}^\top \boldsymbol{p}_{\cdot mk}] + \mathbf{I}_M\right)^{-1}$ $\boldsymbol{\mu}_{\boldsymbol{f}_{\boldsymbol{m}\cdot}} = \boldsymbol{\Sigma}_{\boldsymbol{f}_{\boldsymbol{m}\cdot}} \left(\sum_k \mathbb{E}[\tau_k]\{\mathbb{E}[(\boldsymbol{P}_k^\top)_m]\boldsymbol{X}_k^\top \mathbb{E}[\boldsymbol{A}]\mathbb{E}[\boldsymbol{D}_k] - \mathbb{E}[\boldsymbol{D}_k \boldsymbol{A}^\top \boldsymbol{A} \boldsymbol{D}_k] \sum_{m' \setminus m} \mathbb{E}[\boldsymbol{p}_{\cdot mk}^\top \boldsymbol{p}_{\cdot m'k}] \boldsymbol{f}_{m'\cdot}^\top\}\right)$
$\mathrm{q}(\mathcal{P}) \sim \prod \mathrm{vMF}(\boldsymbol{B}_{\boldsymbol{P}_k})$	$\boldsymbol{B}_{\boldsymbol{P}_k} = \mathbb{E}[\tau_k]\mathbb{E}[\boldsymbol{F}]\mathbb{E}[\boldsymbol{D}_k]\mathbb{E}[\boldsymbol{A}^\top]\boldsymbol{X}_k$ $\mathbb{E}[\boldsymbol{P}_k] = \boldsymbol{V}_k \boldsymbol{\Psi} \boldsymbol{U}_k^\top$, where $\boldsymbol{B}_{\boldsymbol{P}_k} = \boldsymbol{U}_k \boldsymbol{S}_k \boldsymbol{V}_k^\top$ (SVD) ($\boldsymbol{\Psi}$ given by [46], Appendix A.2)
$\mathrm{q}(\mathcal{P}) \sim \prod_k c\mathcal{MN}\left(\boldsymbol{M}_{\boldsymbol{P}_k}, \mathbf{I}_J, \boldsymbol{\Sigma}_{\boldsymbol{P}_k}\right)$	$\boldsymbol{\Sigma}_{\boldsymbol{P}_k} = (\mathbb{E}[\boldsymbol{F}\boldsymbol{D}_k \boldsymbol{A}^\top \boldsymbol{A} \boldsymbol{D}_k \boldsymbol{F}^\top] + \mathbf{I})^{-1}$ $\boldsymbol{M}_{\boldsymbol{P}_k} = \boldsymbol{V}_k \boldsymbol{U}_k^\top$, where $\mathbb{E}[\tau_k]\mathbb{E}[\boldsymbol{F}]\mathbb{E}[\boldsymbol{D}_k]\mathbb{E}[\boldsymbol{A}^\top]\boldsymbol{X}_k = \boldsymbol{U}_k \boldsymbol{S}_k \boldsymbol{V}_k^\top$ (SVD)
$\mathrm{q}(\boldsymbol{\tau}) \sim \prod \mathrm{Gamma}(a_{\tau_k}, b_{\tau_k})$	$a_{\tau_k} = a_{\boldsymbol{\tau}} + \frac{I \cdot J}{2}$ $b_{\tau_k} = \left(b_{\boldsymbol{\tau}}^{-1} + \frac{1}{2}\mathrm{Tr}(\boldsymbol{X}_k \boldsymbol{X}_k^\top) + \frac{1}{2}\mathbb{E}[\mathrm{Tr}(\boldsymbol{A}\boldsymbol{D}_k \boldsymbol{F}^\top \boldsymbol{P}_k^\top \boldsymbol{P}_k \boldsymbol{F} \boldsymbol{D}_k \boldsymbol{A}^\top)] - \mathbb{E}[\mathrm{Tr}(\boldsymbol{A}\boldsymbol{D}_k \boldsymbol{F}^\top \boldsymbol{P}_k^\top \boldsymbol{X}_k^\top)]\right)^{-1}$
$\arg\max_{\alpha_m} \mathrm{ELBO}(\alpha_m)$	$\alpha_m = K\left(\sum_k \mathbb{E}[c_{km}^2]\right)^{-1}$

which lead to the element with index ij of the expectation to be equal to

$$\begin{aligned}\mathbb{E}[\boldsymbol{F}^\top \boldsymbol{P}_k{}^\top \boldsymbol{P}_k \boldsymbol{F}]_{ij} &= \mathbb{E}[\sum_m (\boldsymbol{F}^\top)_{im} (\boldsymbol{P}_k{}^\top \boldsymbol{P}_k \boldsymbol{F})_{mj}] \\ &= \mathbb{E}[\sum_m \boldsymbol{F}_{mi}^\top \sum_{m'} (\boldsymbol{P}_k{}^\top \boldsymbol{P}_k)_{mm'} \boldsymbol{F}_{m'j}] \\ &= \sum_m \sum_{m'} \mathbb{E}[\boldsymbol{F}_{mi}^\top \boldsymbol{F}_{m'j}] \mathbb{E}[(\boldsymbol{P}_k{}^\top \boldsymbol{P}_k)_{mm'}]\end{aligned}$$

where since the m'th and m' components are independent, we have

$$\mathbb{E}[\boldsymbol{F}_{mi}^\top \boldsymbol{F}_{m'j}] = \begin{cases} \mathbb{E}[\boldsymbol{F}_{mi}^\top]\mathbb{E}[\boldsymbol{F}_{m'j}] + (\boldsymbol{\Sigma}_{\boldsymbol{f}_{\boldsymbol{m}\cdot}})_{ij} & \text{for } m = m'. \\ \mathbb{E}[\boldsymbol{F}_{mi}^\top]\mathbb{E}[\boldsymbol{F}_{m'j}] & \text{for } m \neq m'. \end{cases}$$

These are the most involved expectations when computing the update rules, and the remaining are either simpler or depend upon the expectations derived here.

2.5 Noise Modeling

The probabilistic formulation of PARAFAC2 requires the specification and estimation of the noise precision $\boldsymbol{\tau}$. We presently consider two specifications, i.e. homoscedastic noise in which the noise of each slab $\boldsymbol{X}_k$ is identical, i.e. $\tau_1 = \ldots = \tau_K$ as assumed in the direct fitting algorithm, and heteroscedatic noise where the model includes a separate precision for each of the K slabs.

2.6 Model Selection

A benefit of a fully probabilistic formulation of the PARAFAC2 model is that it provides model order quantification using tools from Bayesian inference. We presently exploit automatic relevance determination by learning the length scale $\boldsymbol{\alpha}$, see also [44]. In practice we use the MAP estimates for the automatic relevance determination because we are more interested in the pruning ability than the uncertainty estimates on $\boldsymbol{\alpha}$. If desired, a variational estimate is easily found by letting α_m follow a Gamma distribution, c.f. [37]. Finally, the estimated ELBO on the data can also be used to compare different model orders.

3 Results and Discussion

We evaluate the proposed models on both synthetic data and 3 real data sets; an amino acid fluorescence (AAF) data set and two gas chromatography mass spectrometry (GC-MS) data sets. We initialize the model parameters for the probabilistic models as the PARAFAC2 solution computed using the direct fitting algorithm[2] and repeat the initialization 5 times for the synthetic data and 50 times for the real data to minimize the risk of getting stuck in a local extrema. The final model parameters are chosen as the parameters with the lowest R2 for the direct fitting models and highest ELBO for the probabilistic models among the fitted models. Each model estimation is limited to 10^4 iterations for the synthetic data and 5×10^4 iterations for the real data. If the improvement in R2 for the direct fitting models and the ELBO for the probabilistic models after an iteration goes below 10^{-9} we invoke an early stop. Empirically we experienced better learning of the probabilistic models by keeping the precision parameter of the added noise fixed for some number of iterations while estimating the length scale $\boldsymbol{\alpha}$. We choose this delay to last for the first 50 iterations. The hyperparameters of the precision was set to $(a_{\tau_k}, b_{\tau_k}) = (1, 10^{32})$ in order to be uninformative for the variational distribution as their influence on the updated parameters are very small on the considered data sets.

3.1 Synthetic Data

To investigate the performance of the proposed model, we generate synthetic data sets in a similar manner as in [12]. We generated the data tensor $\mathcal{X}$ by sampling $\boldsymbol{A}$ from a standard multivariate normal distribution. $\boldsymbol{F}$ was taken from a Cholesky factorization of a matrix with 1's in its diagonal and 0.4 in all the off-diagonal elements. This essentially keeps the M components from being too similar. Each element of $\boldsymbol{C}$ was sampled from a uniform distribution on the interval 0 to 30. $\boldsymbol{P}_k$ was constructed by the standard orthonormalization function in MATLAB of a

2. As implemented by Bro [12] at http://www.models.life.ku.dk/go?filename=parafac2.m (13 Oct 2017)

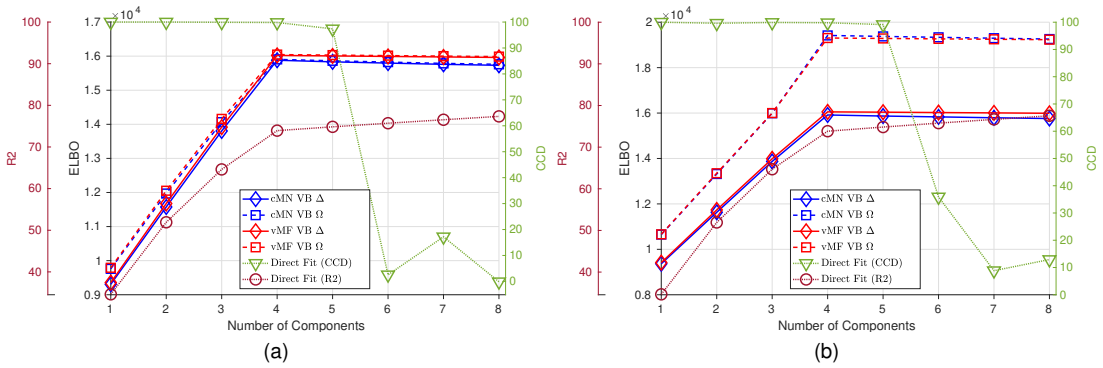


Fig. 1. Mean of model selection criteria R2, CCD, and ELBO reported on the conventional PARAFAC2 and probabilistic PARAFAC2 models with 1 to 8 components on 10 synthetic data sets with added homoscedastic ((a)) and heteroscedastic ((b)) noise both with a SNR equal to 4. In the legend Δ indicates a homoscedastic noise model and Ω indicates a heteroscedastic noise model.

set of vectors sampled from a multivariate normal distribution. The synthetic data sets were generated with either homoscedastic or heteroscedastic additive noise, at different signal-to-noise ratios (SNR) in the interval $[-20, 10]$ with increments of 2. Each configuration was generated 10 times resulting in 320 data sets. Each data set was given the dimensions $50 \times 50 \times 10$ with 4 components.

The probabilistic PARAFAC2 models were fitted to the data sets with the results on the synthetic data shown in Fig. 1 and 2. To investigate the effect of the principled model selection approach based on the ELBO we compare it to the existing model order selection heuristics by plotting the different selection criteria as a function of the number of components used in the model in Fig. 1a and 1b. The figures show the mean result of the models fitted on the 10 synthetic data sets with 4 components and an SNR of 4. Overall the ELBO suggests the same number of components as the other two criteria, R2 and CCD. When the data has heteroscedastic noise the two probabilistic models that incorporate this have a substantially higher ELBO compared to the homoscedastic models.

The results for varying SNR using the true number of components in each model are shown in Fig. 2a for data with homoscedastic noise and in Fig. 2b for data with heteroscedastic noise. We report the R2 on the noiseless data, i.e. using the formula from (5) with the modification that the noise $\boldsymbol{E}_k$ has been subtracted from $\boldsymbol{X}_k$ for each slab. Thereby, we measure the different models' ability to capture the true underlying structure in the data.

On the homoscedastic data we see a small advantage of using the two vMF models compared to the direct fitting algorithm when we decrease the SNR of the data. The $c\mathcal{MN}$ models performs slightly worse compared to the direct fitting algorithm. When we move to the the heteroscedastic data, we see a stronger separation of the four different probabilistic methods. Naturally, the models with heteroscedastic noise outperform the ones with homoscedastic noise. It is also evident that the penalty of modeling the noise as heteroscedastic even though it is homoscedastic is small.

Furthermore, if the number of components is misspecified, cf. Fig. 2c and Fig. 2d, we see a larger difference between the performance of the probabilistic models accounting for the heteroscedastic noise and the direct fitting algorithm. Again, the vMF models perform better compared to the $c\mathcal{MN}$ parameterization and we see a larger positive effect of using the probabilistic models over the direct fitting algorithm. This is mainly explained by the reduced tendency to overfit when accounting for the uncertainty and the automatic relevance determination pruning irrelevant components as the Bayesian modeling promotes the simplest possible representation.

3.2 Real Data

As our synthetic results suggest both formulations of the orthogonality constraint appear to be reasonable, we further investigate their performance on the 3 real world data sets. The first is amino acid fluorescence (AAF) data[3] described in [47], [51] in which the core-consistency diagnostic based on the PARAFAC2 model previously has successfully identified the 3 underlying constituents; tyrosine, tryptophan and phenylalanine [42]. The dataset contains 5 samples with 201 emission and 61 excitation intervals.

Furthermore, the models have been evaluated on two gas-chromatography mass-spectrometry data sets. The first of these originating from wine (GC-MS-WINE)[4] described in detail in [52]. PARAFAC2 has previously been used on GC-MS data obtained from measuring wine [42], [53]. The second data set based on tobacco (GC-MS-TOBAC) is produced by [19] and kindly made available by the authors upon request. The GC-MS-WINE data contains 44 samples of wine and here we specifically consider the unaligned data at the elution times 4.5903-4.7527 min over mass range m/z $5 - 204$. The GC-MS-TOBAC data contains 65 samples of tobacco measured at elution times 4.95-5.03 min over the mass range m/z $50.0 - 350.0$.

In Fig. 3-6 we consider the estimated components using the direct fitting algorithm and the proposed probabilistic PARAFAC2 respectively with homo- and heteroscedastic

3. Available at http://www.models.life.ku.dk/Amino_Acid_fluo
4. Available at http://www.models.life.ku.dk/Wine_GCMS_FTIR

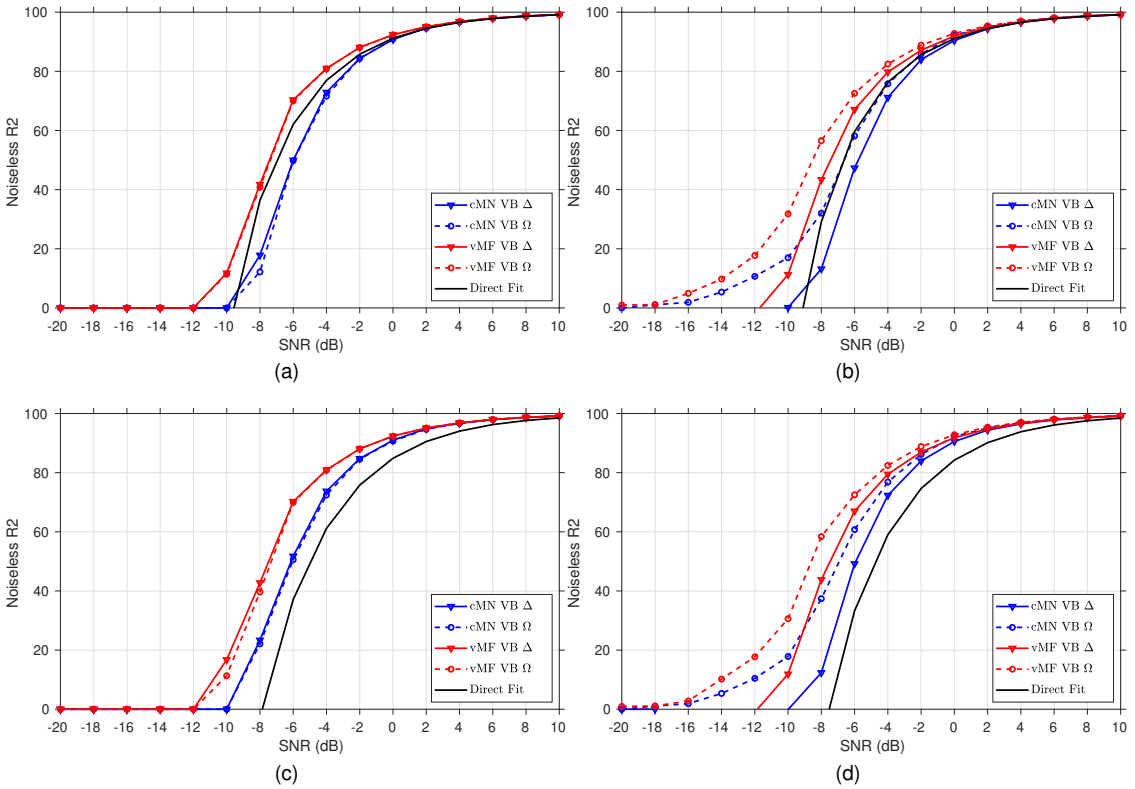


Fig. 2. Recovery of the underlying signal in synthetic data with varying levels of homoscedastic ((a),(c)) and heteroscedastic ((b),(d)) added noise as measured by noiseless R2. Both for the conventional PARAFAC2 and probabilistic PARAFAC2 models fitted with the true number of components ((a),(b), with $M = 4$), and with an overspecified number of components ((c),(d), with $M = 6$) . In the legend Δ indicates a homoscedastic noise model and Ω indicates a heteroscedastic noise model.

noise. In Fig. 3 we report the ELBO using the probabilistic models as well as the R2 and CCD using the direct fitting algorithm and in Fig. 4-6 the extracted profiles for each data set.

For the amino acid fluorescence data we observe that both the R2 and CCD strongly suggest that a three component model sufficiently describes the data whereas the ELBO finds no substantial improvements beyond three components (Fig. 3a) as well. Investigating the extracted excitation loadings in Fig. 4 we observe that both the probabilistic and direct fitting PARAFAC2 models extract similar components when too few or the correct number of components are specified, i.e. $M \leq 3$. However, facing misspecification by having chosen too many components the direct fitting algorithm extracts noisy profiles that incorrectly reflect the underlying three constituents whereas the probabilistic PARAFAC2 models more robustly recover the three constituents when overspecifying the number of components in particular when assuming homoscedastic noise.

For the GC-MS-WINE data R2 and CCD point to a 4 or 5 component model whereas the ELBO points to adding additional components (cf. Fig. 3b). Inspecting the extracted components in Fig. 5, we again observe close agreement between the extracted components using the probabilistic and direct fitting PARAFAC2 approaches when specifying a low number of components ($M \leq 5$). Furthermore, the estimated elution profiles facing model order misspecification appears less influenced by noise than the elution profiles extracted using the direct fitting algorithm emphasizing the improved robustness by the Bayesian approach.

For the GC-MS-TOBAC data Fig. 3c indicates a 3 component model following R2 and CCD, whereas it is harder to decide based on the ELBO. The change in the ELBO from 2 to 3 components for the homoscedastic noise models suggests that local maxima have been identified. Inspecting the extracted components in Fig. 6 it is also evident that local maxima have been reached for most of the probabilistic PARAFAC2 models with $M < 4$. For $M > 3$ most of the probabilistic models successfully recover the 3 components without using the extra components where the direct fitting algorithm splits the three components into multiple components.

On the three considered data sets the ELBO itself does not strongly indicate an optimal number of components, however, most of the probabilistic models still manage

recover the underlying structure given by the ground-truth or expert conclusion in spite of being overspecified.

We attribute this to the regularization invoked by accounting for uncertainty and the automatic relevance determination promoting the pruning of excess components. The relative importance of each component can be observed from the Hinton diagrams in Fig. 4-6. Each square in the Hinton diagrams indicates the relative contribution of each component to the full data reconstruction computed as the squared Frobenius norm of the componentwise data reconstruction divided by the sum of the squared Frobenius norms of all the componentwise data reconstructions.

4 Conclusion

We developed a fully probabilistic PARAFAC2 model and demonstrated how orthogonality can be imposed in the context of variational inference in two different ways. Firstly, using the von Mises-Fisher matrix distribution assuming $\mathbb{E}[\boldsymbol{Y}^\top \boldsymbol{Y}] = \boldsymbol{I}$ as proposed in the context of variational PCA in [46]. Using this distribution forces all the realizations of the given matrix parameter to be orthogonal. Secondly, using the constrained matrix normal distribution assuming $\mathbb{E}[\boldsymbol{Y}^\top]\mathbb{E}[\boldsymbol{Y}] = \boldsymbol{I}$ in which the mean is constrained to the Stiefel manifold. This effectively results in a more flexible model as only the expectation of the realizations of the matrix are orthogonal and not the realizations themselves. For the latter approach we presently derived a simple closed form solution based on optimizing the lower bound.

Both probabilistic PARAFAC2 approaches were able to successfully recover the underlying signal in synthetic data both when considering homoscedastic and heteroscedastic added noise. However, we found that the specification of orthogonality based on vMF was more robust to noise than the specification based on $c\mathcal{MN}$. In particular, we observed substantial noise robustness in the probabilistic PARAFAC2 models when compared to the conventional direct fitting approach both when the correct model order was specified and when overestimating the number of components.

On the AAF data the probabilistic PARAFAC2 framework was able to correctly identify the underlying constituents and demonstrated improved robustness to model misspecification when compared to the conventional direct fitting algorithm. The ELBO of the models on this data suggest a model order of 3 components similar to the CCD and R2 heuristics computed from the direct fitting estimations. For the two gas-chromatography mass-spectrometry data GC-MS-WINE and GC-MS-TOBAC we also observed agreement between the probabilistic and direct fitting PARAFAC2 models but with more mixed results. The model order is not so clearly evident from the ELBO on these data sets. However, we see that the automatic relevance determination suppresses unnecessary components fairly well on both data sets ensuring robustness to overspecification of the model, which otherwise leads to degenerate solutions when the direct fitting approach is used. A few results from the probabilistic PARAFAC2 were not matching the results of the direct fitting approach. This can most likely be explained by encountering local maxima since variational methods are known to suffer from issues of underestimating uncertainty and thereby becoming overly confident on estimated parameters.

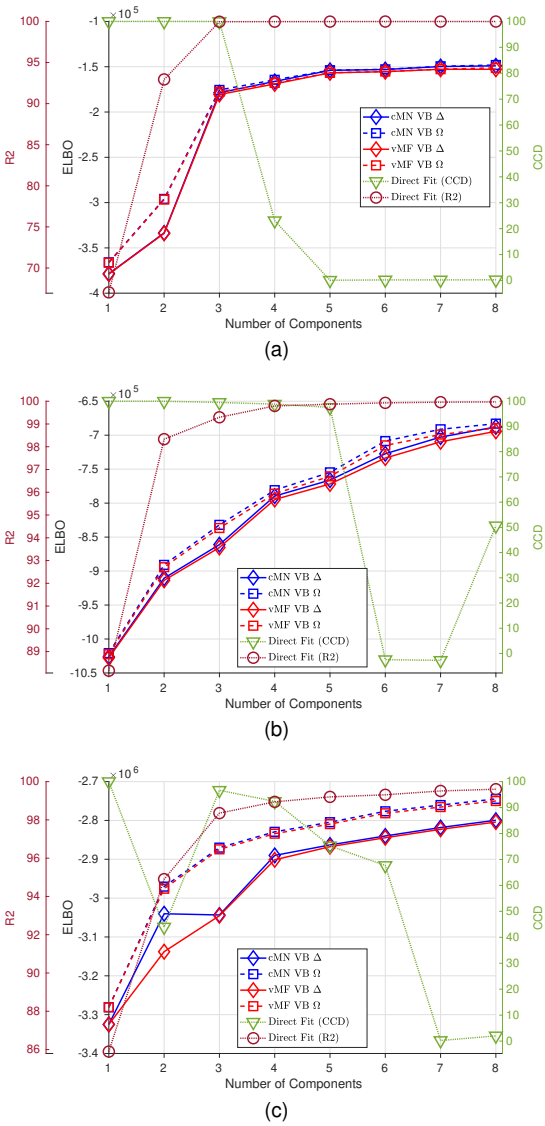


Fig. 3. Model selection criteria R2, CCD, and ELBO reported on the conventional PARAFAC2 and probabilistic PARAFAC2 models with 1 to 8 components on the AAF ((a)), GC-MS-WINE ((b)), and GC-MS-TOBAC ((c)) data sets. In the legend Δ indicates a homoscedastic noise model and Ω indicates a heteroscedastic noise model.

The proposed probabilistic PARAFAC2 models form an important step in the direction of applying probabilistic approaches to more advanced tensor decomposition approaches and a new direction for handling orthogonality constraints in probabilistic modeling in general using the proposed constrained matrix normal distribution framework that has a simple variational update. In particular, we anticipate that the orthogonality constraints within a prob-

abilistic setting may be useful also for the Tucker decomposition in which orthogonality is typically imposed [5] as well as for block-term decompositions [54] in which orthogonality may be beneficial to impose within each block or to improve identifiability within the CP decomposition by imposing orthogonality as implemented in the n-way toolbox (http://www.models.life.ku.dk/nwaytoolbox). PARAFAC2 is actively being advanced and employed for new applications, e.g. recently the higher-order block term decomposition has been embedded with a PARAFAC2 structure [55].

ACKNOWLEDGMENT

The authors would like to thank Rasmus Bro and Kuangda Tian for providing some of the data analyzed for this work.

REFERENCES

[1] J. D. Carroll and J. J. Chang, "Analysis of individual differences in multidimensional scaling via an n-way generalization of "Eckart-Young" decomposition," *Psychometrika*, vol. 35, no. 3, pp. 283–319, 1970.

[2] R. a. Harshman, "Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multimodal factor analysis," *UCLA Working Papers in Phonetics*, vol. 16, no. 10, pp. 1– 84, 1970.

[3] R. Bro, "Parafac. tutorial and applications," *Chemometrics and intelligent laboratory systems*, vol. 38, no. 2, pp. 149–171, 1997.

[4] C. J. Appellof and E. R. Davidson, "Strategies for analyzing data from video fluorometric monitoring of liquid chromatographic effluents," *Analytical chemistry*, vol. 53, no. 13, pp. 2053–2056, 1981.

[5] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.

[6] M. Mørup, "Applications of tensor (multiway array) factorizations and decompositions in data mining," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 1, no. 1, pp. 24–40, 2011.

[7] L. R. Tucker, "Some mathematical notes on three-mode factor analysis," *Psychometrika*, vol. 31, no. 3, pp. 279–311, 1966.

[8] F. L. Hitchcock, "The expression of a tensor or a polyadic as a sum of products," *Studies in Applied Mathematics*, vol. 6, no. 1-4, pp. 164–189, 1927.

[9] R. Bro, C. A. Andersson, and H. A. Kiers, "Parafac2-part ii. modeling chromatographic data with retention time shifts," *Journal of Chemometrics*, vol. 13, no. 3-4, pp. 295–309, 1999.

[10] R. A. Harshman and M. E. Lundy, "Uniqueness proof for a family of models sharing features of tucker's three-mode factor analysis and parafac/candecomp," *Psychometrika*, vol. 61, no. 1, pp. 133–154, 1996.

[11] J. M. ten Berge and H. A. Kiers, "Some uniqueness results for parafac2," *Psychometrika*, vol. 61, no. 1, pp. 123–132, 1996.

[12] H. A. Kiers, J. M. Ten Berge, and R. Bro, "Parafac2-part i. a direct fitting algorithm for the parafac2 model," *Journal of Chemometrics*, vol. 13, no. 3-4, pp. 275–294, 1999.

[13] B. M. Wise, N. B. Gallagher, and E. B. Martin, "Application of parafac2 to fault detection and diagnosis in semiconductor etch," *Journal of chemometrics*, vol. 15, no. 4, pp. 285–298, 2001.

[14] M. Weis, D. Jannek, F. Roemer, T. Guenther, M. Haardt, and P. Husar, "Multi-dimensional parafac2 component analysis of multi-channel eeg data including temporal tracking," in *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*. IEEE, 2010, pp. 5375–5378.

[15] K. H. Madsen, N. W. Churchill, and M. Mørup, "Quantifying functional connectivity in multi-subject fmri data using component models," *Human Brain Mapping*, 2016.

[16] P. A. Chew, B. W. Bader, T. G. Kolda, and A. Abdelali, "Cross-language information retrieval using parafac2," in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2007, pp. 143–152.

[17] Y. Panagakis and C. Kotropoulos, "Automatic music tagging via parafac2," in *acoustics, speech and signal processing (ICASSP), 2011 IEEE international conference on*. IEEE, 2011, pp. 481–484.

[18] E. Pantraki and C. Kotropoulos, "Automatic image tagging and recommendation via parafac2," in *Machine Learning for Signal Processing (MLSP), 2015 IEEE 25th International Workshop on*. IEEE, 2015, pp. 1–6.

[19] K. Tian, L. Wu, S. Min, and R. Bro, "Geometric search: A new approach for fitting PARAFAC2 models on GC-MS data," *Talanta*, vol. 185, pp. 378–386, Aug. 2018.

[20] I. Perros, E. E. Papalexakis, F. Wang, R. Vuduc, E. Searles, M. Thompson, and J. Sun, "SPARTan: Scalable PARAFAC2 for large & sparse data," Mar. 2017.

[21] Y. Cheng and M. Haardt, "Enhanced direct fitting algorithms for PARAFAC2 with algebraic ingredients," *IEEE Signal Process. Lett.*, vol. 26, no. 4, pp. 533–537, Apr. 2019.

[22] J. E. Cohen and R. Bro, "Nonnegative PARAFAC2: A flexible coupling approach," in *Latent Variable Analysis and Signal Separation*. Springer International Publishing, 2018, pp. 89–98.

[23] I. Porteous, E. Bart, and M. Welling, "Multi-hdp: A non parametric bayesian model for tensor factorization." in *Aaai*, vol. 8, 2008, pp. 1487–1490.

[24] G. Sheng, L. Denoyer, P. Gallinari, and G. Jun, "Probabilistic latent tensor factorization model for link pattern prediction in multi-relational networks," *The Journal of China Universities of Posts and Telecommunications*, vol. 19, pp. 172–181, 2012.

[25] A. Bhattacharya, D. B. Dunson *et al.*, "Sparse bayesian infinite factor models," *Biometrika*, vol. 98, no. 2, p. 291, 2011.

[26] H. Shan, A. Banerjee, and R. Natarajan, "Probabilistic tensor factorization for tensor completion," 2011.

[27] B. Ermis, Y. K. Yılmaz, a. T. Cemgil, and E. Acar, "Variational Inference For Probabilistic Latent Tensor Factorization with KL Divergence," 2014. [Online]. Available: http://arxiv.org/abs/1409.8083

[28] Z. Xu, F. Yan, and A. Qi, "Infinite tucker decomposition: Nonparametric bayesian models for multiway data analysis," in *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, 2012, pp. 1023–1030.

[29] Q. Zhao, L. Zhang, and A. Cichocki, "Bayesian cp factorization of incomplete tensors with automatic rank determination," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 9, pp. 1751–1763, 2015.

[30] V. Hore, A. Viñuela, A. Buil, J. Knight, M. I. McCarthy, K. Small, and J. Marchini, "Tensor decomposition for multiple-tissue gene expression experiments," *Nature Genetics*, vol. 48, no. 9, pp. 1094–1100, 2016.

[31] V. Beliveau, G. Papoutsakis, J. L. Hinrich, and M. Mørup, "Sparse probabilistic parallel factor analysis for the modeling of pet and task-fmri data," in *Bayesian and Graphiphal Models for Biomedical Imaging*. MICCAI, 2016.

[32] M. N. Schmidt and S. Mohamed, "Probabilistic non-negative tensor factorization using markov chain monte carlo," in *2009 17th European Signal Processing Conference*, Aug. 2009, pp. 1918–1922.

[33] Z. Xu, F. Yan, and Y. Qi, "Bayesian nonparametric models for multiway data analysis," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 2, pp. 475–487, 2015.

[34] Q. Zhao, G. Zhou, L. Zhang, A. Cichocki, and S.-I. Amari, "Bayesian robust tensor factorization for incomplete multiway data," *IEEE transactions on neural networks and learning systems*, vol. 27, no. 4, pp. 736–748, 2016.

[35] K. Hayashi, T. Takenouchi, T. Shibata, Y. Kamiya, D. Kato, K. Kunieda, K. Yamada, and K. Ikeda, "Exponential family tensor factorization: an online extension and applications," *Knowledge and information systems*, vol. 33, no. 1, pp. 57–88, 2012.

[36] L. Cheng, Y.-C. Wu, and H. V. Poor, "Probabilistic tensor canonical polyadic decomposition with orthogonal factors," *IEEE Transactions on Signal Processing*, vol. 65, pp. 663–676, 2017.

[37] C. M. Bishop, "Variational principal components," *9th International Conference on Artificial Neural Networks ICANN 99*, vol. 1999, no. 470, pp. 509–514, 1999. [Online]. Available: http://link.aip.org/link/IEECPS/v1999/iCP470/p509/s1{\&}Agg=doi

[38] R. A. Harshman, "PARAFAC2: Mathematical and technical notes," *UCLA Working Papers in Phonetics*, vol. 22, no. 10, pp. 30–44, 1972. [Online]. Available: http://www.bibsonomy.org/bibtex/2a964ff885ba59d4c7be518a3914f737a/threemode

[39] B. F. Green, "The Orthogonal Approximation of An Oblique Staructre in Factor Analysis," *Psychometrika*, vol. 17, no. 4, pp. 429–440, 1952.

[40] R. Bro, "PARAFAC. tutorial and applications," *Chemometrics Intellig. Lab. Syst.*, vol. 38, no. 2, pp. 149–171, 1997.

[41] R. Bro and H. A. Kiers, "A new efficient method for determining the number of components in parafac models," *Journal of chemometrics*, vol. 17, no. 5, pp. 274–286, 2003.
[42] M. H. Kamstrup-Nielsen, L. G. Johnsen, and R. Bro, "Core consistency diagnostic in parafac2," *Journal of Chemometrics*, vol. 27, no. 5, pp. 99–105, 2013.
[43] H. Attias *et al.*, "A variational baysian framework for graphical models." in *NIPS*, vol. 12, 1999.
[44] C. M. Bishop, *Pattern Recognition and Machine Learning*, 2006. [Online]. Available: http://www.library.wisc.edu/selectedtocs/bg0137.pdf
[45] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational Inference: A Review for Statisticians," pp. 1–33, 2016. [Online]. Available: http://arxiv.org/abs/1601.00670
[46] V. Šmídl and A. Quinn, "On Bayesian principal component analysis," *Computational Statistics & Data Analysis*, vol. 51, no. 9, pp. 4101–4123, 2007.
[47] R. Bro, "Multi-way analysis in the food industry: models, algorithms, and applications," Ph.D. dissertation, Københavns Universitet, Det Biovidenskabelige Fakultet for Fødevarer, Veterinærmedicin, 1998.
[48] N. Gillis and F. Glineur, "Nonnegative factorization and the maximum edge biclique problem," *arXiv preprint arXiv:0810.4225*, 2008.
[49] S. F. V. Nielsen and M. Mørup, "Non-negative tensor factorization with missing data for the modeling of gene expressions in the human brain," in *Machine Learning for Signal Processing (MLSP), 2014 IEEE International Workshop on*. IEEE, 2014, pp. 1–6.
[50] C. Khatri and K. Mardia, "The von mises-fisher matrix distribution in orientation statistics," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 95–106, 1977.
[51] H. A. Kiers, "A three-step algorithm for candecomp/parafac analysis of large data sets with multicollinearity," *Journal of Chemometrics*, vol. 12, no. 3, pp. 155–171, 1998.
[52] T. Skov, D. Ballabio, and R. Bro, "Multiblock variance partitioning: A new approach for comparing variation in multiple data blocks," *analytica chimica acta*, vol. 615, no. 1, pp. 18–29, 2008.
[53] J. M. Amigo, T. Skov, R. Bro, J. Coello, and S. Maspoch, "Solving gc-ms problems with parafac2," *TrAC Trends in Analytical Chemistry*, vol. 27, no. 8, pp. 714–725, 2008.
[54] L. De Lathauwer, "Decompositions of a higher-order tensor in block terms—part ii: Definitions and uniqueness," *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 3, pp. 1033–1066, 2008.
[55] C. Chatzichristos, E. Kofidis, M. Morante, and S. Theodoridis, "Blind fMRI source unmixing via higher-order tensor decompositions," *J. Neurosci. Methods*, vol. 315, pp. 17–47, Mar. 2019.

Philip J. H. Jørgensen received his B.Sc. and M.Sc. degrees in mathematical modeling and computation from the Technical University of Denmark (DTU) in 2014 and 2016, respectively. Currently, he is a Ph.D. student at the intersection of the statistics and data analysis section and cognitive systems section at DTU. His research focus includes probabilistic methods and lifelong machine learning.

Søren F. V. Nielsen Søren F. V. Nielsen completed a M.Sc. degree in mathematical modeling and engineering from the Technical University of Denmark (DTU) in 2015. He received his Ph.D. degree, also from DTU, in the field of Bayesian machine learning applied to neuroimaging in 2018. His thesis was centered around applying Bayesian model selection to dynamic functional connectivity models. He now works as a digital signal processing engineer in Sennheiser Communications.

Jesper Løve Hinrich received the B.Sc. degree in mathematics and technology and the M.Sc. degree in mathematical modeling and computation from the Technical University of Denmark in 2013 and 2016, respectively.

He is currently a Ph.D. student at the Technical University of Denmark, DTU Compute, Department for Cognitive Systems. His primary interests are in probabilistic multi-way modeling, machine learning and biomedical imaging.

Mikkel N. Schmidt is Associate Professor at DTU Compute, Technical University of Denmark. He is interested in probabilistic modeling and statistical machine learning with applications in both science and industry. His primary focus is the development of computational procedures for inference and validation.

Kristoffer H. Madsen Kristoffer H. Madsen received his Ph.D. degree from the Technical University of Denmark (DTU), he is currently heading the computational neuroimaging group at the Danish Research Centre for Magnetic Resonance and has an academic appointment as associate professor at DTU Compute. His primary research focus is on statistical machine learning for functional neuroimaging applications.

Morten Mørup Morten Mørup (mmor@dtu.dk) received his M.S. and Ph.D. degrees in applied mathematics at the Technical University of Denmark and he is currently associate professor at the Section for Cognitive Systems at DTU Compute, Technical University of Denmark. He has been associate editor of IEEE Transactions on Signal Processing. His research interests include machine learning, neuroimaging, and complex network modeling.

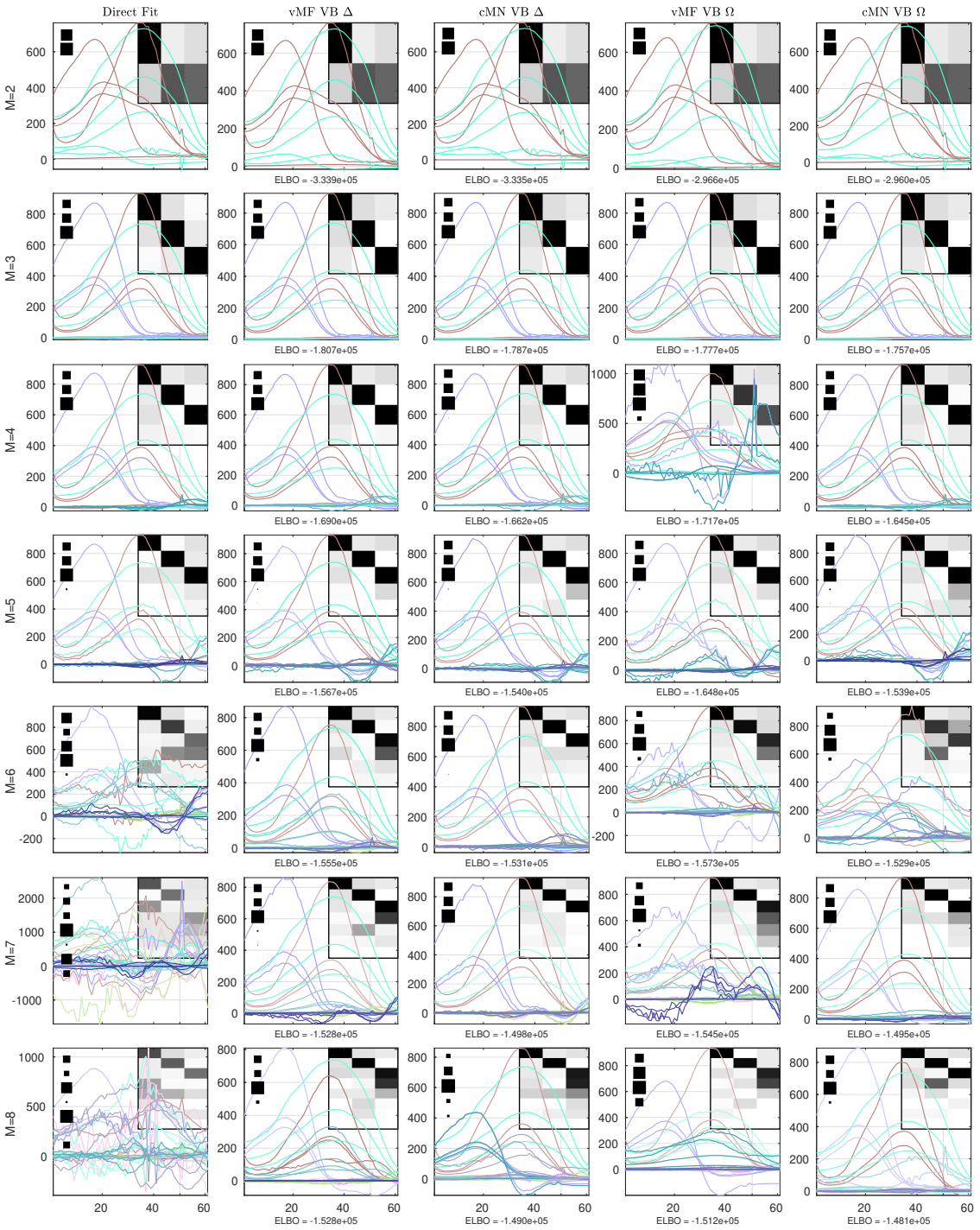


Fig. 4. The excitation loadings of the AAF data given by the conventional PARAFAC2 and probabilistic PARAFAC2 models. From top to bottom the loadings consist of 2 to 8 components. For each model the background heatmap visualizes the correlation between the data reconstruction for each identified component and the componentwise data reconstruction of the conventional PARAFAC2 model with 3 components (ground-truth). Furthermore, to the left a Hinton diagram indicates the relative squared Frobenius norm of the componentwise data reconstructions to the sum of them all. In the headers Δ indicates a homoscedastic noise model and Ω indicates a heteroscedastic noise model.

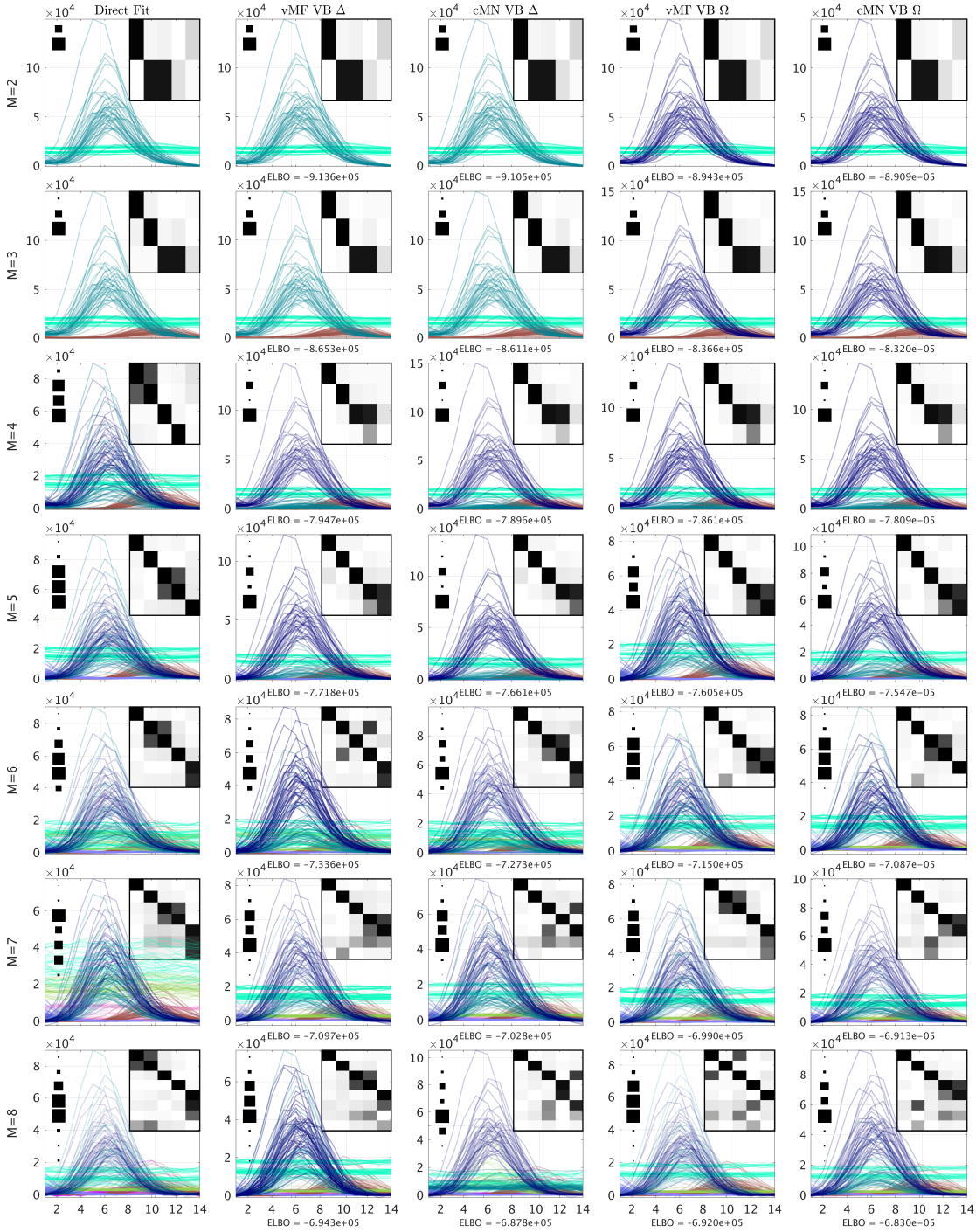


Fig. 5. The elution profiles of the GC-MS-WINE data given by the conventional PARAFAC2 and probabilistic PARAFAC2 models. From top to bottom the profiles consist of 2 to 8 components. For each model the background heatmap visualizes the correlation between the data reconstruction for each identified component and the componentwise data reconstruction of the conventional PARAFAC2 model with 5 components (expert conclusion). Furthermore, to the left a Hinton diagram indicates the relative squared Frobenius norm of the componentwise data reconstructions to the sum of them all. In the headers Δ indicates a homoscedastic noise model and Ω indicates a heteroscedastic noise model.

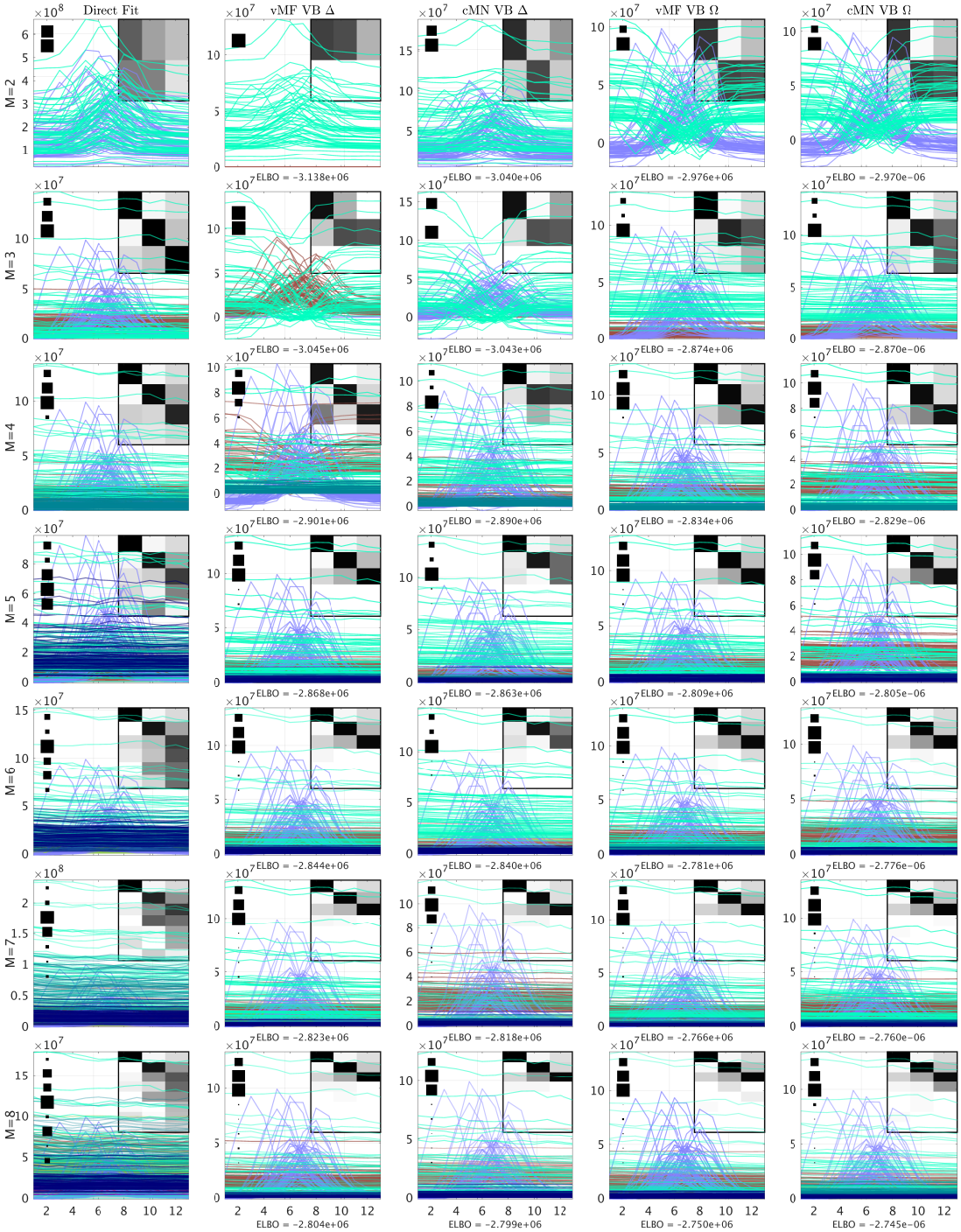


Fig. 6. The elution profiles of the GC-MS-TOBAC data given by the conventional PARAFAC2 and probabilistic PARAFAC2 models. From top to bottom the profiles consist of 2 to 8 components. For each model the background heatmap visualizes the correlation between the data reconstruction for each identified component and the componentwise data reconstruction of the conventional PARAFAC2 model with 3 components (expert conclusion). Furthermore, to the left a Hinton diagram indicates the relative squared Frobenius norm of the componentwise data reconstructions to the sum of them all. In the headers Δ indicates a homoscedastic noise model and Ω indicates a heteroscedastic noise model.

APPENDIX A
SOFTWARE

A MATLAB implementation of the probabilistic PARAFAC2 model was used to run all experiments and generate the results in the paper. The source code is made available on GitHub (https://github.com/philipjhj/VBParafac2) including a guide on setup and usage.

APPENDIX B
DERIVING THE VARIATIONAL INFERENCE

In the following we derive the most important expressions used to identify the update rules of the model parameters. Below is a overview of the used notation.

NOMENCLATURE

$\boldsymbol{A}$	Matrix (bold face uppercase)
$\boldsymbol{A}^\top$	Transpose of matrix $\boldsymbol{A}$
$\boldsymbol{a}_{i\cdot}$	i'th row of matrix $\boldsymbol{A}$ (bold face lowercase)
$\mathbf{a}_{\cdot i}$	i'th column of matrix $\boldsymbol{A}$ (bold face lowercase)
$\mathcal{X}$	Tensor / Multi-way array (calligraphy)
$\boldsymbol{X}_k$	k'th frontal slice (matrix) of tensor $\mathcal{X}$
$\boldsymbol{x}_{i\cdot k}$	i'th row of the k'th frontal slice $\boldsymbol{X}_k$ of tensor $\mathcal{X}$
$\mathrm{p}(\cdot)$	Probability density function (pdf)
$\mathrm{q}(\cdot)$	Variational distribution
$\mathbb{E}[\cdot]$	Expectation
$\mathbb{E}_{-z}[\cdot]$	Expectation with respect to all variables but z
$\mathrm{h}(\cdot)$	Entropy
c_x	Constant term(s) x irrelevant for the optimization problem at hand
$\circ$	The Hadamard product (element-wise)

B.1 The Evidence Lower Bound (ELBO)

An expansion of the evidence lower bound (ELBO) is shown here:

$$
\begin{aligned}
&\mathrm{ELBO}(\mathrm{q}(\boldsymbol{\theta})) \\
&\quad= \mathbb{E}[\log p(\mathcal{X},\boldsymbol{\theta})] - \mathbb{E}[\log \mathrm{q}(\boldsymbol{\theta})] \\
&\quad= \mathbb{E}[\log \mathrm{p}(\mathcal{X},\boldsymbol{A},\boldsymbol{C},\boldsymbol{F},\mathcal{P},\boldsymbol{\tau},\boldsymbol{\alpha})] \\
&\qquad - \mathbb{E}[\log \mathrm{q}(\boldsymbol{A},\boldsymbol{C},\boldsymbol{F},\mathcal{P},\boldsymbol{\tau},\boldsymbol{\alpha})] \\
&\quad= \mathbb{E}[\log \mathrm{p}(\mathcal{X} \mid \boldsymbol{A},\boldsymbol{C},\boldsymbol{F},\mathcal{P},\boldsymbol{\tau})\mathrm{p}(\boldsymbol{C} \mid \boldsymbol{\alpha})\mathrm{p}(\boldsymbol{F}) \\
&\qquad \mathrm{p}(\mathcal{P})\mathrm{p}(\boldsymbol{\tau})] \\
&\qquad - \mathbb{E}[\log \mathrm{q}(\boldsymbol{A})\mathrm{q}(\boldsymbol{C})\mathrm{q}(\boldsymbol{F})\mathrm{q}(\boldsymbol{P}_k)\mathrm{q}(\boldsymbol{\tau})] \\
&\quad= \mathbb{E}[\log \mathrm{p}(\mathcal{X} \mid \boldsymbol{A},\boldsymbol{C},\boldsymbol{F},\mathcal{P},\boldsymbol{\tau})] + \mathbb{E}[\log \mathrm{p}(\boldsymbol{A})] \\
&\qquad + \mathbb{E}[\log \mathrm{p}(\boldsymbol{C} \mid \boldsymbol{\alpha})] + \mathbb{E}[\log \mathrm{p}(\boldsymbol{F})] + \mathbb{E}[\log \mathrm{p}(\mathcal{P})] \\
&\qquad + \mathbb{E}[\log \mathrm{p}(\boldsymbol{\tau})] - \mathbb{E}[\log \mathrm{q}(\boldsymbol{A})] - \mathbb{E}[\log \mathrm{q}(\boldsymbol{C})] \\
&\qquad - \mathbb{E}[\log \mathrm{q}(\boldsymbol{F})] - \mathbb{E}[\log \mathrm{q}(\mathcal{P})] - \mathbb{E}[\log \mathrm{q}(\boldsymbol{\tau})] \\
&\quad= \mathbb{E}[\log \mathrm{p}(\mathcal{X} \mid \boldsymbol{A},\boldsymbol{C},\boldsymbol{F},\mathcal{P},\boldsymbol{\tau})] + \mathbb{E}[\log \mathrm{p}(\boldsymbol{A})] \\
&\qquad + \mathbb{E}[\log \mathrm{p}(\boldsymbol{C} \mid \boldsymbol{\alpha})] + \mathbb{E}[\log \mathrm{p}(\boldsymbol{F})] + \mathbb{E}[\log \mathrm{p}(\mathcal{P})] \\
&\qquad + \mathbb{E}[\log \mathrm{p}(\boldsymbol{\tau})] + \mathrm{h}(\mathrm{q}(\boldsymbol{A})) + \mathrm{h}(\mathrm{q}(\boldsymbol{C})) \\
&\qquad + \mathrm{h}(\mathrm{q}(\boldsymbol{F})) + \mathrm{h}(\mathrm{q}(\mathcal{P})) + \mathrm{h}(\mathrm{q}(\boldsymbol{\tau}))
\end{aligned}
$$

How to derive each of these terms is shown in the following.

B.2 Standard Moment Matching

As the formulation of the probabilistic PARAFAC2 model consists of the multivariate normal and gamma distribution we expand the logarithm of their general expressions below. This will serve as a reference for identifying the parameters of the variational distribution when reading the derivations of the update rules.

B.2.1 Multivariate Normal Distribution

Deriving the log density function of the multivariate normal distribution amounts to:

$$f(x_1,\ldots,x_k) \sim \mathcal{N}(\boldsymbol{X};\boldsymbol{\mu}_{\boldsymbol{X}},\boldsymbol{\Sigma}_{\boldsymbol{X}})$$

$$
\begin{aligned}
f(x_1,\ldots,x_k) =& (2\pi)^{-\frac{k}{2}}(|\boldsymbol{\Sigma}_{\boldsymbol{X}}|)^{-\frac{1}{2}} \\
&\exp(-\frac{1}{2}(\mathbf{X}-\boldsymbol{\mu}_{\boldsymbol{X}})^\top \boldsymbol{\Sigma}_{\boldsymbol{X}}^{-1}(\mathbf{X}-\boldsymbol{\mu}_{\boldsymbol{X}})) \\
\Rightarrow \ln f(x_1,\ldots,x_k) =& \ln\Big[(2\pi)^{-\frac{k}{2}}(|\boldsymbol{\Sigma}_{\boldsymbol{X}}|)^{-\frac{1}{2}} \\
&\exp\left(-\frac{1}{2}(\mathbf{X}-\boldsymbol{\mu}_{\boldsymbol{X}})^\top \boldsymbol{\Sigma}_{\boldsymbol{X}}^{-1}(\mathbf{X}-\boldsymbol{\mu}_{\boldsymbol{X}})\right)\Big] \\
=& -\frac{k}{2}\ln(2\pi) - \frac{1}{2}\ln(|\boldsymbol{\Sigma}_{\boldsymbol{X}}|) \\
&-\frac{1}{2}(\mathbf{X}-\boldsymbol{\mu}_{\boldsymbol{X}})^\top \boldsymbol{\Sigma}_{\boldsymbol{X}}^{-1}(\mathbf{X}-\boldsymbol{\mu}_{\boldsymbol{X}}) \\
=& -\frac{k}{2}\ln(2\pi) - \frac{1}{2}\ln(|\boldsymbol{\Sigma}_{\boldsymbol{X}}|) \\
&-\frac{1}{2}\mathbf{X}^\top \boldsymbol{\Sigma}_{\boldsymbol{X}}^{-1}\mathbf{X} - \frac{1}{2}\boldsymbol{\mu}_{\boldsymbol{X}}{}^\top \boldsymbol{\Sigma}_{\boldsymbol{X}}^{-1}\boldsymbol{\mu}_{\boldsymbol{X}} \\
&+\boldsymbol{\mu}_{\boldsymbol{X}}{}^\top \boldsymbol{\Sigma}_{\boldsymbol{X}}^{-1}\mathbf{X} \\
=& -\frac{1}{2}\ln(|\boldsymbol{\Sigma}_{\boldsymbol{X}}|) - \frac{1}{2}\mathbf{X}^\top \boldsymbol{\Sigma}_{\boldsymbol{X}}^{-1}\mathbf{X} \\
&-\frac{1}{2}\boldsymbol{\mu}_{\boldsymbol{X}}{}^\top \boldsymbol{\Sigma}_{\boldsymbol{X}}^{-1}\boldsymbol{\mu}_{\boldsymbol{X}} + \boldsymbol{\mu}_{\boldsymbol{X}}{}^\top \boldsymbol{\Sigma}_{\boldsymbol{X}}^{-1}\mathbf{X} + c
\end{aligned}
$$

where c is the constant terms with respect to $\boldsymbol{X}_k$ and its parameters.

B.2.2 Gamma Distribution

Deriving the log density function of the gamma distribution amounts to:

$$
\begin{aligned}
f(x;a,b) =& \frac{1}{\Gamma(a)b^a}x^{a-1}\exp(-xb^{-1}) \\
\Rightarrow \ln f(x;a,b) =& \ln\left[\frac{1}{\Gamma(a)b^a}x^{a-1}\exp(-xb^{-1})\right] \\
=& \ln\frac{1}{\Gamma(a)b^a} + (a-1)\ln x - xb^{-1} \\
=& (a-1)\ln x - xb^{-1} + c
\end{aligned}
$$

where c is the constant terms with respect to x.

B.3 Non-trivial Moment Matching

To identify the parameters for $\boldsymbol{C}$ and $\boldsymbol{F}$ non-trivial steps had to be performed.

B.3.1 The F Matrix

The variational factor for $\boldsymbol{F}$ is defined as:

$$\begin{aligned}\mathrm{q}(\boldsymbol{F}) &\propto \exp \mathbb{E}_{-\boldsymbol{F}}[\log \mathrm{p}(\mathcal{X}, \boldsymbol{\theta})] \\ &\propto \exp \mathbb{E}_{-\boldsymbol{F}}[\log \mathrm{p}(\mathcal{X}, \boldsymbol{F} \mid \boldsymbol{A}, \boldsymbol{C}, \mathcal{P}, \boldsymbol{\tau})]\end{aligned}$$

where

$$\begin{aligned}&\mathbb{E}_{-\boldsymbol{F}}[\log \mathrm{p}(\mathcal{X}, \boldsymbol{F} \mid \boldsymbol{A}, \boldsymbol{C}, \mathcal{P}, \boldsymbol{\tau})] \\ &= \mathbb{E}_{-\boldsymbol{F}}[\log \mathrm{p}(\mathcal{X} \mid \boldsymbol{A}, \boldsymbol{C}, \boldsymbol{F}, \mathcal{P}, \boldsymbol{\tau})] + \mathbb{E}_{-\boldsymbol{F}}[\log \mathrm{p}(\boldsymbol{F})] \\ &= \sum_k \sum_i \mathbb{E}_{-\boldsymbol{F}}[\log \mathrm{p}(\boldsymbol{x}_{i\cdot k} \mid \boldsymbol{a}_{i\cdot}, \boldsymbol{D}_k, \boldsymbol{F}, \boldsymbol{P}_k, \tau_k)] \\ &\quad + \sum_m \mathbb{E}_{-\boldsymbol{F}}[\log \mathrm{p}(\boldsymbol{f}_{m\cdot})] \\ &= \sum_k \sum_i \mathbb{E}_{-\boldsymbol{F}}[-\frac{1}{2}(\boldsymbol{a}_{i\cdot}\boldsymbol{D}_k\boldsymbol{F}^\top\boldsymbol{P}_k^\top)\mathbf{I}_M\tau_k(\boldsymbol{a}_{i\cdot}\boldsymbol{D}_k\boldsymbol{F}^\top\boldsymbol{P}_k^\top)^\top \\ &\quad + \boldsymbol{a}_{i\cdot}\boldsymbol{D}_k\boldsymbol{F}^\top\boldsymbol{P}_k{}^\top\mathbf{I}_M\tau_k\boldsymbol{x}_{i\cdot k}{}^\top] \\ &\quad + \sum_m \mathbb{E}_{-\boldsymbol{F}}[-\frac{1}{2}\boldsymbol{f}_{m\cdot}\mathbf{I}_M\boldsymbol{f}_{m\cdot}{}^\top] + c \\ &= -\frac{1}{2}\sum_k \sum_i \mathbb{E}_{-\boldsymbol{F}}[\tau_k(\boldsymbol{a}_{i\cdot}\boldsymbol{D}_k\boldsymbol{F}^\top\boldsymbol{P}_k{}^\top\boldsymbol{P}_k\boldsymbol{F}\boldsymbol{D}_k\boldsymbol{a}_{i\cdot}{}^\top)] \\ &\quad - \frac{1}{2}\sum_m \boldsymbol{f}_{m\cdot}\boldsymbol{f}_{m\cdot}{}^\top \\ &\quad + \sum_k \sum_i \mathbb{E}_{-\boldsymbol{F}}[\tau_k\boldsymbol{a}_{i\cdot}\boldsymbol{D}_k\boldsymbol{F}^\top\boldsymbol{P}_k{}^\top\boldsymbol{x}_{i\cdot k}{}^\top] + c \\ &= -\frac{1}{2}\sum_k \mathbb{E}[\tau_k]\sum_i \mathbb{E}_{-\boldsymbol{F}}[\boldsymbol{a}_{i\cdot}\boldsymbol{D}_k\boldsymbol{F}^\top\boldsymbol{P}_k{}^\top\boldsymbol{P}_k\boldsymbol{F}\boldsymbol{D}_k\boldsymbol{a}_{i\cdot}{}^\top] \\ &\quad - \frac{1}{2}\sum_m \boldsymbol{f}_{m\cdot}\boldsymbol{f}_{m\cdot}{}^\top \\ &\quad + \sum_k \sum_i \mathbb{E}[\tau_k]\mathbb{E}_{-\boldsymbol{F}}[\boldsymbol{a}_{i\cdot}\boldsymbol{D}_k\boldsymbol{F}^\top\boldsymbol{P}_k{}^\top\boldsymbol{x}_{i\cdot k}{}^\top] + c\end{aligned}$$

Again, we reorder the parameters using the trace operator to identify the quadratic term. This time the quadratic term separates into a quadratic and linear part revealing a linear intercomponent dependency.

$$\begin{aligned}&\mathbb{E}_{-\boldsymbol{F}}[\boldsymbol{a}_{i\cdot}\boldsymbol{D}_k\boldsymbol{F}^\top\boldsymbol{P}_k{}^\top\boldsymbol{P}_k\boldsymbol{F}\boldsymbol{D}_k\boldsymbol{a}_{i\cdot}{}^\top] \\ &= \mathbb{E}_{-\boldsymbol{F}}[\mathrm{Tr}(\boldsymbol{a}_{i\cdot}\boldsymbol{D}_k\boldsymbol{F}^\top\boldsymbol{P}_k{}^\top\boldsymbol{P}_k\boldsymbol{F}\boldsymbol{D}_k\boldsymbol{a}_{i\cdot}{}^\top)] \\ &= \mathbb{E}_{-\boldsymbol{F}}[\mathrm{Tr}(\boldsymbol{F}\boldsymbol{D}_k\boldsymbol{a}_{i\cdot}{}^\top\boldsymbol{a}_{i\cdot}\boldsymbol{D}_k\boldsymbol{F}^\top\boldsymbol{P}_k{}^\top\boldsymbol{P}_k)] \\ &= \mathrm{Tr}(\boldsymbol{F}\mathbb{E}_{-\boldsymbol{F}}[\boldsymbol{D}_k\boldsymbol{a}_{i\cdot}{}^\top\boldsymbol{a}_{i\cdot}\boldsymbol{D}_k]\boldsymbol{F}^\top\mathbb{E}_{-\boldsymbol{F}}[\boldsymbol{P}_k{}^\top\boldsymbol{P}_k]) \\ &= \sum_{mm\prime}(\boldsymbol{F}\mathbb{E}[\boldsymbol{D}_k\boldsymbol{a}_{i\cdot}{}^\top\boldsymbol{a}_{i\cdot}\boldsymbol{D}_k]\boldsymbol{F}^\top)_{mm\prime}(\mathbb{E}[\boldsymbol{P}_k{}^\top\boldsymbol{P}_k])_{mm\prime} \\ &= \sum_{mm\prime}\boldsymbol{f}_{m\cdot}\mathbb{E}[\boldsymbol{D}_k\boldsymbol{a}_{i\cdot}{}^\top\boldsymbol{a}_{i\cdot}\boldsymbol{D}_k]\mathbf{f}^{\mathrm{T}}_{m\prime\cdot}\mathbb{E}[\mathbf{p}^{\mathrm{T}}_{\cdot mk}\mathbf{p}_{\cdot m\prime k}] \\ &= \sum_m \boldsymbol{f}_{m\cdot}\mathbb{E}[\boldsymbol{D}_k\boldsymbol{a}_{i\cdot}{}^\top\boldsymbol{a}_{i\cdot}\boldsymbol{D}_k]\mathbb{E}[\mathbf{p}^{\mathrm{T}}_{\cdot mk}\mathbf{p}_{\cdot mk}]\mathbf{f}^{\mathrm{T}}_{m\cdot} \\ &\quad + 2\sum_m \sum_{m\prime\setminus m} \boldsymbol{f}_{m\cdot}\mathbb{E}[\boldsymbol{D}_k\boldsymbol{a}_{i\cdot}{}^\top\boldsymbol{a}_{i\cdot}\boldsymbol{D}_k]\mathbb{E}[\mathbf{p}^{\mathrm{T}}_{\cdot mk}\mathbf{p}_{\cdot m\prime k}]\mathbf{f}^{\mathrm{T}}_{m\prime\cdot}\end{aligned}$$

Again, we have to reorder and include the linear terms as before.

$$\begin{aligned}&\textstyle\sum_i \mathbb{E}_{-\boldsymbol{F}}[\boldsymbol{a}_{i\cdot}\boldsymbol{D}_k\boldsymbol{F}^\top\boldsymbol{P}_k{}^\top\boldsymbol{x}_{i\cdot k}{}^\top] \\ &= \sum_i \sum_m \mathbb{E}_{-\boldsymbol{F}}[\boldsymbol{a}_{i\cdot}\boldsymbol{D}_k\boldsymbol{f}_{m\cdot}{}^\top(\boldsymbol{P}_k{}^\top)_m\boldsymbol{x}_{i\cdot k}{}^\top] \\ &= \sum_i \sum_m \mathbb{E}[\boldsymbol{a}_{i\cdot}]\mathbb{E}[\boldsymbol{D}_k]\boldsymbol{f}_{m\cdot}{}^\top\mathbb{E}[(\boldsymbol{P}_k{}^\top)_m]\boldsymbol{x}_{i\cdot k}{}^\top \\ &= \sum_i \sum_m \mathbb{E}[(\boldsymbol{P}_k{}^\top)_m]\boldsymbol{x}_{i\cdot k}{}^\top\mathbb{E}[\boldsymbol{a}_{i\cdot}]\mathbb{E}[\boldsymbol{D}_k]\boldsymbol{f}_{m\cdot}{}^\top \\ &= \sum_m \mathbb{E}[(\boldsymbol{P}_k{}^\top)_m](\sum_i \boldsymbol{x}_{i\cdot k}{}^\top\mathbb{E}[\boldsymbol{a}_{i\cdot}])\mathbb{E}[\boldsymbol{D}_k]\boldsymbol{f}_{m\cdot}{}^\top\end{aligned}$$

Accounting for all terms and matching them to the ones in (B.2) we arrive at the following variational distribution for $\boldsymbol{F}$.

$$\mathrm{q}(\boldsymbol{F}) = \prod_m \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{f}_{m\cdot}}, \boldsymbol{\Sigma}_{\boldsymbol{f}_{m\cdot}})$$

with the parameters being

$$\begin{aligned}\boldsymbol{\mu}_{\boldsymbol{f}_{m\cdot}} &= \boldsymbol{\Sigma}_{\boldsymbol{f}_{m\cdot}}(\sum_k \mathbb{E}[\tau_k](\mathbb{E}[(\boldsymbol{P}_k{}^\top)_m]\boldsymbol{X}_k{}^\top\mathbb{E}[\boldsymbol{A}]\mathbb{E}[\boldsymbol{D}_k] \\ &\quad - \sum_i \mathbb{E}[\boldsymbol{D}_k\boldsymbol{a}_{i\cdot}{}^\top\boldsymbol{a}_{i\cdot}\boldsymbol{D}_k]\sum_{m\prime\setminus m}\mathbb{E}[\mathbf{p}^{\mathrm{T}}_{\cdot mk}\mathbf{p}_{\cdot m\prime k}]\mathbf{f}^{\mathrm{T}}_{m\prime\cdot})) \\ \boldsymbol{\Sigma}_{\boldsymbol{f}_{m\cdot}} &= (\sum_k \mathbb{E}[\tau_k]\sum_i \mathbb{E}[\boldsymbol{D}_k\boldsymbol{a}_{i\cdot}{}^\top\boldsymbol{a}_{i\cdot}\boldsymbol{D}_k]\mathbb{E}[\mathbf{p}^{\mathrm{T}}_{\cdot mk}\mathbf{p}_{\cdot m\prime k}] \\ &\quad + \mathbf{I}_M)^{-1}\end{aligned}$$

B.3.2 Constrained Matrix Normal Distribution

The orthogonality constraint in the model can be handled with two formulations. This section concerns the approach where the mean parameters of the variational approximation for $\boldsymbol{P}_k$ is constrained to be orthogonal and the following section describes the solution using the von-Mises Fisher distribution. Instead of using the free form variational updates we optimizes the ELBO with respect to the mean parameters $\boldsymbol{M}_{\boldsymbol{P}_k} = \mathbb{E}[\boldsymbol{P}_k]$ constrained to be orthogonal.

$$
\begin{aligned}
\boldsymbol{M}_{\boldsymbol{P}_k} = & \underset{\substack{\boldsymbol{M}_{\boldsymbol{P}_k} \\ \text{s.t. } \boldsymbol{M}_{\boldsymbol{P}_k}\boldsymbol{M}_{\boldsymbol{P}_k}{}^\top = \mathbf{I}}}{\arg\max} \quad \text{ELBO}(\boldsymbol{M}_{\boldsymbol{P}_k}) \\
= & \underset{\substack{\boldsymbol{M}_{\boldsymbol{P}_k} \\ \text{s.t. } \boldsymbol{M}_{\boldsymbol{P}_k}\boldsymbol{M}_{\boldsymbol{P}_k}{}^\top = \mathbf{I}}}{\arg\max} \quad \mathbb{E}[\log \mathrm{p}(\mathcal{X} \mid \boldsymbol{A}, \boldsymbol{C}, \boldsymbol{F}, \mathcal{P}, \boldsymbol{\tau})] \\
& + \mathbb{E}[\log \mathrm{p}(\boldsymbol{A})] + \mathbb{E}[\log \mathrm{p}(\boldsymbol{C} \mid \boldsymbol{\alpha})] \\
& + \mathbb{E}[\log \mathrm{p}(\boldsymbol{\alpha})] + \mathbb{E}[\log \mathrm{p}(\boldsymbol{F})] + \mathbb{E}[\log \mathrm{p}(\mathcal{P})] \\
& + \mathbb{E}[\log \mathrm{p}(\boldsymbol{\tau})] + \mathrm{h}(\mathrm{q}(\boldsymbol{A})) + \mathrm{h}(\mathrm{q}(\boldsymbol{C})) \\
& + \mathrm{h}(\mathrm{q}(\boldsymbol{F})) + \mathrm{h}(\mathrm{q}(\mathcal{P})) + \mathrm{h}(\mathrm{q}(\boldsymbol{\tau})) + \mathrm{h}(\mathrm{q}(\boldsymbol{\alpha})) \\
= & \underset{\substack{\boldsymbol{M}_{\boldsymbol{P}_k} \\ \text{s.t. } \boldsymbol{M}_{\boldsymbol{P}_k}\boldsymbol{M}_{\boldsymbol{P}_k}{}^\top = \mathbf{I}}}{\arg\max} \quad \mathbb{E}[\log \mathrm{p}(\mathcal{X} \mid \boldsymbol{A}, \boldsymbol{C}, \boldsymbol{F}, \mathcal{P}, \boldsymbol{\tau})] + c_1 \\
= & \underset{\substack{\boldsymbol{M}_{\boldsymbol{P}_k} \\ \text{s.t. } \boldsymbol{M}_{\boldsymbol{P}_k}\boldsymbol{M}_{\boldsymbol{P}_k}{}^\top = \mathbf{I}}}{\arg\max} \\
& - \frac{1}{2} \sum_k \sum_i \mathbb{E}[\tau_k (\boldsymbol{a}_{i\cdot} \boldsymbol{D}_k \boldsymbol{F}^\top \boldsymbol{P}_k{}^\top \boldsymbol{P}_k \boldsymbol{F} \boldsymbol{D}_k \boldsymbol{a}_{i\cdot}{}^\top)] \\
& + \sum_k \sum_i \mathbb{E}[\tau_k \boldsymbol{a}_{i\cdot} \boldsymbol{D}_k \boldsymbol{F}^\top \boldsymbol{P}_k{}^\top \boldsymbol{x}_{i\cdot k}{}^\top] + c_2 \\
= & \underset{\substack{\boldsymbol{M}_{\boldsymbol{P}_k} \\ \text{s.t. } \boldsymbol{M}_{\boldsymbol{P}_k}\boldsymbol{M}_{\boldsymbol{P}_k}{}^\top = \mathbf{I}}}{\arg\max} \quad \sum_k \sum_i \mathbb{E}[\tau_k \boldsymbol{a}_{i\cdot} \boldsymbol{D}_k \boldsymbol{F}^\top \boldsymbol{P}_k{}^\top \boldsymbol{x}_{i\cdot k}{}^\top] \\
& + c_3 \\
= & \underset{\substack{\boldsymbol{M}_{\boldsymbol{P}_k} \\ \text{s.t. } \boldsymbol{M}_{\boldsymbol{P}_k}\boldsymbol{M}_{\boldsymbol{P}_k}{}^\top = \mathbf{I}}}{\arg\max} \\
& \sum_k \mathbb{E}[\tau_k] \mathrm{Tr}(\mathbb{E}[\boldsymbol{A}] \mathbb{E}[\boldsymbol{D}_k] \mathbb{E}[\boldsymbol{F}^\top] \mathbb{E}[\boldsymbol{P}_k{}^\top] \boldsymbol{X}_k{}^\top) + c_3 \\
= & \underset{\substack{\boldsymbol{M}_{\boldsymbol{P}_k} \\ \text{s.t. } \boldsymbol{M}_{\boldsymbol{P}_k}\boldsymbol{M}_{\boldsymbol{P}_k}{}^\top = \mathbf{I}}}{\arg\max} \\
& \sum_k \mathbb{E}[\tau_k] \mathrm{Tr}(\mathbb{E}[\boldsymbol{F}] \mathbb{E}[\boldsymbol{D}_k] \mathbb{E}[\boldsymbol{A}^\top] \boldsymbol{X}_k \boldsymbol{M}_{\boldsymbol{P}_k}) + c_3
\end{aligned}
$$

Only the linear term of the probability density function of the data $\mathcal{X}$ depends on $\boldsymbol{M}_{\boldsymbol{P}_k}$ since $\boldsymbol{M}_{\boldsymbol{P}_k}$ in the quadratic terms is the identity matrix. Except for a scalar the optimization problem reduces to the same one as finding $\boldsymbol{P}_k$ in the alternating least squares algorithm, where one maximizes $\mathrm{Tr}(\mathbb{E}[\boldsymbol{F}]\mathbb{E}[\boldsymbol{D}_k]\mathbb{E}[\boldsymbol{A}^\top]\boldsymbol{X}_k\boldsymbol{M}_{\boldsymbol{P}_k})$ subject to the orthogonality constraint. The solution to this is found by simply applying a SVD as stated in the main text[1].

1. The alternating least squares method is described in *Kiers, Henk A. L., Jos M. F. Ten Berge, and Rasmus Bro. 1999. PARAFAC2 Part I. A Direct Fitting Algorithm for the PARAFAC2 Model. Journal of Chemometrics 13: 27594.* and the solution to the optimization problem was first described in *Green, Bert F. 1952. The Orthogonal Approximation of An Oblique Staructre in Factor Analysis. Psychometrika 17 (4): 42940.*

Analysis of Chromatographic Data using the Probabilistic PARAFAC2

Philip J. H. Jørgensen
Department of
Applied Mathematics and Computer Science
Technical University of Denmark
2800 Kgs. Lyngby, Denmark
phav@dtu.dk

Søren F. V. Nielsen
Research and Development
Sennheiser Communications
2750 Ballerup, Denmark
sfvnielsen@gmail.com

Jesper L. Hinrich
Department of
Applied Mathematics and Computer Science
Technical University of Denmark
2800 Kgs. Lyngby, Denmark
jehi@dtu.dk

Mikkel N. Schmidt
Department of
Applied Mathematics and Computer Science
Technical University of Denmark
2800 Kgs. Lyngby, Denmark
mnsc@dtu.dk

Kristoffer H. Madsen
Department of
Applied Mathematics and Computer Science
Technical University of Denmark
2800 Kgs. Lyngby, Denmark
khma@dtu.dk

Morten Mørup
Department of
Applied Mathematics and Computer Science
Technical University of Denmark
2800 Kgs. Lyngby, Denmark
mmor@dtu.dk

Abstract

PARAFAC2 is a widely applicable method often used for analyzing multi-way chromatographic data. We recently proposed a probabilistic framework for PARAFAC2[1]. The probabilistic formulations allow for a principled way of determining the number of latent components as well as modeling heteroscedastic noise. In this work we present a summary of the probabilistic PARAFAC2 models and their properties by revisiting the previous results of the analyzed data sets in a concise fashion.

1 Introduction

Multi-way analysis was originally developed within the field of psychometrics [2, 3], and since been used widely in other fields such as chemometrics [4]. Multi-way analysis appears in many fields of research including signal processing, neuroimaging, and information retrieval [5, 6]. The PARAFAC2 model, an extension of the CandeComp/PARAFAC (CP) model [2, 3, 7], was proposed by [8], has proven very useful for modeling chromatographic data handling variations occuring during experiments well[9, 10]. We recently proposed a probabilistic framework for the PARAFAC2 model for which we summarize the high-level details and more concisely present our results here.

Second Workshop on Machine Learning and the Physical Sciences (NeurIPS 2019), Vancouver, Canada.

2 Probabilistic PARAFAC2

Using the model formulation as described by [11], the three-way PARAFAC2 model can be written as,

$$\boldsymbol{X}_k = \boldsymbol{A}\boldsymbol{D}_k\boldsymbol{F}^\top\boldsymbol{P}_k^\top + \boldsymbol{E}_k \text{ s.t. } \boldsymbol{P}_k^\top\boldsymbol{P}_k = \mathbf{I}. \tag{1}$$

Based on this formulation of the PARAFAC2 model we developed two probabilistic PARAFAC2 formulations. The two formulations comes from the fact that in a probabilistic setting the orthogonality constraint $\boldsymbol{P}_k^\top\boldsymbol{P}_k = \boldsymbol{I}_M$ can be interpreted either as i) $\mathbb{E}[\boldsymbol{P}_k^\top\boldsymbol{P}_k] = \boldsymbol{I}_M$ or ii) $\mathbb{E}[\boldsymbol{P}_k]^\top\mathbb{E}[\boldsymbol{P}_k] = \boldsymbol{I}_M$. These formulations result in the following generative models i) and ii),

$$\boldsymbol{a}_{i\cdot} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_M), \boldsymbol{f}_{m\cdot} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_M), \boldsymbol{c}_{k\cdot} \sim \mathcal{N}(\mathbf{0}, \text{diag}(\boldsymbol{\alpha}^{-1})), \tau_k \sim \text{Gamma}(a_{\tau_k}, b_{\tau_k})$$
$$\text{i) } \boldsymbol{P}_k \sim \text{vMF}(\mathbf{0}), \text{ ii) } \boldsymbol{P}_k \sim \mathcal{MN}(\mathbf{0}, \mathbf{I}_J, \mathbf{I}_M)$$
$$\boldsymbol{X}_k \sim \mathcal{N}(\boldsymbol{A}\boldsymbol{D}_k\boldsymbol{F}^\top\boldsymbol{P}_k^\top, \tau_k^{-1}\mathbf{I}_J)$$

Using the notation where $\boldsymbol{a}_{i\cdot}$ denotes the ith row of the matrix $\boldsymbol{A}$, and where $\boldsymbol{\alpha}$ is a vector where each element defines the length scale of a corresponding component.

Variational Inference Choosing the mean-field approximation for these model formulations lead to the factorized variational distribution given as $\text{q}(\boldsymbol{\theta}) = q(\boldsymbol{A})q(\boldsymbol{C})\prod_m q(\boldsymbol{f}_{m\cdot})\prod_k q(\boldsymbol{P}_k)q(\tau_k)$. The update rules of the parameters for this distribution follow the standard iterative scheme and are described in detail in [1], as well as the corresponding evidence lower bound (ELBO). Note that careful attention had to paid to the updates of the $\boldsymbol{F}$ matrix due to intercomponent dependencies, as well as the updates for the constrained $\boldsymbol{P}_k$ matrix. In the following we outline the details of the updates of the two different variational distributions of $\boldsymbol{P}_k$.

Matrix Von Mises-Fisher Loadings The model formulation using i) constrains the expectation of the inner product of $\boldsymbol{P}_k$ to be orthogonal. This fully conforms to the conventional PARAFAC2 model ensuring that every realization of the loadings are orthogonal. The variational distribution of $\boldsymbol{P}_k$ has the density, $\text{vMF}(\boldsymbol{P}_k|\boldsymbol{B}_{\boldsymbol{P}_k}) = \kappa(J, \boldsymbol{B}_{\boldsymbol{P}_k}^\top\boldsymbol{B}_{\boldsymbol{P}_k})^{-1}\exp\big(\text{tr}[\boldsymbol{B}_{\boldsymbol{P}_k}^\top\boldsymbol{P}_k]\big)$ which has support only on the Stiefel manifold. Details on how this was computed this can be found in [1].

Constrained Matrix Normal Loadings The model formluation using ii) constrains the expectation of the loadings themselves to the orthogonal. This results in a more flexible model than i) as the realizations of the loadings are no longer constrained to be orthogonal. However, the interpretation of the orthogonal factor becomes identical to the direct fitting method and also the update rule is in closed form in a similar manner to the direct fitting solution as shown in [1].

Noise Modeling The probabilistc formulation allow for both modeling homoscedastic noise or heteroscedastic noise. Either τ_k can be updated collectively for all k or individually.

Model Selection In the probabilistic framework the scale vector $\boldsymbol{\alpha}$ is used for exploiting automatic relevance determination [12] by modeling the length scale of each component. Since the abililty to prune excess components is more of interest than uncertainty estimates on the length scales we proposed in [1] to use maximum a posteriori estimates instead of a variational estimate.

3 Results

In [1] the proposed models were evaluated on synthetic data and 3 real data sets; an amino acid fluorescence (AAF) data set and two gas chromatography mass spectrometry (GC-MS) data sets. In the following we revisit the results of the synthetic data and one of the GC-MS data sets. We refer to [1] for descriptions of the analyzed data and full details on these experiments as well as the results left out here.

Model Comparison Conventionally, different PARAFAC2 models have been compared by the ratio of explained variance and the core consistency diagnostic, respectively denoted R2 and CCD in [1] and the reminder of this work. These have been used to compare their ability to determine the correct number of components in relation to the ELBO used for the probabilistic models.

Synthetic Data The models were fitted to the synthetic data sets in order to investigate the ability to recover an underlying signal in different noise settings and noise levels, as seen in Figure 2. Also, a comparison of the different statistics for determining model order on these synthetic data sets can be seen in Figure 1.

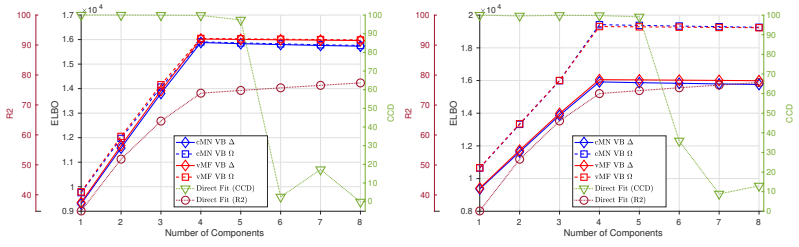


Figure 1: Model selection criteria given by the conventional PARAFAC2 and probabilistic PARAFAC2 models with 1 to 8 components on the synthetic data sets. In the legend Δ indicates a homoscedastic noise model and Ω indicates a heteroscedastic noise model.

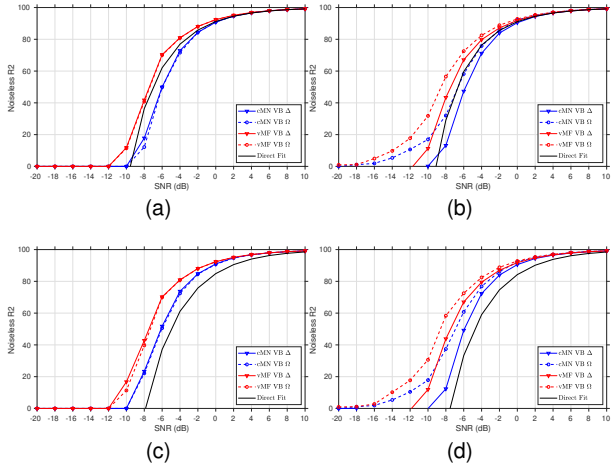


Figure 2: Noiseless R2 measured on different PARAFAC2 models fitted on synthetic data with varying levels of homoscedastic ((a),(c)) and heteroscedastic ((b),(d)) added noise. (a) and (b) show the result for models fitted with the true number of components (4 by design), and (c) and (d) for models with an overspecified number of components (6 by design) . In the legend Δ indicates a homoscedastic noise model and Ω indicates a heteroscedastic noise model.

Real Data We include the results from [1] on the GC-MS data originating from tobacco (GC-MS-TOBAC). For the different models we see the model selection performance based on the R2, CCD and ELBO in Figure 3 as well as the resulting elution profiles in Figure 4.

4 Discussion

The probabilistc PARAFAC2 model recently developed and analyzed in [1] shows promising results for delivering important properties such as the principled approach of performing model selection through automatic relevance determination and handling varying noise settings and increased noise

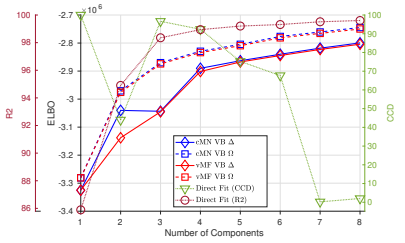


Figure 3: Model selection criteria given by the conventional PARAFAC2 and probabilistic PARAFAC2 models with 1 to 8 components on GC-MS-TOBAC data set. In the legend Δ indicates a homoscedastic noise model and Ω indicates a heteroscedastic noise model.

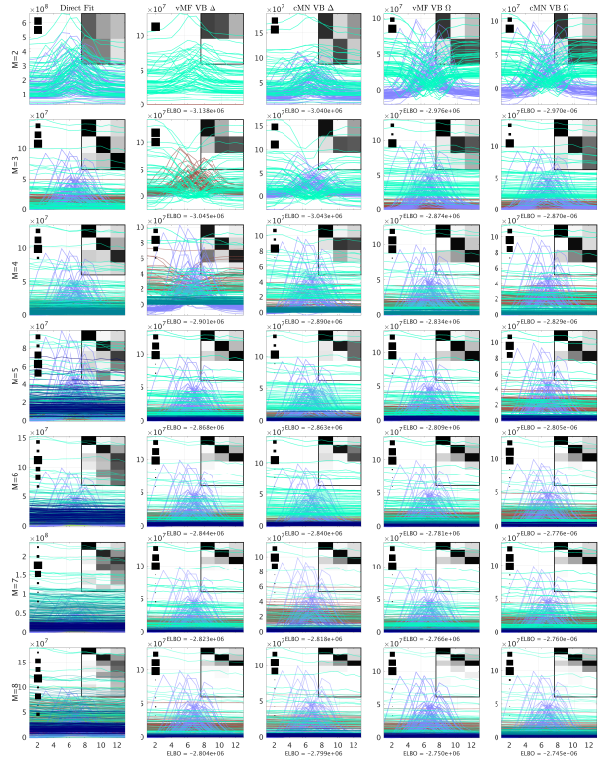


Figure 4: The resulting elution profiles of the GC-MS-TOBAC data given by the different PARAFAC2 models. From top to bottom the models is specified using model 2 to 8 components. The background heatmap visualizes the correlation between the data reconstruction for each identified component and the componentwise data reconstruction of the conventional PARAFAC2 model with 3 components (expert conclusion). Hinton diagrams indicate the relative squared Frobenius norm of the componentwise data reconstructions to the sum of them all to the left of each plot. In the headers Δ indicates a homoscedastic noise model and Ω indicates a heteroscedastic noise model.

levels, although known limitations of variational inference such as encountering local maxima are still present.

References

[1] Philip J. H. Jørgensen, Søren F. V. Nielsen, Jesper L. Hinrich, Mikkel N. Schmidt, Kristoffer H. Madsen, and Morten Mørup. Probabilistic parafac2, 2018, 1806.08195.

[2] J. Douglas Carroll and Jih Jie Chang. Analysis of individual differences in multidimensional scaling via an n-way generalization of "Eckart-Young" decomposition. *Psychometrika*, 35(3): 283–319, 1970. ISSN 00333123.

[3] Richard a Harshman. Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multimodal factor analysis. *UCLA Working Papers in Phonetics*, 16(10):1– 84, 1970.

[4] Rasmus Bro. Parafac. tutorial and applications. *Chemometrics and intelligent laboratory systems*, 38(2):149–171, 1997.

[5] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51 (3):455–500, 2009.

[6] Morten Mørup. Applications of tensor (multiway array) factorizations and decompositions in data mining. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(1): 24–40, 2011.

[7] Frank L Hitchcock. The expression of a tensor or a polyadic as a sum of products. *Studies in Applied Mathematics*, 6(1-4):164–189, 1927.

[8] Richard A. Harshman. PARAFAC2: Mathematical and technical notes. *UCLA Working Papers in Phonetics*, 22(10):30–44, 1972. URL `http://www.bibsonomy.org/bibtex/2a964ff885ba59d4c7be518a3914f737a/threemode`.

[9] José Manuel Amigo, Thomas Skov, Rasmus Bro, Jordi Coello, and Santiago Maspoch. Solving GC-MS problems with PARAFAC2. *TrAC - Trends in Analytical Chemistry*, 27(8):714–725, 2008.

[10] Lea G Johnsen, José Manuel Amigo, Thomas Skov, and Rasmus Bro. Automated resolution of overlapping peaks in chromatographic data. *J. Chemom.*, 28(2):71–82, 2014.

[11] Henk AL Kiers, Jos MF Ten Berge, and Rasmus Bro. Parafac2-part i. a direct fitting algorithm for the parafac2 model. *Journal of Chemometrics*, 13(3-4):275–294, 1999.

[12] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. 2006. ISBN 9780387310732. URL `http://www.library.wisc.edu/selectedtocs/bg0137.pdf`.

Probabilistic PARAFAC2 for Food Authentication

Philip J. H. Jørgensen
Department of Applied Mathematics and Computer Science
Technical University of Denmark
Kgs. Lyngby, Denmark

Morten Mørup
Department of Applied Mathematics and Computer Science
Technical University of Denmark
Kgs. Lyngby, Denmark

***Abstract*—Recent works have demonstrated the use of the PARAFAC2 decomposition as a preprocessing step to de-convolute GC-MS data to use the integrated peaks as features in further analysis downstream. These results indicate that such an approach can make it more widely accessible to analyze multiway GC-MS data. However, the number of components used for the PARAFAC2 still has to be determined, and the downstream analysis requires additional modeling steps. The probabilistic PARAFAC2 can decompose the data and simultaneously able to evaluate test samples using its approximated probability distributions. We investigate the performance of using both the probabilistic PARAFAC2 as a feature extractor similar to the conventional direct fitting algorithm, as well as directly using the probabilistic models for one-class classification. We find that the features extracted using the probabilistic PARAFAC2 yield more robust performance than those of the direct fitting algorithm or by transforming the three-way data into two-way data by for example summing across one of the modes. We see promising results for the one-class classification task on two publicly available GC-MS data sets as well. Furthermore, we propose a method for automatically determining a suitable number of components for the PARAFAC2 model based on the evidence lower bound, which our results indicate can be successful when the data are well explained by the PARAFAC2 model.**

***Keywords*—PARAFAC2, One-Class Classification, Probabilistic Models, Food Authentication**

I. Introduction

Today, most scientific fields generate enormous amounts of complex data intending to gain a deeper understanding which ultimately could allow for automatic decision-making. This could include classification or outlier detection, tasks often encountered in applications for regulatory agencies performing authentication, validation or identification on samples of food [17, 37], oil [16], or environmental forensics [12]. The present work is motivated by applications in food, but the approaches generalize to other domains as long as the data under consideration is suitable for the models.

A. Food Authentication

Economic interests in food trade have been the main driver for food fraud with food authentication being the response to combat it. [17, 35] defines authentication in food control as *"the confirmation of all requirements regarding the legal product description or the detection of fraudulent statements"* partially based on [32, 50], with the primary violations highlighted as 1) substitution by cheaper but similar ingredients, 2) extension of food using adulterant or blending and/or undeclared processes, 3) the origin such as geographic, species or production method. These violations also cover what studies on food fraud for beef, milk and olive oil are trying to detect [34]. Generally, food authentication is a field of high interest with exponential growth in publications since 2000 [14], but it has been a problem since the earliest societies [31, 39]. As the development of new data collection processes, as well as the fact that food products produced by mixing multiple ingredients or processing such as dairy, wine, olive oil remain difficult to authenticate, the potential for advanced analytics using machine learning is prominent [13].

Official control practice has a high demand for reliability as official objections should be standing "beyond reasonable doubt". Therefore, the choice of assessment of food authenticity is usually the robust and well-established targeted approaches where measurements of specific marker compounds in blind samples are evaluated to determine whether they violate or adhere to a control limit [17]. However, a targeted approach is limited by the fact that the objective of food fraudsters is to avoid food testing using knowledge of the test methods or sophisticated techniques not covered by the targeted analysis [34]. To overcome this, non-targeted analysis strives for obtaining and comparing so-called fingerprints of authentic and blind samples based on a more detailed description of their composition instead of being limited to specific marker compounds. These fingerprints are based on the principles of metabolomics, which is the scientific study of a comprehensive analysis of small molecules, the metabolites, in a biological system linking as many as possible to biochemistry, biology and physiology [42]. Food fingerprinting is not necessarily trying to identify the metabolites, but focusing on pattern recognition within the fingerprint matrix, which can be dramatically affected by the genetics of the agricultural commodities, the environmental context, as well as other external influences [17]. All these properties make non-targeted fingerprinting suitable for food authentication. A thorough overview of these approaches is given by [52].

B. Hyphenated Chromatographic Techniques

Mass spectrometry (MS) is a front-line technology across food science including food authentication as it benefits from high sensitivity, selectivity, throughput and multi-analyte capabilities [14]. Gas chromatography (GC) coupled with MS is a powerful and information-rich separation technique for multi-component mixtures of metabolites [4, 51]. GC-MS is useful for the analysis of naturally volatile or semi-volatile molecules [14]. The collection of chromatographic measurements is today highly reproducible, but still faces several sources of variability that result in several challenges. These variations include baseline drifts, peak shift across samples, low signal-to-noise ratios and overlapping or co-elution of the components [3,

4, 51], which makes it difficult to perform automatic curve resolution despite several software packages being readily available [47, 51].

C. Multiway Analysis

Multiway analysis [30] is applied to higher-order data arrays. Multiway data have been recorded using three or more discrete sets of "sensors" resulting in the same number of modes or ways. A matrix with a set of features recorded for some timestamps or subjects would be a two-way data array. If such a matrix has been collected under different experimental conditions, the collection of matrices would result in a three-way data array or tensor. GC-MS data can be collected for several samples resulting in a three-way data array with the three modes being the mass spectrum, elution time and samples. While GC-MS data can contain all of the problematic variations mentioned in the previous section, multiway analysis can be applied in an attempt to deal with them. Here the multiway model known as PARAFAC2, which is especially well suited for tackling many of the issues in chromatographic data [3, 9], is considered.

The PARAFAC2 model [21, 29] fitted using alternating least squares approaches have been used as a feature extractor to transform a full GC-MS multiway data array into a matrix. The matrix is subsequently analyzed with two-way methods such as principal component analysis (PCA) [3] for exploratory analysis of the effect of ripening times of apples or partial least squares discriminant analysis (PLS-DA) for classifying olive oil quality [44] or detection of prostate carcinoma [2]. To determine a suitable number of components for the PARAFAC2 model the PARADISe software can be employed [25], where a user evaluates PARAFAC2 models fitted using a varying number of components based on diagnostics describing the fit. These diagnostics include inspecting the residuals for systematic variance, the core consistency diagnostic [27], and if the fitted components visually appear meaningful. Since the PARAFAC2 model can provide food characterization similar to the less flexible PARAFAC model — which has many examples of its usefulness for food authentication [33] — it can also assist with performing food authentication.

Multiway analysis can by itself be applied for anomaly detection, but it is still in active development with Bayesian versions being one important direction of research [18]. PARAFAC2 has been used to perform fault detection and diagnosis in semiconductor etch [53] using a statistical test on the sum of the squared residuals computed by a calibration model. Anomaly detection is also commonly called outlier detection or one-class classification, and it has many applications [53] including food authentication.

This work investigates the use of the probabilistic PARAFAC2 models [26] both as a feature extractor similar to the use of the conventional PARAFAC2 model and as a so-called one-class model. A one-class model is typically fitted to samples of a specific class of interest only, allowing it to predict whether a test sample belongs to a said class or not — like a binary classifier not using information from the negative class. We also propose an algorithm based on the evidence lower bound (ELBO) to automatically determine the number of components in the model, which is one of the biggest challenges for using these models. The use of the probabilistic PARAFAC2 as a one-class model requires an estimation of a score for test samples that we propose to base on either the ELBO or the estimated concentration levels using their variational distribution. Previous work either uses an alternating least squares PARAFAC2 algorithm to compute a point estimate of these concentration levels used as features for downstream two-way models [2, 3, 44] or to estimate a test statistic from the residuals [53]. The proposed approach in this work takes advantage of the approximated distribution to circumvent the need for further modeling as the probabilistic PARAFAC2 model can be used directly to solve the task of one-class classification. This is done by measuring the average distance in terms of Kulllback-Leibler divergence between the distribution of training and test concentrations as well as associated lower bound on the log-likelihood of test samples provided by the probabilistic PARAFAC2 procedure.

II. Methods

A. Binary and One-Class Classification

For the task of authentication either a binary classifier or a one-class model can be used depending on the available data and preferences. Binary classifiers might be suitable if data of both the authentic samples and fraudulent samples are available, while one-class models only attempt to describe the authentic samples without using any examples of fraudulent samples. If fraudulent samples are well understood in terms of what types of fraud can be expected, the authenticator might want to fit models to fraudulent samples. On the other hand, if the possible types of fraud are complex, it might be necessary to use a one-class model, as it would be difficult to fully cover all types of fraud by gathering samples [37, 41].

In this work, we consider both of the above cases. We apply the widely used logistic regression (LR) method and support vector machine (SVM) [11, 46] for binary classification on the different features obtained from a GC-MS data set as described later. To investigate the use of one-class classification on the same features, we apply the one-class SVM (OC-SVM) model [45], which we compare to our proposal of directly estimating a score using the probabilistic PARAFAC2 models.

B. Models

1) PARAFAC2

For a multiway array $\mathcal{X}$ with K frontal slices denoted $\boldsymbol{X}_k$, each of size $I \times J$, the PARAFAC2 model can be written as:

$$\boldsymbol{X}_k = \boldsymbol{A}\boldsymbol{D}_k\boldsymbol{F}^\top\boldsymbol{P}_k^\top + \boldsymbol{E}_k \text{ s.t. } \boldsymbol{P}_k^\top\boldsymbol{P}_k = \mathbf{I}, \quad (1)$$

where $\boldsymbol{A}$ is a $I \times M$ matrix with the profiles of the components of the first mode shared across all the samples in the data. $\boldsymbol{D}_k$ is a $M \times M$ diagonal matrix with the intensity or concentration levels of the components as values when the other matrices are normalized and sign indeterminacy [10] has been handled. The values of these K diagonal matrices $\boldsymbol{D}_k$ are also commonly arranged as being the rows in a matrix $\boldsymbol{C}$ of size $K \times M$. This $\boldsymbol{C}$ matrix of the concentration levels of the components is the two-way matrix used for further modeling. $\boldsymbol{P}_k\boldsymbol{F}$ are the profiles of the second mode. Solving this model can be done with an alternating least squares algorithm applied directly on the model specification in (1) — hence referred to as the direct fitting algorithm [29].

The probabilistic PARAFAC2 model is obtained by specifying the PARAFAC2 model as a generative model as proposed by [26]. Here the model is specified as

$$
\begin{aligned}
& \boldsymbol{a}_{i\cdot} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_M), \\
& \boldsymbol{f}_{m\cdot} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_M), \\
& \boldsymbol{c}_{k\cdot} \sim \mathcal{N}(\mathbf{0}, \mathrm{diag}(\boldsymbol{\alpha}^{-1})), \\
\text{i)} \quad & \boldsymbol{P}_k \sim \mathrm{vMF}(\mathbf{0}), \\
\text{ii)} \quad & \boldsymbol{P}_k \sim \mathcal{MN}(\mathbf{0}, \mathbf{I}_J, \mathbf{I}_M), \\
& \tau_k \sim \mathrm{Gamma}(a_{\tau_k}, b_{\tau_k}), \\
& \boldsymbol{x}_{i\cdot k} \sim \mathcal{N}(\boldsymbol{a}_{i\cdot} \boldsymbol{D}_k \boldsymbol{F}^\top \boldsymbol{P}_k^\top, \tau_k^{-1} \mathbf{I}_J),
\end{aligned}
$$

where either i) or ii) is used depending on how the orthogonality constraint on $\boldsymbol{P}_k$ should be handled. The former is using a von Mises-Fisher matrix distribution [28] (vMF) – a distribution with support on the set of all orthonormal matrices of the given dimensions (the Stiefel manifold); the latter is using a standard matrix normal ($\mathcal{MN}$) distribution for the prior while constraining the mean of the approximated distribution to be orthonormal ($c\mathcal{MN}$). τ_k models the precision of the k'th frontal slice allowing for a heteroscedastic noise model, but can also be specified to be equal for all k resulting in $\boldsymbol{\tau}_1 = \ldots = \boldsymbol{\tau}_k$ and a homoscedastic noise model. The prior on $\boldsymbol{c}_{k\cdot}$ with the hyperparameters $\boldsymbol{\alpha}$ inferred on the data attempts to prune excess components using the idea of Automatic Relevance Determination (ARD) priors [6, 36].

The probabilistic PARAFAC2 is estimated through approximate Bayesian inference by its posterior distribution using variational inference (VI). VI minimizes the Kullback-Leibler (KL) divergence between the posterior distribution $\mathrm{p}(\boldsymbol{\theta}|\boldsymbol{X})$ and the approximate variational distribution $\mathrm{q}(\boldsymbol{\theta})$ [5]:

$$
q^\star(\boldsymbol{\theta}) = \arg\min \mathrm{KL}\big[q(\boldsymbol{\theta}) \big\| p(\boldsymbol{\theta}|\boldsymbol{X})\big].
$$

where the variational distribution is chosen to be the mean-field approximation given as $q(\boldsymbol{\theta}) = \prod_j q_j(\boldsymbol{\theta}_j)$, and the KL divergence is:

$$
\mathrm{KL}\big[q(\boldsymbol{\theta}) \big\| p(\boldsymbol{\theta}|\boldsymbol{X})\big] = \int q(\boldsymbol{\theta}) \log(\frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta}|\boldsymbol{X})}) d\boldsymbol{\theta}. \tag{2}
$$

As the task is to estimate the posterior $\mathrm{p}(\boldsymbol{\theta}|\boldsymbol{X})$, the KL divergence can not be directly evaluated, and instead, we compute the ELBO:

$$
\mathrm{ELBO}(q(\boldsymbol{\theta})) = \mathbb{E}[\log p(\boldsymbol{\theta}, \boldsymbol{X})] - \mathbb{E}[\log q(\boldsymbol{\theta})].
$$

Maximizing the ELBO is the same as minimizing the KL divergence as they only differ by the log-evidence, which is a constant as it does not depend on the variational distribution. Using the mean-field approximation results in the iterative update rules of the variational factors given by $q_j(\theta_j) \propto \exp\big(\mathbb{E}_{-j}[\log p(\theta_j, \boldsymbol{\theta}_{-j}, \boldsymbol{X})]\big)$ to find $q^\star(\boldsymbol{\theta})$ [7].

2) Fitting the Probabilistic PARAFAC2

The complexity of the PARAFAC2 model is controlled by the number of components M, and it is important to use the correct number of components as the solutions are not nested. Having nested solutions means that models with fewer components would be defined by a subset of the components of more complex models, but an increased number of components has implications for the orientation of all the components [20].

To determine the optimal number of components for mixtures of factor analyzers, [19] proposed a birth-death procedure to increase computational efficiency. This procedure removes any mixture component having zero probability in the variational distribution and uses a heuristic based on the mixture probabilities and variational objective function to attempt to create new components. Increasing complexity as necessary is computationally more efficient than using many components and having any excessive number prune away.

Using the same motivation as the birth-death procedure and taking into account that solutions are not nested, we fit the probabilistic PARAFAC2 in the following manner:

1) Set $M = 1$; fit a model with M components; let the ELBO of the fitted model be denoted $\mathrm{ELBO}_{\text{best}}$.
2) Fit a model with $M + 1$ components; let its resulting ELBO be denoted $\mathrm{ELBO}_{\text{new}}$.
3) If $\mathrm{ELBO}_{\text{best}} < \mathrm{ELBO}_{\text{new}}$ and $M \leq M_{\text{max}}$: let $\mathrm{ELBO}_{\text{best}} = \mathrm{ELBO}_{\text{new}}$, set $M = M + 1$, and repeat step 2-3; otherwise, terminate and return the model of $\mathrm{ELBO}_{\text{best}}$.

Here a model refers to any of the probabilistic PARAFAC2 variants. The fitting of any of the probabilistic PARAFAC2 models should be done using several initializations to minimize the risk of finding solutions in local minima. We use the solution estimated by the direct fitting algorithm to initialize the means of the variational factors.

Instead of setting the number of components M large enough to hopefully be greater than the true number of components while expecting the ARD prior to prune any extra components, the complexity is increased for each round. This approach avoids performing computations on sparse arrays. The initial complexity and its increase per iteration could be adjusted to larger values in case a high number of components is expected.

It is common to split the data into intervals across the retention time when modeling GC-MS data. These splits should only contain a limited number of components as expected from the number of peaks seen in the total ion chromatogram (TIC) as described by [25]. They recommend no more than six peaks should be present in a chosen interval, which motivates starting with attempting the simpler models.

3) Evaluating Test Samples

Given a variational distribution $q_{\text{train}}(\boldsymbol{\theta})$ on the parameters of the probabilistic PARAFAC2 model fitted to a collection of training samples arranged as a multiway array $\mathcal{X}_{\text{train}}$, we want to score a test sample $\mathbf{X}_{\text{test}}$ on how likely it is to come from the same variational distribution. To do so, we fit the variational factors $q_{\text{test}}(\mathbf{C})$ and $q_{\text{test}}(\mathcal{P})$ specific to the test sample given the variational factors $q_{\text{train}}(\mathbf{A})$ and $q_{\text{train}}(\mathbf{F})$ shared for all the training samples. The variational factor for the precision $\boldsymbol{\tau}$ is also fitted to the test sample if a heteroscedastic noise model is applied; otherwise it is $q_{\text{train}}(\boldsymbol{\tau})$ if a homoscedastic noise model is applied. This approach follows along the same lines as the one taken by [53] using the direct fitting algorithm for fault detection. This results in a probabilistic PARAFAC2 for the test sample(s) defined by the variational distribution:

$$q_{\text{test}}(\boldsymbol{\theta}) = q_{\text{train}}(\mathbf{A}, \mathbf{F}) q_{\text{test}}(\mathbf{c}, \mathcal{P}) q_{\text{noise model}}(\boldsymbol{\tau}) \tag{3}$$

To assert whether the test sample is authentic, we evaluate how similar the PARAFAC2 decomposition of $\mathbf{X}_{\text{test}}$ is to that of each sample in $\mathcal{X}_{\text{train}}$. Both of these decompositions, as described above, are based on the components in $q_{\text{train}}(\mathbf{A})$ with elution profiles having a covariance structure given by $q_{\text{train}}(\mathbf{F})$. We estimate the similarity between the decompositions based on the variational factors $q_{\text{test}}(\mathbf{c})$ and $q_{\text{train}}(\boldsymbol{C}) = \prod_k q^{(k)}_{\text{train}}(\mathbf{c})$ using the KL divergence:

$$\text{KL}_{\text{avg}}(q_{\text{train}}(\mathbf{C}), q_{\text{test}}(\mathbf{c})) = \frac{1}{K} \sum_k \text{KL}\big[q^{(k)}_{\text{train}}(\mathbf{c}) \big\| q_{\text{test}}(\mathbf{c})\big]. \tag{4}$$

This expresses the average KL divergence of the distributions $q^{(k)}_{\text{train}}(\mathbf{c})$ on the training samples from the distribution $q_{\text{test}}(\mathbf{c})$ on a test sample, or said another way, how well are the $q^{(k)}_{\text{train}}(\mathbf{c})$ explained by $q_{\text{test}}(\mathbf{c})$ on average. Since we only consider the variational factors of the parameters in $\mathbf{C}$, these should express the concentration levels of the components, which is the case when the other variational factors have unit scale.

The choice of only evaluating the similarity between the variational factors of $\mathbf{C}$ comes from the fact that the variational factors of $\mathbf{A}$ and $\mathbf{F}$ are considered shared between the training and test samples — so they are equal — while the variational factor of $\mathcal{P}$ is allowed to differ between samples to handle any retention time shifts. Ideally, the concentration levels $\mathbf{c}$ in an authentic test sample should be close to those found in known authentic samples; a fraudulent sample, however, should be expected to differ in concentration levels of the authentic components as it could have been diluted or mixed with components not found in the authentic samples. This is similar to the idea of using the PARAFAC2 model as a feature extractor where the concentration levels in $\mathbf{C}$ are considered for further modeling.

Another approach would be to use the ELBO as a score since it approximates the marginal likelihood that can be expressed as

$$\log p(\mathbf{X}_{\text{test}}) = \text{KL}\big[q_{\text{test}}(\boldsymbol{\theta}) \big\| p(\boldsymbol{\theta}|\mathbf{X}_{\text{test}})\big] + \text{ELBO}(q_{\text{test}}(\boldsymbol{\theta})). \tag{5}$$

where $q_{\text{test}}(\boldsymbol{\theta})$ is given by (3). The evidence of a fraudulent sample is expected to be lower than an authentic one which in practice might also be expressed by the ELBO, however, using a bound for model selection is not justified in theory [7].

Only point estimates are available without the probabilistic models, as when employing the direct fit model. Such point estimates does not account for uncertainty in the model parameters. Therefore, evaluating test samples should presumably lead to worse performance than posterior distributions estimated by probabilistic models. One heuristic to evaluate test samples based on the point estimates could be the explained variance. The explained variance measures the amount of variation in the data explained by the model [29]. This is computed as:

$$\text{R2} = 1 - \frac{\|\boldsymbol{X}_{\text{test}} - \boldsymbol{A}\boldsymbol{D}_{\text{test}}\boldsymbol{F}^\top \boldsymbol{P}^\top_{\text{test}}\|^2_{\mathcal{F}}}{\|\boldsymbol{X}_{\text{test}}\|^2_{\mathcal{F}}}. \tag{6}$$

where $||\cdot||_{\mathcal{F}}$ denotes the Frobenius norm. $\boldsymbol{D}_{\text{test}}$ and $\boldsymbol{P}_{\text{test}}$ are obtained analogously to the evaluation of test samples using the probabilistic models above.

C. Evaluation

The question of whether a test sample is to be considered identical to the authentic samples can be stated as either "is the test sample authentic?" or "is the test sample fraudulent?". As the main concern is whether a model successfully discovers fraud, the following evaluation will be based on the latter. This intentionally puts a focus on the samples not belonging to the authentic class, and therefore we refer to these samples as being positive for the evaluation. When we fit the models, especially the one-class models, the focus is on the samples of the target class, so they are commonly thought of as being positive. The effect of focusing on the non-authentic samples this way should become clear in the following.

We will use precision-recall (PR) curves and their area-under-the-curve (PR-AUC). A PR curve is the precision of a binary classifier as a function of its recall, where precision and recall are defined as

$$\text{precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}},$$
$$\text{recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}.$$

The true positives is the number of correctly classified positive samples, the false negatives is the number of incorrectly classified positive samples, and the false positives is the number of incorrectly classified negative samples. PR curves have been recognized to be better suited for imbalanced problems than the often considered Receiver Operating Characteristics (ROC) curves [43]. Identical to the ROC curves, the PR curves provide a model-wide evaluation of a binary class as it evaluates the performance for all possible thresholds. The PR curves can like the ROC curves be summarized by a single number by computing the area-under-the-curve. Given the scores of whether each sample belongs to the positive class, the PR-AUC can be computed as

$$\text{PR-AUC} = \sum_i \text{precision}_i(\text{recall}_i - \text{recall}_{i-1}), \tag{7}$$

where the index i indicates the precision and recall values at i'th threshold. A random classifier will have an expected precision equal to the fraction of positive samples for all recall levels, which is often referred to as the *baseline* precision curve. The expected PR-AUC of a random classifier is consequently equal to the fraction of positive samples as well.

One important aspect of the PR curve is the non-achievable region [8, 15]. The area of this region is a lower bound on the PR-AUC score dependent on the number of positive (n^+) and negative (n^-) samples in the data, and computed as [8]

$$\text{PR-AUC}_{\text{MIN}} = 1 + \frac{(1+\sigma)\ln(1-\sigma)}{\sigma}, \tag{8}$$

with $\sigma = n^+/(n^+ + n^-)$. This part of the PR-AUC is especially large for imbalanced problems with more positives than negatives. Since data sets we will encounter will contain few negative samples and consequently have a substantial $\text{PR-AUC}_{\text{MIN}}$, we

recommend using the area under the normalized PR curve (N-PR-AUC) given by

$$\text{N-PR-AUC} = \frac{\text{PR-AUC} - \text{PR-AUC}_{\text{MIN}}}{1 - \text{PR-AUC}_{\text{MIN}}}. \quad (9)$$

The baseline precision can also be normalized which we denote N-baseline in the following. [8] recommends using the N-PR-AUC for aggregated scores as it is less sensitive to the class imbalance across splits. A motivation for focusing on the non-authentic samples is that neither the precision nor recall is dependent on the number of correctly classified negative samples. This means that if the authentic samples were considered positive, the evaluation would not necessarily express the amount of non-authentic samples encountered in the test set. Consequently, the PR-AUC values could be identical for test sets of different sizes. Given a classifier evaluated on two different test set with sizes n_1^{test} and n_2^{test}, where $n_1^{\text{test}} \ll n_2^{\text{test}}$, and both with the same number of positive samples $n_{\text{pos}}^{\text{test}}$. If the ordered-by-classifier-score sequence of positive and negative examples of both test sets until reaching a recall of 1 is identical, then they will have an equal PR-AUC, despite the second test set being larger and hence more demanding. A simple example could be a classifier perfectly classifying the test sets resulting in a PR-AUC equal to 1 despite the more extensive test of the second classifier.

Another motivation is due to the small data set size in this work. For smaller-sized data with few positive examples, the change in PR-AUC values of the best performing classifiers can make large jumps. One example is the next-best classifier indicated by the dashed green line in Fig. 1 (a). This classifier only misclassifies a single negative example at a recall of $(n^+ - 1)/n^+ = 1/2$ before reaching a recall of 1. The PR-AUC of this classifier is equal to $1 * 1/2 + 2/3 * 1/2 = 5/6$, which means $1/6$ of the total area belongs exclusively to the perfect classifier. Another example could be the distribution of the area above and below the random classifier also in Fig. 1 (a) with $n^+ = 2$ and $n^- = 18$. The expected PR-AUC of the random classifier is equal to $2/20 = 1/10$ versus the perfect classifier with a PR-AUC equal to 1 results in a difference of $9/10$. On the other hand, the difference between the PR-AUC of the random classifier and the $\text{PR-AUC}_{\text{MIN}}$ is approximately equal to $1/10 - 1/2(1/19 + 2/20) \approx 0.0237$. The area becomes more uniformly distributed among the classifiers if a switch is made between the positive and negative samples as seen in Fig. 1 (e), especially for the N-PR-AUC scores. This is primarily a concern of the interpretation as the outcome is the same.

III. Results

A. Data

We consider two publicly available GC-MS data sets. The first data set[1] is on 44 wine samples originating from 4 different regions. The regions and the number of samples originating from each one are listed in TABLE I. The wine samples are all from the same grape but harvested in different geographical areas. The authors of the data set provide a detailed description of the instrumentation and process used to perform the measurements in [48]. They have made both a version of the raw GC-MS data matrix and a version where the data has been aligned available. Since PARAFAC2 can handle unaligned data, we use the raw data matrix in our experiments. The dimensionality of the data is $200 \times 2700 \times 44$ with mass-to-charge ratios of 5 to 204 at 2700 elution time points for the 44 samples. They also provide FT-IR measurements which we do not consider for this work.

The second data set[2] consists of 79 rice (*Oryza sativa*) samples [24, 47]. The public repository provides data on 80 samples, but 1 sample discarded due to inconsistencies as described by [47]. The samples are separated into two sets of classes: 4 different grain cultivars and their development as measured at 7, 10, 14, 28 and 42 days after flowering respectively. For simplicity, we here focus on the 4 classes of grain. See TABLE II for the class distribution. After preprocessing[3], the data dimensions ended up being $530 \times 20809 \times 79$ with a mass-to-charge ratio between 50 to 600 sampled with 2 scans per second for the 79 samples.

TABLE I. Wine Data

Class (Origin)	samples
Argentina	6
Chile	15
Australia	12
South Africa	11
Total	44

TABLE II. Rice Data

Class (Variant)	samples
Nipponbare	19
Qingfengai	20
9311	30
Nongken 58	20
Total	79

B. Implementation

The PARAFAC2 models were all implemented in MATLAB. The implementation of the direct fitting made by Rasmus Bro was used, which is available at: http://models.life.ku.dk/sites/default/files/parafac2.m. The probabilistic PARAFAC2 models are available from the code repository at: https://github.com/philipjhj/VBParafac2.

The implementation of logistic regression and SVM — both the binary and one-class model versions — were provided by the python library scikit-learn [38], which also provided the tools for performing the nested cross-validation described in the following.

C. Preprocessing

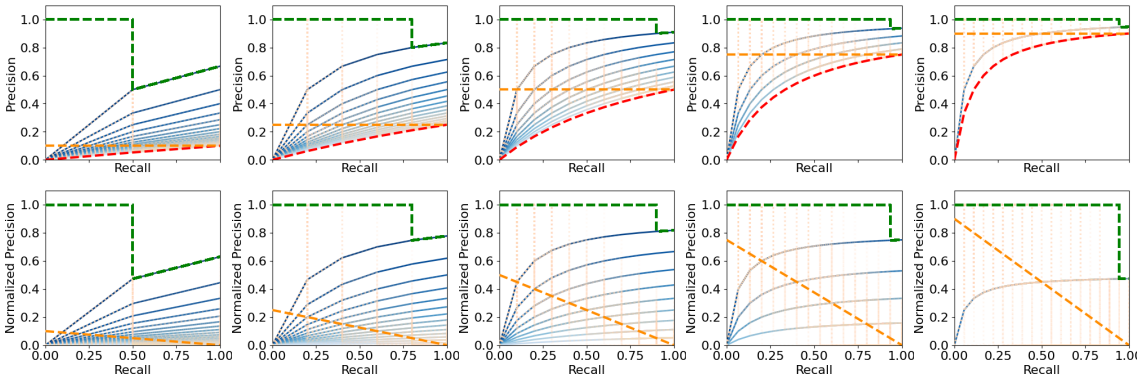
As previously described before fitting a PARAFAC2 model to the raw data, it should be split into shorter intervals as the PARAFAC2 model is better at handling data with fewer components. This is usually done on the TIC diagram and can either be done manually by hand-picking intervals with some number of peaks indicating one or more components as when using the PARADISe software [25], automatically through performing segmentation or simply splitting into equal-sized intervals. We use the automatic segmentation algorithm provided by [47][4], which identifies peaks in the TIC by locating maxima between identified minima. The segmentation depends

[1] http://www.models.life.ku.dk/Wine_GCMS_FTIR

[2] From Metabolights [22] and available at https://www.ebi.ac.uk/metabolights/MTBLS288/assays.

[3] Using the code made available by [47] at https://github.com/kkpsiren/vesi

[4] Made available here https://github.com/kkpsiren/vesi/blob/master/libs/peakdetect.py.



(a) $n^+ = 2, n^- = 18$ (b) $n^+ = 5, n^- = 15$ (c) $n^+ = 10, n^- = 10$ (d) $n^+ = 15, n^- = 5$ (e) $n^+ = 18, n^- = 2$

Fig. 1. Constructed PR and N-PR curves for a fixed data set size of $n = 20$ with varying ratios of positive (n^+) to negative (n^-) examples. The green dashed line is the next-to-perfect classifier, the orange dashed line is the baseline performance for a random classifier, the red dashed line is the minimum achievable precision, and the blue lines from top to bottom indicate the precisions of classifiers with an increasing amount (approximately 5%) of initial false positives before perfectly retrieving the positive samples. The light orange dotted lines are the precision curves of $10 \times n = 200$ randomly sampled classifiers visualized with low alpha values emphasizing repeated curves to indicate their variance.

on two parameters controlling how peaks are recognized, which we tuned manually until intervals with only a few peaks were identified. This circumvents the need to manually select intervals and does not discard any of the data. This resulted in 47 intervals used for the wine data, and 153 intervals used for the rice data.

D. Feature Extraction

For the two-way models we extracted a feature matrix by flattening the data as well as fitting the PARAFAC2 models on the full data. We took two approaches for flattening the data by summing over the mass spectrum (Flat_{ep}), also known as the TIC, and the retention times (Flat_{ms}) respectively similar to the approach taken in the original analysis on this data [48]. This was done both for the full data and the identified intervals used for the PARAFAC2 models [1]. Furthermore, we extracted the PCA scores explaining 99% of the variance fitted to the standard scores (mean-centered and scaled to unit variance) of the summarized data sets.

The features extracted using the PARAFAC2 models were the concentration levels values given by $\mathbf{C}$. For the probabilistic models, we used the mean of the variational factor of $\mathbf{C}$. A matrix $\mathbf{C}$ was identified for each interval and all of these were concatenated into a single two-way matrix of size number of samples $\times$ total number of components.

E. Experiments

For all the models we use a nested cross-validation scheme with 5 folds for both the inner and outer loop. With less than 5 positive samples for the inner loop, we use a leave-one-out scheme. The splits for the binary classifiers were overall classes and stratified. For the one-class models, the splits of the inner loop were only on the target class but the test set was extended with all samples from the non-target class for each fold of the outer loop. This way of splitting into train and test for one-class models lead to test sets with very few positive samples and many negatives. For the evaluation, as explained previously, the labels and scores were flipped and negated respectively when computing the performance measures.

For binary classification, we use logistic regression and support vector machines. Each of the 4 classes is used with a labeling of one-versus-rest for each class resulting in 4 data sets of "authentic" samples. The inner cross-validation loop performs a grid search on a selection of parameters. For the logistic regression, we tune the regularization parameter C searching in the range $[0.25, 2]$ with a step-size of 0.25 and a maximum number of iterations equal to 5000. For the support vector machine, the search parameters include the kernels: linear, radial basis function, sigmoid; as well as the kernel coefficient gamma controlling the influence of each training data point set to either auto or scale and with and without shrinking. These parameters were also used for the OC-SVM together with the parameter ν using the values $0.05, 0.10, 0.15$. This parameter represents the probability of misclassifying a positive sample. This is expected to be low when only training on positive samples. All other parameters in the implementations were kept as their default value. The data were standardized to zero mean and unit variance for all of these models.

We fitted the probabilistic models using the incremental approach described in Section II-B2 with the maximum number of allowed components set as $M_{\text{max}} = 10$. Each round was initialized 10 times with a maximum of 5000 iterations. We fitted all versions of the probabilistic model including using either the vMF or the $c\mathcal{MN}$variational factor for the $\boldsymbol{P}_k$ matrices and either a homoscedastic (Δ) or heteroscedastic (Ω) noise model.

The models estimated with the direct fitting algorithm were fitted to use the same number of components as the lowest number of components required by any of the probabilistic models. The models did not seem to struggle to use enough components, but rather too many, which was why the lowest number of components among the probabilistic models were

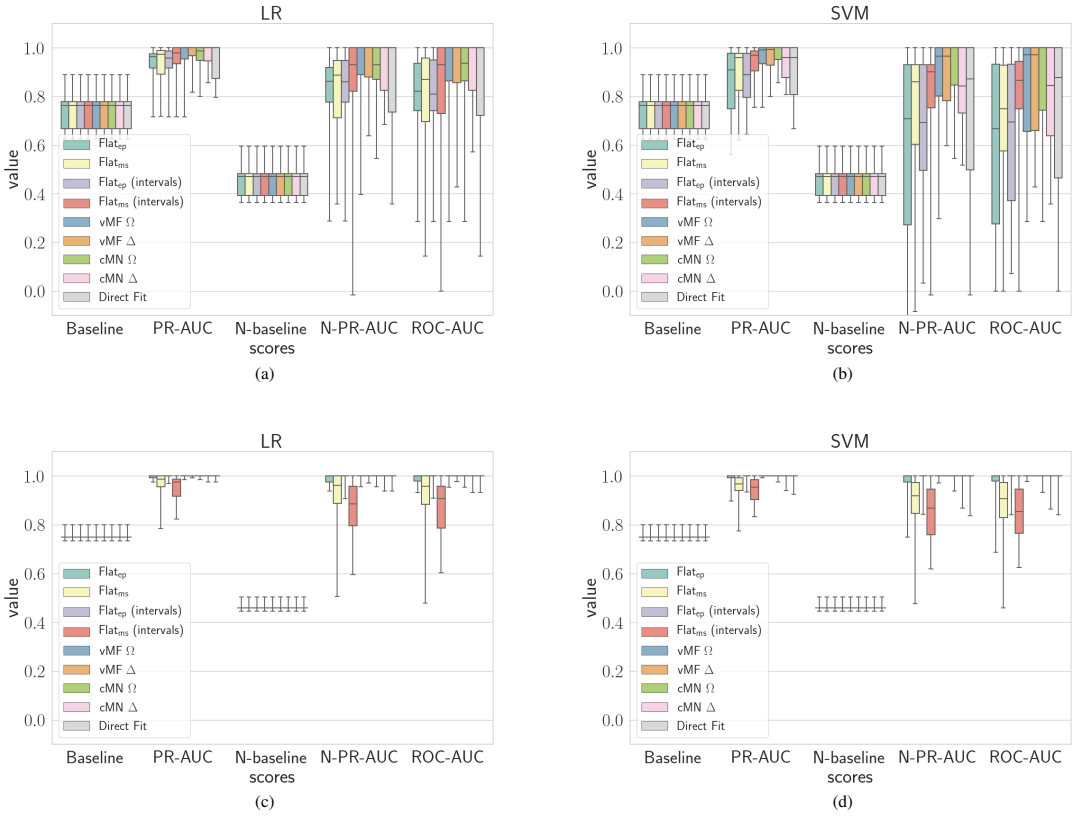


Fig. 2. The top figures (a,b) show box plots of the cross-validated baseline, PR-AUC, N-baseline, N-PR-AUC and ROC-AUC of the LR and SVM models respectively on the different features extracted from the wine data. The bottom figures (c,d) report the same on the different features extracted from the rice data. The top and bottom of the whiskers are the minimum and maximum values in the results respectively. The legend shows flatten data as Flat$_{ep}$: total ion current (TIC) chromatogram; Flat$_{ms}$: summed over retention times; *intervals* indicates the data was flattened per interval - and the features of the probabilistic PARAFAC2 models as: vMF: von Mises-Fisher matrix distribution on $\boldsymbol{P}_k$; $c\mathcal{MN}$: constrained multivariate normal distribution on $\boldsymbol{P}_k$; Δ: homoscedastic noise model; Ω: heteroscedastic noise model - and finally the features of the direct fit PARAFAC2 algorithm.

chosen.

We report both the PR-AUC and N-PR-AUC scores for the models. The ROC-AUC score is also reported for completeness. The results show the performance for all the 4 classes and the 5 splits, while the performance across splits and classes is shown in the N-PR curves in the figures.

F. Binary Classifiers

The result of the binary classifiers LR and SVM used for the different feature matrices is shown in Fig. 2. The baseline shows the expected PR-AUC of a random classifier, where the N-baseline has been normalized as in (9).

Fig. 2 (a,c) shows that the performance of the LR models using the different feature matrices are generally greater for the features extracted using PARAFAC2 on both data sets. The median performance of the direct fit features is very similar to those of the probabilistic models, however, the tail of the direct fit features is longer to the downside on the wine data. The rice data is almost classified perfectly by the LR models using either any of the PARAFAC2 based features or the flattened Flat$_{ep}$ (intervals) features. The other flattened data do not achieve this performance.

Similar results are observed for the SVM model in Fig. 2 (b,d). The performance of the SVM models on both data sets is very comparable to that of the LR models including the ranking among the different feature sets.

G. One-Class Models

1) One-Class SVM

The result of the OC-SVM used for the different feature sets is shown in Fig. 3 (a,e).

For this setting where the models are trained only on data of the authentic class, we considered in turn each class to be authentic with the others being counter-examples. This

allowed obtaining performances of the one-class classification task across the classes in the data.

In Fig. 3 (a,e) we see that the features extracted using the PARAFAC2 models achieve better performance than flattened data except for the Flat_{ep} (intervals) on the rice data. The direct fit PARAFAC2 features provide performance similar to those of the probabilistic PARAFAC2 models on both data sets.

2) Probabilistic PARAFAC2

The result of the one-class model based on the probabilistic PARAFAC2 models are shown in Fig. 3 (b,c,f,g) and Fig. 4 (a,b,d,e).

Here the probabilistic models fitted to the samples of the authentic class estimate a score for a test based on either (4) and (5). In Fig. 3 the score estimated for each test sample is summed across all the intervals. This approach yields lower performance than that of the OC-SVM for both scores. The ELBO performs better than the KL_{avg} estimates for both data sets. The KL_{avg} only performs better than the baseline (random) classifier on the wine data using the vMF based probabilistic models.

In Fig. 4 (a,b,d,e) the scores are reported for each interval. Without summing across the intervals the performance becomes much more spread out. The intervals with the best performance get close to that of the OC-SVM for both scores with a small advantage over the ELBO. The intervals with the worst performance are much lower than the baselines.

3) Direct Fit PARAFAC2

The result of the one-class model based on the explained variance of the direct fit PARAFAC2 models given by (6) is shown in Fig. 3 (d,h) and Fig. 4 (c,f).

In Fig. 3 (d,h) the R2 values have been summed across the intervals. Here the approach achieves better or similar performance to that of the KL_{avg} score, but lower than the ELBO based scores for both data sets.

In Fig. 4 (c,f) the result of each interval is reported. The effect of considering each interval is analogous to that of the ELBO and KL_{avg}. A much larger spread is observed with the best interval achieving performance similar to that of the OC-SVM and the worst much lower than the baseline.

H. Number of Components

In Fig. 5 (a,b) we see the identified number of components of the probabilistic PARAFAC2 models fitted for the one-class modeling problem. The models fitted to the wine data always determine a lower number of components than the maximum number of 10. However, on the rice data, most of the intervals use this maximum of 10 components. This might indicate that the intervals of the rice data contain too many components and should be made smaller, or that they are poorly described by the probabilistic PARAFAC2 models. In the appendix, the number of iterations used can be seen.

IV. Discussion

All the results based on the PARAFAC2 models are subject to solutions identified in local maxima, and the choice of the number of components for the direct fitting model is based on the result of the probabilistic models, which does not necessarily lead to the best solution for that model. Even though the risk of local maximas has been lowered by using multiple initializations, the effect of this might still be present in the presented results. Determining the correct number of components is a difficult task, and using an expert to evaluate the quality of the models is probably the most robust approach. Such an evaluation was not possible to include in this work, limiting the investigation to the approach taken here.

Using the probabilistic PARAFAC2 models for feature extraction shows promising results as those features can achieve higher performance compared to the flattened data and more robust in comparison to the features extracted using the direct fitting algorithm. This is true for both the task of binary classification using models commonly applied on two-way data such as LR and SVM, and most of one-class classification tasks using the OC-SVM models. On the rice data, we see that the Flat_{ep} (intervals) features achieve the highest performance using the OC-SVM model, which might be explained by the fact that many of the intervals uses all 10 components available for the PARAFAC2 models resulting in suboptimal solutions. To improve the performance of the PARAFAC2 based features, one might consider splitting the data into smaller intervals or increase the number of available components.

The one-class model based directly on the probabilistic PARAFAC2 models using either (5) or (4) highly depends on which interval is considered. Consequently, aggregating all the scores by summation does not seem to be the best approach. The performance of the individual intervals does indicate a potential for this approach as some of them achieve a better performance than obtained on the flattened data using the OC-SVM. Future work could consider how to identify which intervals have been resolved meaningfully making it possible to only aggregate those. Automatically identifying components that describe some chemical compound as done by [40] could also be considered. Other methods for comparing densities could also be considered such as relative density-ratio estimation [23, 49, 54]. Notably, the proposed scores of the probabilistic models achieve a better median performance than using the explained variance R2 as a score.

Another potential direction could be to do a bidirectional comparison as fraudulent samples could potentially resemble authentic ones, but the authentic samples might not fully resemble the fraudulent samples. Comparing the full PARAFAC2 decomposition of a set of test samples to the decomposition of the authentic samples could be considered for this. Also, the symmetrical version of the KL divergence could be used in (4), i.e. the KL divergence is computed in both directions.

The incremental fitting algorithm for the probabilistic PARAFAC2 successfully identified a number of components for the intervals of the wine data. However, the maximum of 10 components were selected for most of the intervals of the rice data. Also, previous results in [26] have indicated introducing excessive components does not necessarily result in a decrease in the ELBO for real data, while results on data with a simulated

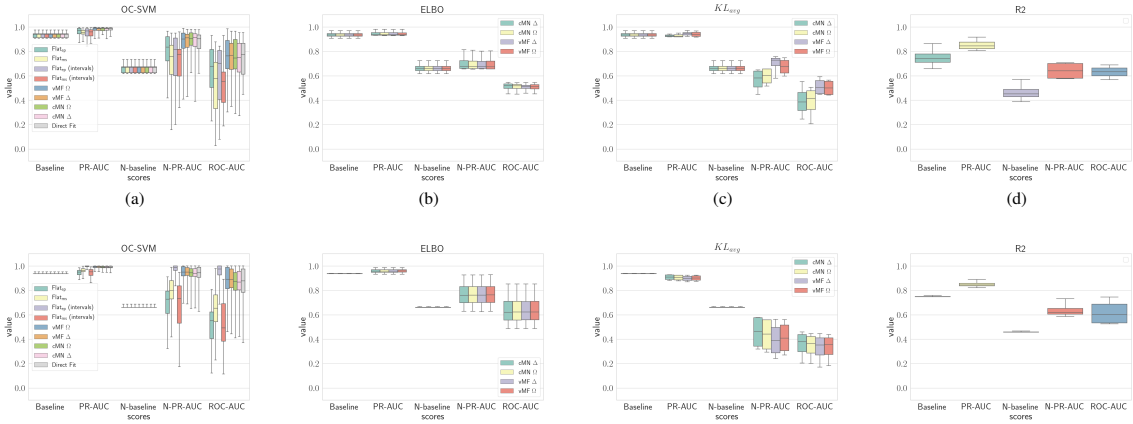


Fig. 3. The figures (a,b,c,d) show box plots of the cross-validated baseline, PR-AUC, N-baseline, N-PR-AUC and ROC-AUC of the OC-SVM using the previously extracted features, as well as the ELBO computed as (3), the KL_{avg} computed as (4) and R2 computed as (6) estimated using the different PARAFAC2 models evaluated on the wine data. The box plots are showing the values summed across all the intervals of the data set. The top and bottom of the whiskers are the minimum and maximum values in the results respectively. The bottom figures (e,f,g,h) report the same on the rice data. Legend lists the same features as in the previous figures for the OC-SVM models and the different probabilistic PARAFAC2 models for the ELBO and KL_{avg}. R2 is estimated based on the direct fit PARAFAC2.

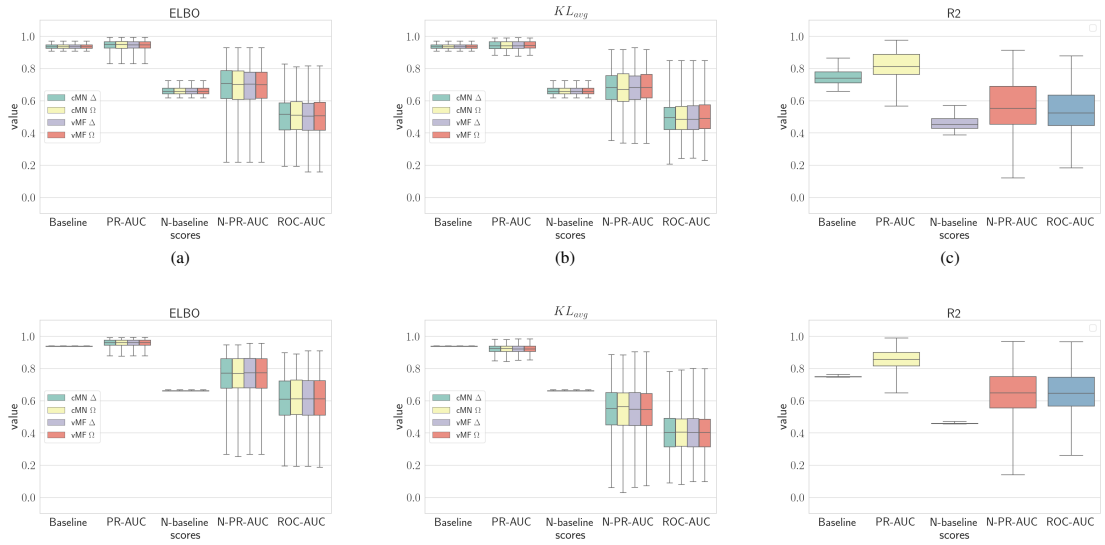
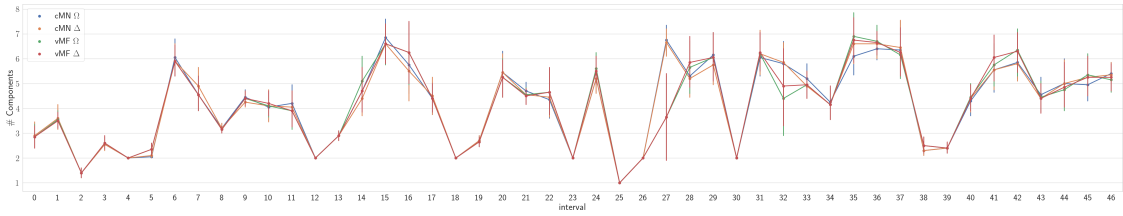
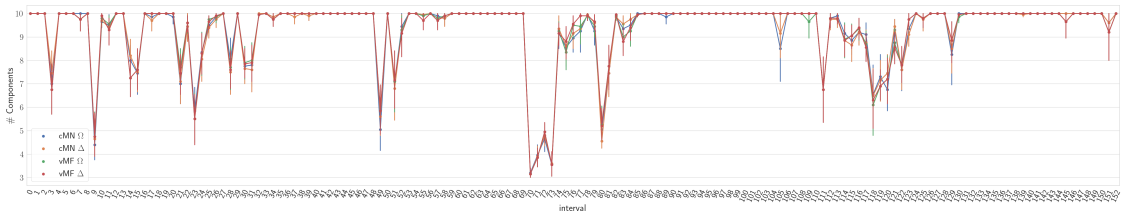


Fig. 4. The figures (a,b,c) show box plots of the cross-validated baseline, PR-AUC, N-baseline, N-PR-AUC and ROC-AUC of the ELBO computed as (3), the KL_{avg} computed as (4) and R2 computed as (6) estimated using the different PARAFAC2 models evaluated on the wine data. The box plots are showing the values for each interval of the data set. The top and bottom of the whiskers are the minimum and maximum values in the results respectively. The bottom figures (d,e,f) report the same on the rice data. Legend lists the different PARAFAC2 models used to obtain the ELBO and KL_{avg}. R2 is estimated based on the direct fit PARAFAC2.

PARAFAC2 structure revealed the ELBO to be maximum for the correct number of components. These results suggest that for the incremental fitting algorithm to be successful, the data must be approximated well by the assumptions in the PARAFAC2 model. The automatic segmentation algorithm might be changed or a higher number of components per interval could be specified in an attempt to improve the solutions. It could be considered to filter out intervals which are not well resolved with the PARAFAC2 model, e.g. if the maximum number of components is selected. Another area to pursue could be to find a way of determining if a new component is meaningful by perhaps using a stopping criterion based



(a)



(b)

Fig. 5. Figure (a) and (b) show the number of components chosen by the probabilistic models estimated by the algorithm proposed in Section II-B2 for each interval of the (a) wine and (b) rice data sets respectively. The probabilistic PARAFAC2 models are specified as vMF: von Mises-Fisher matrix distribution on $\boldsymbol{P}_k$; $c\mathcal{MN}$: constrained multivariate normal distribution on $\boldsymbol{P}_k$; Δ: homoscedastic noise model; Ω: heteroscedastic noise model.

on a minimum improvement in the ELBO from adding the component or the correlation of the new component with those components of the previous solution as adding a new excessive component has often been observed to partially explain already modeled structure throughout the experiments.

V. Conclusion

We have investigated how to use the probabilistic PARAFAC2 models for classification concerning authentication studies using GC-MS data. This work includes using the extracted concentration levels as features for conventional two-way models logistic regression and SVM. These features have been used in the setting of binary classification as well as one-class classification. Our results suggest a benefit of using the probabilistic PARAFAC2 models in both cases, as they achieve a higher performance than simply flattening the three-way data by summing and a more robust performance than on features extracted using the direct fitting PARAFAC2.

The incremental algorithm based on computing the ELBO of probabilistic PARAFAC2 models with an increasing number of components had some success in identifying a suitable model order on the wine data. It was more challenging on the rice data where the maximum number of components were used for many of the intervals. This might be because the rice data had many more intervals with too many components or they were too wide or both.

We proposed two approaches to one-class classification using either the ELBO of test samples fitted using the probabilistic PARAFAC2 model with the shared components estimated for the authentic class only or the KL divergence between the concentration levels obtained similarly. These were compared to both the previously mentioned feature extraction and the explained variance computing by the direct fitting PARAFAC2. The ELBO performed well compared to the KL divergence approach as well as the explained variance on both data sets but did not reach the performance of the OC-SVM models. These results could most likely be improved by filtering out the intervals that the PARAFAC2 model struggles to decompose well. These approaches could potentially eliminate the need to perform further modeling after having estimated the probabilistic PARAFAC2 models.

We have seen some promising results on using the probabilistic PARAFAC2 for these settings, both as an alternative to the conventional direct fitting algorithm, and a new approach based on comparing densities only available through the probabilistic formulations. In the future, how to filter the intervals such that only the meaningful ones are considered as well as other ways of combining the probabilistic PARAFAC2 models across intervals would be interesting to investigate.

VI. References

[1] Lawrence A Adutwum, Robin J Abel, and James Harynuk. “Total Ion Spectra versus Segmented Total Ion Spectra as Preprocessing Tools for Gas Chromatography - Mass Spectrometry Data”. en. In: *J. Forensic Sci.* 63.4 (July 2018), pp. 1059–1068.

[2] Eleonora Amante et al. “Untargeted Metabolomic Profile for the Detection of Prostate Carcinoma-Preliminary Results from PARAFAC2 and PLS-DA Models”. en. In: *Molecules* 24.17 (Aug. 2019).

[3] José Manuel Amigo et al. “Comprehensive analysis of chromatographic data by using PARAFAC2 and principal components analysis”. In: *J. Chromatogr. A* 1217.26 (2010), pp. 4422–4429.

[4] José Manuel Amigo et al. “Solving GC-MS problems with PARAFAC2”. In: *TrAC - Trends in Analytical Chemistry* 27.8 (2008), pp. 714–725.

[5] Hagai Attias. “A Variational Baysian Framework for Graphical Models”. In: *Advances in Neural Information Processing Systems 12*. Ed. by S A Solla, T K Leen, and K Mller. MIT Press, 2000, pp. 209–215.

[6] C M Bishop. “Variational principal components”. In: *9th International Conference on Artificial Neural Networks ICANN 99* 1999.470 (Jan. 1999), pp. 509–514.

[7] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. “Variational Inference: A Review for Statisticians”. In: (Jan. 2016), pp. 1–33. arXiv: 1601.00670 [stat.CO].

[8] Kendrick Boyd et al. “Unachievable Region in Precision-Recall Space and Its Effect on Empirical Evaluation”. en. In: *Proc. Int. Conf. Mach. Learn.* 2012 (Dec. 2012), p. 349.

[9] Rasmus Bro, Claus A Andersson, and Henk A L Kiers. “PARAFAC2—Part II. Modeling chromatographic data with retention time shifts”. In: *J. Chemom.* 13.3-4 (1999), pp. 295–309.

[10] Rasmus Bro, Riccardo Leardi, and Lea Giørtz Johnsen. “Solving the sign indeterminacy for multiway models: Solving sign indeterminacy”. In: *J. Chemom.* 27.3-4 (Mar. 2013), pp. 70–75.

[11] Corinna Cortes and Vladimir Vapnik. “Support-vector networks”. In: *Mach. Learn.* 20.3 (Sept. 1995), pp. 273–297.

[12] Hamidreza Ghasemi Damavandi. “Data analytics, interpretation and machine learning for environmental forensics using peak mapping methods”. PhD thesis. University of Iowa, 2016.

[13] Georgios P Danezis et al. “Food authentication: state of the art and prospects”. In: *Current Opinion in Food Science* 10 (Aug. 2016), pp. 22–31.

[14] Georgios P Danezis et al. “Food authentication: Techniques, trends & emerging approaches”. In: *Trends Analyt. Chem.* 85 (Dec. 2016), pp. 123–132.

[15] Jesse Davis and Mark Goadrich. *The relationship between Precision-Recall and ROC curves*. 2006.

[16] Diako Ebrahimi and D Brynn Hibbert. “Identification of sources of diesel oil spills using parallel factor analysis: a bridge between American Society for Testing and Materials and Nordtest methods”. en. In: *J. Chromatogr. A* 1198-1199 (July 2008), pp. 181–187.

[17] S Esslinger, J Riedl, and C Fauhl-Hassek. “Potential and limitations of non-targeted fingerprinting for authentication of food in official control”. In: *Food Res. Int.* 60 (June 2014), pp. 189–204.

[18] Hadi Fanaee-T and João Gama. “Tensor-based anomaly detection: An interdisciplinary survey”. In: *Knowledge-Based Systems* 98 (Apr. 2016), pp. 130–147.

[19] Zoubin Ghahramani and Mj Matthew J Beal. “Variational inference for Bayesian mixtures of factor analysers”. In: *Advances in Neural Information Processing Systems 12* 12 (2000), pp. 449–455.

[20] Chemometrics Group, Food Science, and The Royal Veterinary. “Parafac2—part ii. modeling chromatographic data with retention time shifts”. In: 309.July 1998 (1999), pp. 295–309.

[21] Richard A Harshman. “PARAFAC2: Mathematical and technical notes”. In: *UCLA Working Papers in Phonetics* 22.10 (1972), pp. 30–44.

[22] Kenneth Haug et al. “MetaboLights–an open-access general-purpose repository for metabolomics studies and associated meta-data”. en. In: *Nucleic Acids Res.* 41.Database issue (Jan. 2013), pp. D781–6.

[23] Shohei Hido et al. “Statistical outlier detection using direct density ratio estimation”. In: *Knowl. Inf. Syst.* 26.2 (Feb. 2011), pp. 309–336.

[24] Chaoyang Hu et al. “Identification of Conserved and Diverse Metabolic Shifts during Rice Grain Development”. en. In: *Sci. Rep.* 6 (Feb. 2016), p. 20942.

[25] Lea G Johnsen et al. “Gas chromatography - mass spectrometry data processing made easy”. en. In: *J. Chromatogr. A* 1503 (June 2017), pp. 57–64.

[26] Philip J H Jørgensen et al. “Probabilistic PARAFAC2”. In: (June 2018). arXiv: 1806.08195 [stat.ML].

[27] Maja H Kamstrup-Nielsen, Lea G Johnsen, and Rasmus Bro. “Core consistency diagnostic in PARAFAC2”. In: *J. Chemom.* 27.5 (2013), pp. 99–105.

[28] C G Khatri and K V Mardia. “The Von Mises-Fisher Matrix Distribution in Orientation Statistics”. In: *J. R. Stat. Soc. Series B Stat. Methodol.* 39.1 (1977), pp. 95–106.

[29] Henk A L Kiers, Jos M F Ten Berge, and Rasmus Bro. “PARAFAC2 — Part I . A Direct Fitting Algorithm for the PARAFAC2 Model”. In: *J. Chemom.* 13 (1999), pp. 275–294.

[30] Pieter M Kroonenberg. *Applied Multiway Data Analysis*. 2007, pp. 1–589.

[31] Radomir Lasztity, Marta Petro-Turza, and Tamas Foldesi. “HISTORY OF FOOD QUALITY STANDARDS”. In: *Food Quality and Standards*. Ed. by Radomir Lásztity. Vol. 1. Encyclopedia of Life Support Systems (EOLSS), Developed under the Auspices of the UNESCO, Eolss Publishers, Oxford ,UK, 2004.

[32] Michèle Lees. *Food authenticity and traceability*. Elsevier, 2003.

[33] Lea Lenhardt et al. “Modeling Food Fluorescence with PARAFAC: From Basics to Medical Applications”. In: *Calcium-Binding Proteins of the EF-Hand Superfamily*. unknown, Jan. 2018, pp. 161–197.

[34] Terry F McGrath et al. “What are the scientific challenges in moving from targeted to non-targeted methods for food fraud testing and how can they be addressed? – Spectroscopy case study”. In: *Trends Food Sci. Technol.* 76 (June 2018), pp. 38–55.

[35] Jeffrey C Moore, John Spink, and Markus Lipp. “Development and application of a database of food ingredient fraud and economically motivated adulteration from 1980 to 2010”. en. In: *J. Food Sci.* 77.4 (Apr. 2012), R118–26.

[36] Radford M Neal. *Bayesian Learning for Neural Networks*. en. Springer New York, Sept. 1996.

[37] Paolo Oliveri and Gerard Downey. “Multivariate class modeling for the verification of food-authenticity claims”. In: *Trends Analyt. Chem.* 35 (May 2012), pp. 74–86.

[38] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[39] Yuriy I Posudin, Kamaranga S Peiris, and Stanley J Kays. *Non-destructive detection of food adulteration to guarantee human health and safety*. http://ekmair.ukma.edu.ua/bitstream/handle/123456789/4214/Posudin_Non_destructive.pdf. Accessed: NA-NA-NA. 2015.

[40] Anne Bech Risum and Rasmus Bro. “Using deep learning to evaluate peaks in chromatographic data”. en. In: *Talanta* 204 (Nov. 2019), pp. 255–260.

[41] Oxana Ye Rodionova, Paolo Oliveri, and Alexey L Pomerantsev. “Rigorous and compliant approaches to one-class classification”. In: *Chemometrics Intellig. Lab. Syst.* 159 (Dec. 2016), pp. 89–96.

[42] U Roessner et al. “1.33 - Metabolomics – The Combination of Analytical Biochemistry, Biology, and Informatics”. In: *Comprehensive Biotechnology (Second Edition)*. Ed. by Murray Moo-Young. Burlington: Academic Press, Jan. 2011, pp. 447–459.

[43] Takaya Saito and Marc Rehmsmeier. “The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets”. en. In: *PLoS One* 10.3 (Mar. 2015), e0118432.

[44] Carlos Sales Martınez et al. “Olive oil quality classification and measurement of its organoleptic attributes by untargeted GC–MS and multivariate statistical-based approach”. In: (2019).

[45] B Schölkopf et al. “Estimating the support of a high-dimensional distribution”. en. In: *Neural Comput.* 13.7 (July 2001), pp. 1443–1471.

[46] Bernhard Schölkopf. “Support vector learning”. PhD thesis. Oldenbourg München, Germany, 1997.

[47] Kimmo Sirén, Ulrich Fischer, and Jochen Vestner. “Automated supervised learning pipeline for non-targeted GC-MS data analysis”. In: *Analytica Chimica Acta: X* 1 (Mar. 2019), p. 100005.

[48] Thomas Skov, Davide Ballabio, and Rasmus Bro. “Multiblock variance partitioning: A new approach for comparing variation in multiple data blocks”. In: *Anal. Chim. Acta* 615.1 (May 2008), pp. 18–29.

[49] A Smola, L Song, and C H Teo. “Relative novelty detection”. In: *Artificial Intelligence and Statistics* (2009).

[50] Gary R Takeoka and Susan E Ebeler. “Progress in Authentication of Food and Wine”. In: *Progress in Authentication of Food and Wine*. Vol. 1081. ACS Symposium Series. American Chemical Society, Jan. 2011, pp. 3–11.

[51] Jochen Vestner et al. “Toward automated chromatographic fingerprinting: A non-alignment approach to gas chromatography mass spectrometry data”. en. In: *Anal. Chim. Acta* 911 (Mar. 2016), pp. 42–58.

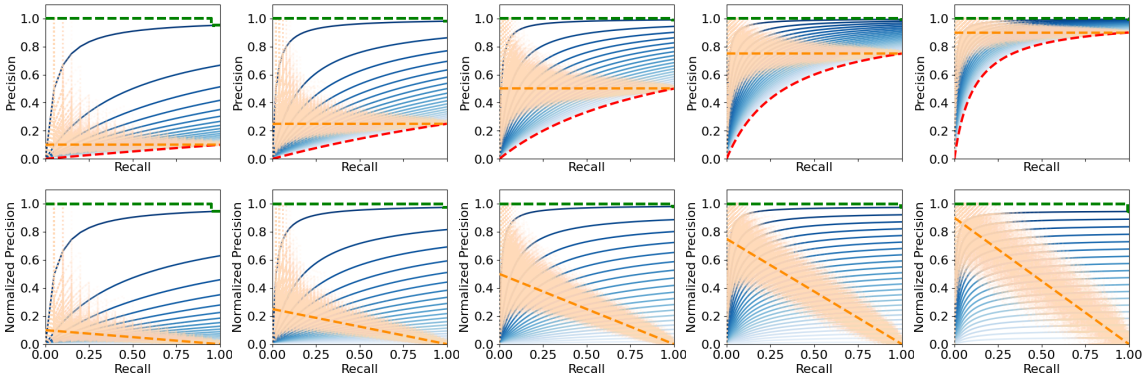
[52] Amelie Sina Wilde. “Detection of Food Fraud in high value products-Exemplary authentication studies on Vanilla, Black Pepper and Bergamot oil”. In: (2019).

[53] Barry M Wise, Neal B Gallagher, and Elaine B Martin. “Application of PARAFAC2 to fault detection and diagnosis in semiconductor etch”. In: *J. Chemom.* 15.4 (May 2001), pp. 285–298.

[54] Makoto Yamada et al. “Relative Density-Ratio Estimation for Robust Distribution Comparison”. In: (June 2011). arXiv: 1106.4729 [stat.ML].

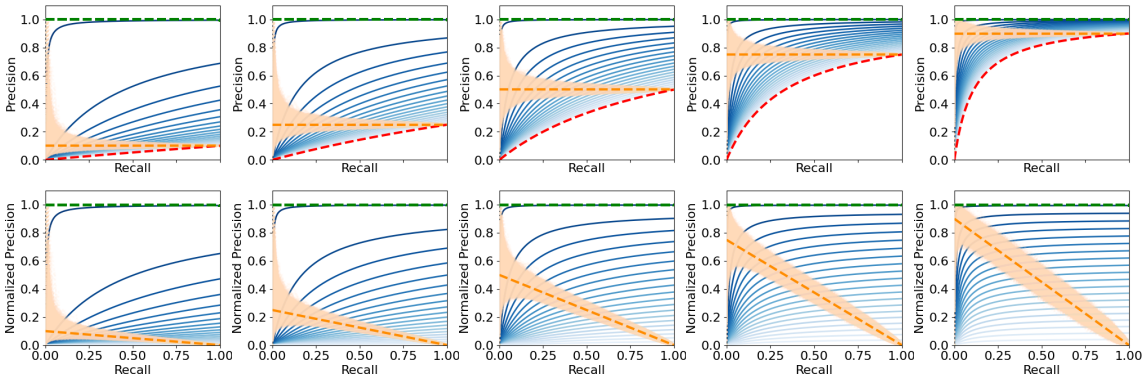
APPENDIX A
CONSTRUCTED PRECISION-RECALL CURVES

Below is shown the same figures as Fig. 1 for larger data sets. The impact of the ratio of positive and negative samples is less severe here compared to the smaller data set as discussed in the main text. Also, the visual change for a data set with a low ratio of positive to negative samples when going from a data size of $n = 20$ to $n = 200$ is evidently larger than going from $n = 200$ to $n = 2000$.



(a) $n^+ = 20, n^- = 180$ (b) $n^+ = 50, n^- = 150$ (c) $n^+ = 100, n^- = 100$ (d) $n^+ = 150, n^- = 50$ (e) $n^+ = 180, n^- = 20$

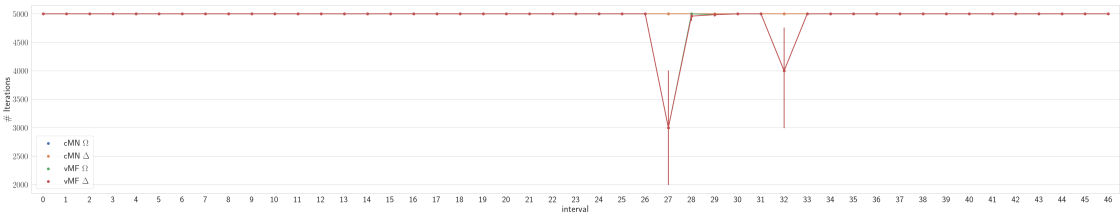
Fig. 6. Constructed PR and N-PR curves for a fixed data set size of $n = 200$ with varying ratio of positive (n^+) to negative (n^-) examples. The green dashed line is the next-to-perfect classifier, the orange dashed line is the baseline performance for a random classifier, the red dashed line is the minimum achievable precision, and the blue lines from top to bottom indicate the precisions of classifiers with an increasing amount (approximately 5%) of initial false positives before perfectly retrieving the positive samples. The light orange dotted lines is the precision curves of $10 \times n = 2000$ randomly sampled classifiers visualized with low alpha values emphasizing repeated curves to indicate their variance.



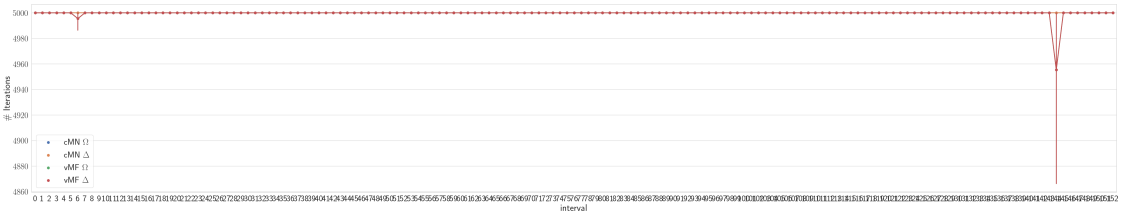
(a) $n^+ = 200, n^- = 1800$ (b) $n^+ = 500, n^- = 1500$ (c) $n^+ = 1000, n^- = 1000$ (d) $n^+ = 1500, n^- = 500$ (e) $n^+ = 1800, n^- = 200$

Fig. 7. Constructed PR and N-PR curves for a fixed data set size of $n = 2000$ with varying ratio of positive (n^+) to negative (n^-) examples. The green dashed line is the next-to-perfect classifier, the orange dashed line is the baseline performance for a random classifier, the red dashed line is the minimum achievable precision, and the blue lines from top to bottom indicate the precisions of classifiers with an increasing amount (approximately 5%) of initial false positives before perfectly retrieving the positive samples. The light orange dotted lines is the precision curves of $10 \times n = 20000$ randomly sampled classifiers visualized with low alpha values emphasizing repeated curves to indicate their variance.

APPENDIX B
NUMBER OF ITERATIONS



(a)



(b)

Fig. 8. Figure (a) and (b) show the number of iterations used by the probabilistic models estimated by the algorithm proposed in Section II-B2 for each interval of the (a) wine and (b) rice data sets respectively. The probabilistic PARAFAC2 models is specified as: vMF: von Mises-Fisher matrix distribution on $\boldsymbol{P}_k$; $c\mathcal{MN}$: constrained multivariate normal distribution on $\boldsymbol{P}_k$; Δ: homoscedastic noise model; Ω: heteroscedastic noise model.

Online Bayesian Hierarchical Clustering

Philip J. H. Jørgensen
Department of
Applied Mathematics and
Computer Science
Technical University
of Denmark
Kgs. Lyngby, Denmark

Lars Kai Hansen
Department of
Applied Mathematics and
Computer Science
Technical University
of Denmark
Kgs. Lyngby, Denmark

Tom Heskes
Institute for Computing
and Information Sciences
Radboud University
Nijmegen, The Netherlands

Jesse H. Krijthe
Department of
Intelligent Systems
Delft University of Technology
Delft, The Netherlands

Abstract

Bayesian hierarchical clustering (BHC) performs hierarchical agglomerative clustering by building a tree based on merge probabilities given by a posterior distribution. Compared to metric-based algorithms, its benefits include a predictive distribution over cluster assignments of a new data point, a general approach to deal with different types of data and a principled estimate of the number of clusters. However, existing algorithms for BHC have limited scalability and are not suited for an online setting where data arrives sequentially one data point at a time. To bring BHC into the online setting, we propose online Bayesian Hierarchical Clustering (OBHC) which updates a previously learned hierarchy using its underlying probability model. These updates have several desired traits including splitting or merging existing clusters as determined by the probabilistic model and an option to use collapsed subtrees characterized by their sufficient statistics for memory efficiency. We show that our incremental update scheme is a good approximation to the offline algorithm and compare its performance to the more efficient randomized algorithm for BHC.

1 Introduction

Clustering similar objects is an important task to make sense of data. It is used to perform dimensionality reduction, to visualize, or simply to discover some structure in the data. Offline clustering algorithms process a fully collected data set that fits in memory, while online variants are used to handle data sets larger than the available memory or to deal with a stream of data. Clustering of streams of data are of interest in many real-world applications including, for instance, finding motifs in time-series, novelty detection or concept-evolution detection (Hao et al., 2013; Hassani, 2019; Garcia et al., 2019; ZareMoodi et al., 2019).

Online clustering is a difficult task as it often involves a large amount of data. A broadly applicable online clustering model should, therefore, have the following properties:

(P1) Scale well with n (number of observations)

(P2) Scale well with k (number of clusters)

(P3) Have a bounded memory footprint

One successful approach to online clustering includes models building a hierarchy, such as BIRCH (Zhang et al., 1996) or PERCH (Kobren et al., 2017), which will be described in detail in section 4. Hierarchical clustering methods inherently include or allow for the properties (P1)-(P3) as they provide an efficient data structure for search, updates and by capturing an order of the similarity between data points, a natural choice on which data to summarize to adhere to a memory limit.

BIRCH and PERCH are both metric-based methods operating in Euclidean space with the same challenges as similar methods in the offline setting such as determining the number of clusters and evaluating new data given the model. Bayesian Hierarchical Clustering proposed by (Heller and Ghahramani, 2005b) differs from metric-based methods by formulating the decision for merging nodes using a probabilistic model. The probabilistic model makes it easy to compare models, provides a principled way of determining the number of clusters, as well as a predictive probability of new data to have been generated by the model. This work proposes to use BHC as the base model for an online algorithm to gain all the benefits of the probabilistic framework and the possibility to have the desired properties in an online setting

described above, as existing algorithms for estimating a BHC model are developed for an offline setting.

We develop an online algorithm for incrementally updating the tree by using the posterior predictive distribution to route new data into the hierarchical structure. The probabilistic model also quantifies the probability of a cluster which in turn can guide which subtrees of the hierarchical structure are candidates for collapsing in order to limit the memory footprint. These collapsed subtrees are only described by their sufficient statistics discarding the associated data and hence effectively summarizing the collapsed subtree in a single node of the tree.

For the rest of this paper, we describe BHC and its existing approximations in section 2, section 3 describes our incremental update scheme denoted *online Bayesian Hierarchical Clustering* (OBHC), section 4 covers related work on online hierarchical clustering methods and section 5 contains our experimental results on comparing the algorithms.

2 Bayesian Hierarchical Clustering

The principles of Bayesian Hierarchical Clustering are similar to traditional agglomerative clustering in that it constructs the hierarchical structure bottom-up starting with each data point assigned to individual clusters and iteratively pairs clusters based on a merging criterion. However, instead of using a metric for determining cluster pairings, BHC uses a Bayesian hypothesis test. We introduce the model formulation together with the original offline algorithm (Heller and Ghahramani, 2005b) and a randomized algorithm introduced by the same authors (Heller and Ghahramani, 2005a).

2.1 Model Specification

Following the notation of (Heller and Ghahramani, 2005b), we denote the observed data as $\mathcal{D} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}\}$ and $\mathcal{D}_i \subset \mathcal{D}$ is the data at the leaves of the subtree $\mathcal{T}_i$. Beginning with n nodes assigned a single data point, pairings are determined iteratively by greedily choosing the pair of nodes with the highest posterior probability of merging. The chosen pair is combined into a new binary tree $\mathcal{T}_k$ capturing $D_k = D_L \cup D_R$ with the left and right child of $\mathcal{T}_k$ being one of the paired nodes respectively.

The probability of observing the data $\mathcal{D}_k$ under the merging hypothesis denoted $\mathcal{H}_1^k$ consists of a specified probabilistic model $p(\mathbf{x}|\theta)$ and its prior $p(\theta|\beta)$. Its marginal distribution is

$$p(\mathcal{D}_k|\mathcal{H}_1^k) = \int \prod_{\mathbf{x}\in\mathcal{D}} p(\mathbf{x}|\theta)p(\theta|\beta)d\theta, \qquad (1)$$

and assumes that the given data $\mathcal{D}_k$ is generated identically and independently from the same distribution. This integral is tractable when a conjugate prior is chosen, which we use throughout this work for efficient computations.

The competing hypothesis denoted $\mathcal{H}_2^k$ stating that the two given subtrees D_L and D_R are individual clusters and hence should not merge is given by

$$p(\mathcal{D}_k|\mathcal{H}_2^k) = p(\mathcal{D}_L|\mathcal{T}_L)p(\mathcal{D}_R|\mathcal{T}_R), \qquad (2)$$

where the probability of $\mathcal{D}_k$ given a tree $\mathcal{T}_k$ is

$$p(\mathcal{D}_k|\mathcal{T}_k) = \pi_k p(\mathcal{D}_k|\mathcal{H}_1^k) + (1-\pi_k)p(\mathcal{D}_k|\mathcal{H}_2^k), \quad (3)$$

which can be seen as the probability of the data to be generated from a single cluster or any partition consistent with the hierarchy of the tree. The number of such partitions is exponential in n, as shown by (Heller and Ghahramani, 2005b). The prior of the merging hypothesis for the tree $\mathcal{T}_k$ denoted π_k is given as

$$\pi_k = \frac{\alpha\Gamma(n_k)}{d_k} \qquad (4)$$

with $d_k = \alpha\Gamma(n_k) + d_{L_k}d_{R_k}$, n_k as the number of data point associated with $\mathcal{T}_k$, d_{L_k} and d_{R_k} being the d_i of the left and right child of $\mathcal{T}_k$ respectively and α is a hyperparameter of the model. Each leaf i has $d_i = \alpha$ and $\pi_i = 1$.

Applying Bayes' rule with these terms we can compute the posterior probability of the merging hypothesis as

$$r_k = \frac{\pi_k p(\mathcal{D}_k|\mathcal{H}_1^k)}{\pi_k p(\mathcal{D}_k|\mathcal{H}_1^k) + (1-\pi_k)p(\mathcal{D}_k|\mathcal{H}_2^k)} \qquad (5)$$

which is used to choose pairings of trees while building the hierarchy bottom-up and later to determine whether two subtrees are considered merged or not. A natural choice for this decision is to cut the final tree where $r_k < 0.5$ resulting in a partition of the data.

2.2 The Offline Algorithm

The offline algorithm for building the hierarchy is bottom-up and greedy. Given a data set $\mathcal{D}$ and a probabilistic model $p(\mathbf{x}|\theta)$ and a prior $p(\theta|\beta)$ it computes (5) for all pairs of nodes and merges the pair with the largest value. After each merge, the algorithm updates the set

of nodes considered for the next iteration by removing the two nodes that have merged into a new binary tree $\mathcal{T}_k$ and adds the root node of the new subtree to be considered for futures merges. The algorithm terminates when all nodes have been merged into a single binary tree. To build the whole tree requires computing $O(n^2)$ potential merges and comparing them. The result of the algorithm is a binary tree where each node can be seen as a mixture component in a Bayesian mixture model. Evaluating the marginal likelihood in (3) scales with complexity $O(n \log n)$.

2.3 Posterior Predictive Distribution

The resulting Bayesian mixture model defined by the tree has a corresponding posterior predictive distribution for new data. The distribution of a new observation denoted $\mathbf{x}_{t+1}$ is given by

$$p(\mathbf{x}_{t+1}|\mathcal{D}) = \sum_{k \in \mathcal{N}} \omega_k p(\mathbf{x}_{t+1}|\mathcal{D}_k), \tag{6}$$

where the posterior probability of node k is $\omega_k = r_k \prod_{i \in \mathcal{P}_k}(1-r_i)$, $\mathcal{N}$ is the set of all nodes in the tree and $\mathcal{P}_k$ is the set of nodes along the path from the parent of node k to the root node. Each node has the predictive distribution $p(\mathbf{x}_{t+1}|\mathcal{D}_k) = \int p(\mathbf{x}_{t+1}|\theta)p(\theta|\mathcal{D}_k, \beta)d\theta$. The computational complexity for computing the posterior predictive distribution is $O(n)$.

2.4 Approximate Algorithms

The construction of the tree becomes prohibitively expensive for large data sets. To scale the BHC model to such cases (Heller and Ghahramani, 2005a) proposed two randomized versions of the algorithm. One is based solely on random subsampling with a computational complexity of $O(n \log n)$ and another combines random subsampling with an expectation-maximization (EM) step fitting a set of clusters which the offline BHC algorithm is then fitted to, resulting in a computational complexity of $O(n)$ for a small number of clusters and with a low number of subsamples drawn per step. In a similar fashion to the EM approach, (Xu et al., 2009) proposed to first use a Bayes K-means to fit the data followed by using BHC on the determined clusters. For our experiments, we focus on the first version following the work of (Darkins et al., 2013).

The randomized algorithm randomly subsamples m data points from $\mathcal{D}$ and fits a BHC model on them using the offline algorithm. The remaining $n - m$ points are then filtered by assigning them to either the left or right subtree of the root in the fitted BHC tree, based on the posterior merging probabilities similar to the offline algorithm. This splits the data points into two sets, and the randomized algorithm is then used on both sets. This procedure is then applied recursively until the number of points in a set is lower than m, then the recursion ends and a final BHC model is fitted to those points becoming the subtree containing the data points. A more formal description is given by (Heller and Ghahramani, 2005a) and (Darkins et al., 2013).

All of these algorithms depend on either clustering the whole data set before using BHC or being able to subsample from it, thus unsuitable to be used for an online setting. The randomized algorithm could perhaps be applied by performing batch updates, but would be highly dependent on the ordering of the data.

3 Online Bayesian Hierarchical Clustering

We propose an incremental algorithm for BHC in an online setting where we assume that data is generated by a stream and arriving one data point at a time. The incremental updates are performed in the following manner: The arrival of the first data point $\mathbf{x}_1$ results in a tree $\mathcal{T}_1$ consisting of a single node containing $\mathbf{x}_1$. The second data point $\mathbf{x}_2$ is simply merged with $\mathbf{x}_1$ resulting in $\mathcal{T}_2$ with the nodes of $\mathbf{x}_1$ and $\mathbf{x}_2$ as the children of its root. At this stage, the update becomes non-trivial with the arrival of any additional data point $\mathbf{x}_{t+1}$ as it has to be inserted into the tree $\mathcal{T}_t$. To do so, the online algorithm performs the following steps: 1) estimate the leaf $\ell^\star$ of $\mathcal{T}_t$ with the maximum posterior predictive probability of $\mathbf{x}_{t+1}$; 2) split the tree $\mathcal{T}_t$ along the path from the root of $\mathcal{T}_t$ to $\ell^\star$; 3) build a new tree $\mathcal{T}_{t+1}$ from the output of the split of $\mathcal{T}_t$ and the new data point $\mathbf{x}_{t+1}$ using the offline BHC algorithm. These steps are repeated using $\mathcal{T}_t$ for any newly arrived data point $\mathbf{x}_{t+1}$ resulting in a new tree $\mathcal{T}_{t+1}$ computed by the BHC model. Each step is described in detail below.

Estimate the leaf $\ell^\star$ of $\mathcal{T}_t$ with the maximum posterior predictive probability of the new data point $\mathbf{x}_{t+1}$ by performing a search in the tree. Starting from the root of $\mathcal{T}_t$, select its child with the maximum posterior probability of $\mathbf{x}_{t+1}$ given the subtree of the child. Repeat this selection for the children of the selected child recursively until a leaf is encountered. We refer to this path from the root of $\mathcal{T}_t$ to the leaf $\ell^\star$ as $\mathcal{P}^\star$.

The posterior predictive probability of $\mathbf{x}_{t+1}$ given a subtree $\mathcal{T}_k$ is:

$$p(\mathbf{x}_{t+1}|\mathcal{T}_k) = \sum_{k \in \mathcal{N}_k} \omega_k p(\mathbf{x}_{t+1}|\mathcal{D}_k), \tag{7}$$

where $\mathcal{N}_k$ is the set of all nodes in the subtree $\mathcal{T}_k$, and ω_k is the same as in (6).

Split the tree $\mathcal{T}_t$ by removing the nodes along the path $\mathcal{P}^\star$ found by the previous step resulting in a set of trees. This set is a forest $\mathcal{F}$ of the subtrees having the direct descendants of $\mathcal{P}^\star$ as their roots.

Rebuild the tree by fitting a BHC model on the union of the forest $\mathcal{F}$ and $\mathbf{x}_{t+1}$ using the offline algorithm.

These steps result in an updated tree $\mathcal{T}_{t+1}$ incorporating the new data point. The updated tree contains all the subtrees of the nodes returned by splitting $\mathcal{T}_t$, effectively not changing them, and the new nodes created from merging the roots of these subtrees or $\mathbf{x}_{t+1}$. This means $\mathcal{T}_t$ is only updated locally. Where the update takes place is determined by which part of the tree best describes $\mathbf{x}_{t+1}$. By routing the $\mathbf{x}_{t+1}$ to a leaf in this way instead of routing it to the node with overall largest posterior predictive probability, the full structure of the most important subtree is forced to be rebuild, which allows for the merge probabilities to be updated at all levels of the tree.

Furthermore, to limit the memory footprint of the model we propose to collapse subtrees of high certainty, as given by their merge probabilities in (5), by adding a post-processing step to the first three steps:

Collapse the subtree(s) determined by either (C1) the subtree with the maximum r_k when the number of leaves in $\mathcal{T}_{t+1}$ reaches a hard limit ℓ_{max}, or (C2) when any of the subtrees has a r_k larger than a threshold $T_{collapse}$.

A collapsed subtree is only described by the sufficient statistics of the data points previously assigned to one of its leaves. These leaves can themselves be already collapsed subtrees as the sufficient statistics are additive. We discard the original data point as all future computations depends only on the sufficient statistics, while at the same time the hierarchy of the collapsed subtrees can not be updated anymore.

The computational complexity of the online algorithm is quadratic in the number of leaves similar to the offline algorithm, but only in the worst case. Notice, the number of subtrees n_s to fit the offline algorithm on depends on the depth of the tree so if the tree $\mathcal{T}_t$ is fairly balanced, i.e. the tree is shallow, the number of leaves will be $n \gg n_s$ for large n for large n. In the worst case, it becomes $O(|leaves(\mathcal{T}_{t+1})|^2)$ with $|leaves(\mathcal{T}_{t+1})|$ being the number of leaves in $\mathcal{T}_{t+1}$, as we might have to rebuild the whole tree. However, empirically we find this is rarely the case when the probabilistic model describes the data well. Also, using method (C1) for collapsing limits this worst-case to be $O(\ell_{max}^2)$ for any data set, while method (C2) have the potential to dramatically decrease the number of leaves if many of the subtrees have a large posterior probability of merging, where the optimal number is the most suitable number of clusters $m^\star$ for the data resulting in the updates taking $O(m^{\star 2})$.

4 Related Work

Despite being particularly amenable to the online setting, few hierarchical clustering methods have been developed for it specifically. One of the first and most cited online hierarchical clustering algorithms is the Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH) algorithm (Zhang et al., 1996). BIRCH uses *clustering feature trees* to keep track of the characteristics of the nodes in the tree. These clustering features are equivalent to our summarization using the sufficient statistics in the case of a Gaussian model. BIRCH computes the similarities in Euclidean space using 5 different notions of distance. Additionally, BIRCH uses multiple phases, where some of the phases include additional passes over the data to recover from mistakes, but it can also be run for the one pass setting we consider here.

The *Purity Enhancing Rotations for Cluster Hierarchies* (PERCH) algorithm (Kobren et al., 2017) is one of the most recent methods to construct a cluster hierarchy incrementally. Similar to BIRCH, PERCH operates on d-dimensional data in Euclidean space and uses a rotation procedure inspired by self-balancing binary search trees to both correct for clustering decisions decreasing the dendrogram purity after greedily routing a new data point to a leaf using a nearest-neighbor search, and to balance the tree without compromising the dendrogram purity of the clustering. The balancing of the tree increases the efficiency of the incremental updates as they are heavily dependent on the depth of the tree. In the worst case the run time of PERCH scales with $O(n^2)$ as it would have to explore all leaves of the tree, but this is not commonly encountered due to the balancing rotations as suggested by their results. PERCH also uses a bounding box approximation to increase the efficiency of the nearest neighbor searches, which is another approach for summarizing a subtree like the clustering features of BIRCH, or the sufficient statistics used in this work. PERCH makes no assumption on the ordering of the data.

Another recent idea for two incremental algorithms for building a cluster hierarchy was proposed by (Menon et al., 2019). Their first proposal is denoted *online top down clustering* (OTD) and starts at the root node of $\mathcal{T}_1$ and compares the average similarity between its leaves to their average similarity to the leaves of another given subtree $\mathcal{T}_2$ (in this case a single new data point). If the average similarity of the leaves of $\mathcal{T}_1$ is larger than the average similarity across the leaves of both subtrees then the $\mathcal{T}_1$ and $\mathcal{T}_2$ become siblings of a new root node as their parent resulting in the new tree $\mathcal{T}_3$. Otherwise, the

Table 1: Summary of the online hierarchical clustering methods and their properties: (P1) scale well with n, (P2) scale well with k, (P3) a bounded memory footprint

	P1	P2	P3
BIRCH	√	√	√
PERCH	√	√	√
OTD	√	×	×
OHAC	×	×	×
OBHC	√	√	√

same comparison is made between $\mathcal{T}_2$ and the child of $\mathcal{T}_1$ with the largest average similarity to $\mathcal{T}_2$. This procedure is recursively performed until $\mathcal{T}_3$ is created or a leaf node is encountered. OTD scales with $O(d)$ with d being the depth of the tree and possibly $d = n$ in the worst case. Their second algorithm denoted *online HAC* (OHAC) takes a different approach more similar to the approach of OBHC. They break a given tree $\mathcal{T}_k$ into a forest of subtrees similar to split step of OBHC but from the nearest-neighbor leaf, which is computed over all leaves instead of searching for the most similar leaf using the tree. This results in a complexity of $O(n)$ given an admissible linkage function. An admissible linkage function is defined as a linkage function which computes the set $\{L(C_i, C_j)\}_{i \neq j}$ in time $O(m^2 + \sum_{i=1}^{m} |C_i|)$ with m being the number of clusters.

5 Results

We compare OBHC to the offline and randomized algorithm for BHC applied to both synthetic and real data. We employ a probabilistic model using a multivariate normal likelihood with the conjugate prior following a normal-inverse-Wishart distribution (Murphy, 2007) defined by

$$\text{NIW}(\mu, \Sigma \mid \mu_0, \kappa, \Psi, \nu) = \mathcal{N}(\mu \mid \mu_0, \frac{1}{\kappa}\Sigma)\mathcal{W}^{-1}(\Sigma \mid \Psi, \nu) \tag{8}$$

5.1 Data

We perform experiments on the following data sets.

Synthetic GMM data generated from a mixture of Gaussians by drawing a categorical distribution from a symmetric Dirichlet distribution with hyperparameter $\alpha = \mathbf{1}_m$, where m is the number of components. Each component is drawn from a 2-dimensional normal-inverse-Wishart distribution $\text{NIW}(\mu_0, \kappa, \Psi, \nu)$ with the following hyperparameters: mean μ equal to $(0, 0)$, diagonal scale matrix Ψ equal to 50 along the diagonal, degrees of freedom ν equal to 10 and κ equal to 0.005. This leads to a high variance on the means of the components, giving a high probability for well-separated clusters. We generate data sets from this underlying distribution each with 200 samples for m equal to $[1, 2, .., 5, 10, 15, .., 50]$. We generate 10 realizations of each configuration leading to a total of 140 data sets.

Gene expression data analyzed using the randomized algorithm by (Darkins et al., 2013). This data is constructed from several realizations drawn from a BHC model based on Gaussian processes fitted to a 169-gene subset of the cell cycle gene expression data of (Cho et al., 1998), resulting in 999 genes spread across 13 distinct clusters. However, the labels provided separate the data into 77 subclusters, which each belong to one of the 13 superclusters. The superclusters can be verified by visualizing the data using t-SNE (Maaten and Hinton, 2008).

5.2 Evaluation

As many of the related works, we use the *adjusted Rand index* (ARI) to evaluate flat clusterings and the *dendrogram purity* (DP) introduced by (Heller and Ghahramani, 2005b) to evaluate the hierarchical clusterings. The DP is defined with the respect to a true clustering Z^*, and written as

$$\frac{1}{|Z^*_{pairs}|} \sum_{m=1}^{M} \sum_{\mathbf{x}_i, \mathbf{x}_j \in Z^*_k} \frac{|leaves(LCA(\mathbf{x}_i, \mathbf{x}_j)) \cap Z^*_k|}{|leaves(LCA(\mathbf{x}_i, \mathbf{x}_j))|}, \tag{9}$$

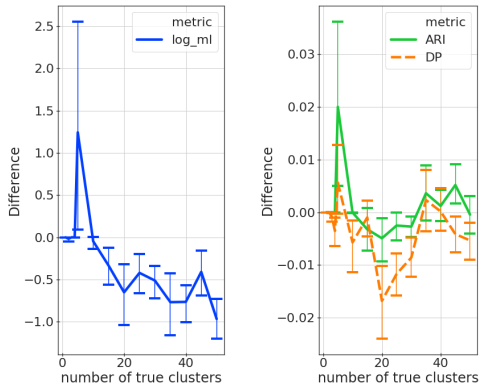
with $Z^*_{pairs} = \{(x_i, x_j) \mid Z^*(x_i) = Z^*(x_j)\}$ being the set of pairs of points belonging to the same cluster in the true clustering and $LCA(x_i, x_j)$ being the least common ancestor of x_i and x_j in $\mathcal{T}_k$.

Besides these quantities, we consider the run times and number of clusters when cutting the trees, as described in section 2.1, as important indicators for the performance of the models.

5.3 Experiments

For the GMM data, we run both the offline and online BHC algorithms with the same hyperparameters as the data were generated by (see section 5.1). The online algorithm was computed without collapsing the tree in order to compare it to the offline algorithm without limitation. Since the offline algorithm is deterministic for each data set, we simply ran it once for this and the other data sets. The online algorithm was run for 10 different

orderings of each of the GMM data sets, as the order of the data points influences the computation of the model. The objective with these data is to evaluate the difference between the offline and online algorithms and to do so, we compute the difference between the ARI, DP and the logarithm of the marginal likelihood given by (3) for the resulting models. The scores of the offline algorithm are subtracted from the scores of the online algorithm. This means that a value larger than 0 suggests a better online model than the offline, and vice-versa. The result is shown in Figure 1 with error bars showing 95% of the variance over the different data set realizations and orderings. The mean difference in ARI and DP between the models is less than 0.02, while the log marginal likelihood keeps within a mean difference of 1.5.



(a) (b)

Figure 1: Difference between the offline and online BHC algorithms as measured by the log marginal likelihood in (a), and the scores ARI and DP in (b) on the synthetic GMM data.

For the gene expression data, we have run the online algorithm using method (C1) and (C2) for collapsing the trees. We also fit the offline and randomized BHC algorithms for comparison. Both the online variants and the randomized were repeated over 10 random orderings for all settings. The models are dependent on the settings of the hyperparameters, and the online variants together with the randomized algorithm are dependent on their parameters ℓ_{max}, $T_{collapse}$ and m respectively. Before evaluating the influence of the parameters controlling the approximations, they are set to fairly high values, $\ell_{max} = 200$, $T_{collapse} = 0.99$ and $m = 300$, and the hyperparameters are varied to determine the sensitivity of the models. Each model is run with the combination of all the following settings of the hyperparameters; $\alpha \in [0.5, 1, 5, 10]$, $F_\psi \in [0.1, 0.5, 1, 2]$, $\nu \in [-1, 1, 5, 10, 20]$ and $\kappa \in [0.1, 1]$. The mean μ_0 is estimated from the data. The diagonal of Ψ is also estimated to be some factor of the diagonal of the covariance matrix of the data controlled by the parameter F_ψ. The DP and ARI of the models are visualized as a function of the log marginal likelihood in Figure 2. A clear positive correlation between the log marginal likelihood and the ARI and DP of the offline and online algorithms. The ARI and DP of the random algorithm show a positive correlation with log marginal likelihood as well, but it is less convincing than the rest. Some settings resulted in that the models would degenerate while taking longer than reasonable to fit, and thus not included in these results

To investigate the influence of the parameters ℓ_{max}, $T_{collapse}$ and m, we use one of the best settings of the hyperparameters across the models in terms of log marginal likelihood among the ranges mentioned above. These values were $(\alpha, F_\psi, \kappa, \nu) = (1, 2, 0.1, 10)$. Each algorithm was varied in its model complexity by choosing their respective parameters in a suitable range. The parameters m and ℓ_{max} varied over the values 2, 50, 100, 150, 200, 250, 300, 400, 500, 600, 700, 800, 900 and 999, while $T_{collapse}$ was set as 12 values in the interval $[0.8, 1]$ with more values closer to 1. The results compared to the offline algorithm is illustrated in Figure 3.

We see both methods for collapsing in the online algorithm approximates the scores of the offline algorithm. Especially, method (C1) as it is being more conservative on when to collapse a subtree. But even for low values of ℓ_{max} it manages to approximate the performance of the offline algorithm. Method (C2) is very sensitive to the setting of the parameter $T_{collapse}$ on this data, but it manages to achieve comparable performance to that of the random algorithm for a fraction of the run time. The random algorithm slowly reaches the same level as the offline algorithm as it increases the number of data points used. However, the run time gains are lost as the random algorithm approximates the full algorithm. The online algorithms have a better performance at the lower model complexities with shorter run times than the random algorithm. Also, the random algorithm heavily overestimates the number of clusters for smaller m, and continues until the full data set is considered, whereas the online algorithms approximately estimate the same number of clusters as the offline version with almost an exact match for collapse method (C1), and a slight underestimation for collapse method (C2).

6 Discussion

We proposed an online algorithm for the BHC model taking full advantage of the probabilistic properties of the

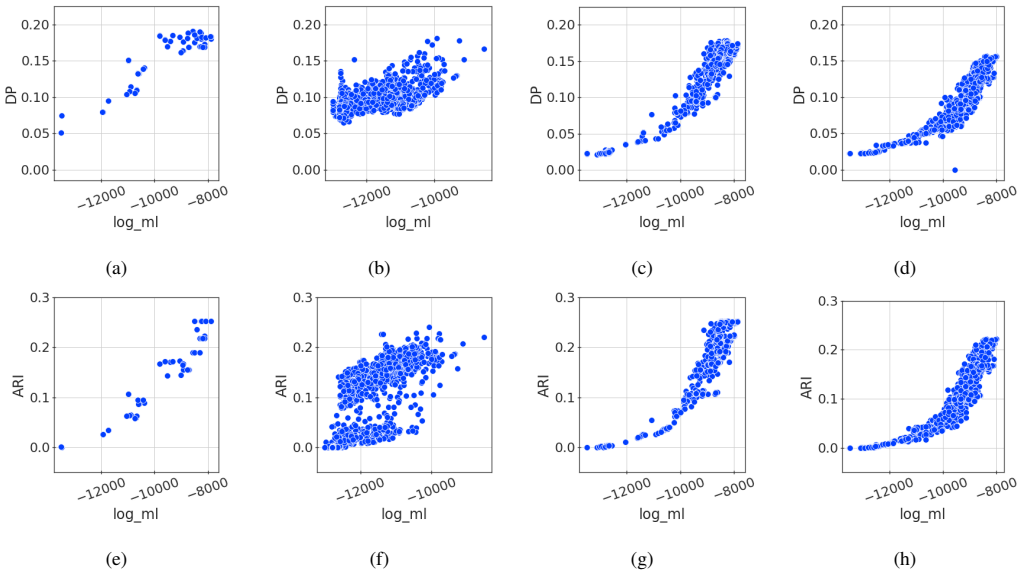


Figure 2: Resulting scores of DP and ARI for the offline algorithm (a,e), the randomized algorithm (b,f), the online algorithm using (C1) (c,g) and the online algorithm using (C2) (d,h) for varying settings of hyperparameters.

model to efficiently update the clustering tree while approximating the result given by the offline algorithm. In experiments, we see that this online algorithm can construct a model that is similar in performance to the offline variant. The online algorithm shows robustness to different orderings of the data, as we run many experiments with random orderings while seeing a low variance in the performance.

In setting up our experiments, we found that finding the right settings for the hyperparameters of these models can be a difficult task. A nice property that helps in this regard is that, like for the offline algorithm, for the online version the marginal likelihood seems to be highly correlated with the performance measures considered here. This connection is less pronounced for the randomized variant. An open problem is how to efficiently optimize this marginal likelihood of the model in the online setting, something that is possible in the offline algorithm.

Even in offline settings, where a randomized algorithm can be used to speed up computations, we find the online algorithm offers an effective alternative to the randomized algorithm to apply the BHC model to large data sets. It shows different properties to those of the randomized algorithm by approximating the performance of the offline algorithm while mostly using less run time. The randomized algorithm also seems to overestimate the number of clusters given by cutting the tree, which the online algorithm does not seem to suffer from. However, the randomized algorithm more consistently approximates the offline algorithm when given more resources in terms of the number of points subsampled. As this number becomes larger, however, the benefit of faster run times of the randomized algorithm is decreased.

To further speed up computations, the online algorithm could be extended to be computed in a distributed manner by updating separate subtrees. This could be done similarly to how concurrent updates in binary search trees are performed. It would very interesting to compare to existing online hierarchical clustering algorithms on large data set after additional efforts to speed up the implementation. There exist many interesting future directions for building on this work, which we hope to explore.

References

Cho, R. J., Campbell, M. J., Winzeler, E. A., Steinmetz, L., Conway, A., Wodicka, L., Wolfsberg, T. G., Gabrielian, A. E., Landsman, D., Lockhart, D. J., and Davis, R. W. (1998). A genome-wide transcriptional analysis of the mitotic cell cycle. *Mol. Cell*, 2(1):65–73.

Darkins, R., Cooke, E. J., Ghahramani, Z., Kirk, P. D. W., Wild, D. L., and Savage, R. S. (2013). Accelerating bayesian hierarchical clustering of time series data with a randomised algorithm. *PLoS One*, 8(4):e59795.

Garcia, K. D., de Faria, E. R., de Sá, C. R., Mendes-Moreira, J., Aggarwal, C. C., de Carvalho, A. C. P. L. F., and Kok, J. N. (2019). Ensemble clustering for novelty detection in data streams. In *Discovery Science*, pages 460–470. Springer International Publishing.

Hao, Y., Chen, Y., Zakaria, J., Hu, B., Rakthanmanon, T., and Keogh, E. (2013). Towards never-ending learning from time series streams. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 874–882, New York, NY, USA. ACM.

Hassani, M. (2019). Overview of efficient clustering methods for High-Dimensional big data streams. In Nasraoui, O. and Ben N'Cir, C.-E., editors, *Clustering Methods for Big Data Analytics: Techniques, Toolboxes and Applications*, pages 25–42. Springer International Publishing, Cham.

Heller, K. and Ghahramani, Z. (2005a). Randomized algorithms for fast bayesian hierarchical clustering. In *PASCAL Workshop on Statistics and Optimization of Clustering*, volume 25, pages 1–22.

Heller, K. A. and Ghahramani, Z. (2005b). Bayesian hierarchical clustering. In *Proceedings of the 22Nd International Conference on Machine Learning*, ICML '05, pages 297–304, New York, NY, USA. ACM.

Kobren, A., Monath, N., Krishnamurthy, A., and McCallum, A. (2017). A hierarchical algorithm for extreme clustering. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, pages 255–264, New York, NY, USA. ACM.

Maaten, L. v. d. and Hinton, G. (2008). Visualizing data using t-SNE. *J. Mach. Learn. Res.*, 9(Nov):2579–2605.

Menon, A. K., Rajagopalan, A., Sumengen, B., Citovsky, G., Cao, Q., and Kumar, S. (2019). Online hierarchical clustering approximations.

Murphy, K. P. (2007). Conjugate bayesian analysis of the gaussian distribution. *DEF*, 1($2\sigma 2$):16.

Xu, Y., Heller, K., and Ghahramani, Z. (2009). Tree-Based inference for dirichlet process mixtures. 5:623–630.

ZareMoodi, P., Siahroudi, S. K., and Beigy, H. (2019). Concept-evolution detection in non-stationary data streams: a fuzzy clustering approach. *Knowl. Inf. Syst.*, 60(3):1329–1352.

Zhang, T., Ramakrishnan, R., and Livny, M. (1996). BIRCH: an efficient data clustering method for very large databases. *ACM Sigmod Record*.

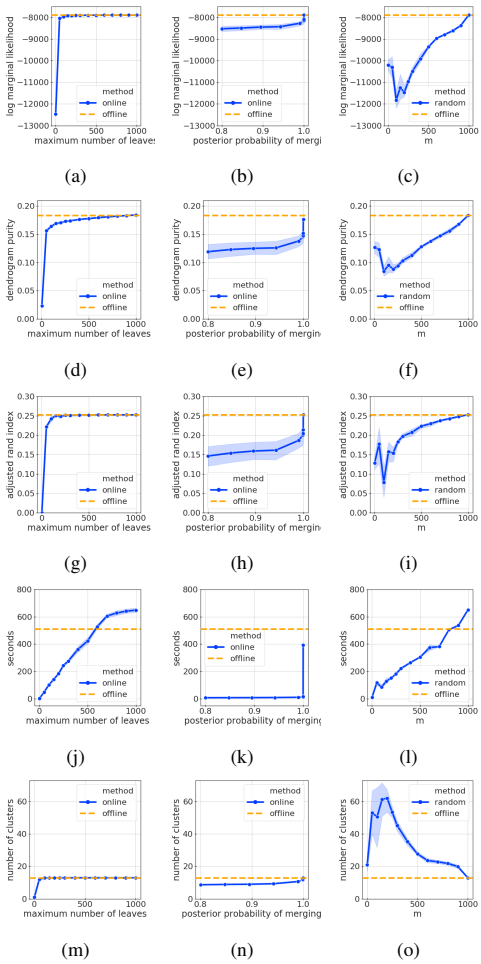


Figure 3: Comparison of the online algorithms to the randomized and offline algorithm showing from top to bottom: log marginal likelihood, DP, ARI, run time in seconds and number of clusters as suggested by the BHC model.