RICE UNIVERSITY

Bayesian Methods for Learning Analytics with Applications

by

Andrew E. Waters

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE

Doctor of Philosophy

APPROVED, THESIS COMMITTEE:

Dr. Richard G. Baraniuk, Chair
Department of Electrical & Computer Engineering, Rice University

Dr. Caleb Kemere
Department of Electrical & Computer Engineering, Rice University

Dr. Marina Vannucci
Department of Statistics, Rice University

HOUSTON, TEXAS
DECEMBER, 2013
Abstract

Bayesian Methods for Learning Analytics with Applications

by

Andrew E. Waters

Learning Analytics (LA) is a broad umbrella term used to describe statistical models and algorithms for understanding the relationship between a set of learners and a set of questions. The end goal of LA is to understand the dynamics of the responses provided by each learner. LA models serve to answer important questions concerning learners and questions, such as which educational concepts a learner understands well, which ones they do not, and how these concepts relate to the individual question. LA models additionally predict future learning outcomes based on learner performance to date. This information can then be used to adapt learning to achieve specific educational goals.

In this thesis, we adopt a fully Bayesian approach to LA, which allows us both to have superior flexibility in modeling as well as achieve superior performance over methods based on convex optimization. We first develop novel models and algorithms for LA. We showcase the performance of these methods on both synthetic as well as real-world educational datasets.

Second, we apply our LA framework to the problem of collaboration-type detection in educational data sets. Collaboration amongst learners in educational settings is problematic for two reasons. First, such collaboration may be prohibited and considered a form of cheating. Detecting
this form of collaboration is essential for maintaining fairness and academic integrity in a course. Finally, collaboration inhibits the ability of LA methods to accurately model learners. We develop several novel techniques for collaboration-type detection where we not only identify collaboration in a statistically principled way, but also classify the type of collaborative behavior.
ACKNOWLEDGEMENTS

First and foremost my thanks go out to my advisor, Dr. Richard Baraniuk, for his helpful guidance over the past several years at Rice. His infectious enthusiasm and far-reaching vision has made me a better researcher, writer, presenter, and person.

Many thanks also to my committee for their help in making this thesis possible. I first met Marina Vannucci when she sat on my ELEC599 committee and then in her series of courses on Bayesian methods. She has been a fantastic collaborator and mentor over the past several years. Caleb Kemere and I first met when he interviewed at Rice. His broad depth of knowledge in all things related to statistical signal processing have been a tremendous aid both in refining the work in this thesis as well as for realizing connections to related work.

This thesis would not be possible without the many collaborators with whom I have worked over the past several years. My endless thanks go out to Christoph Studer, Aswin Sankaranarayanan, Mr. Lan, Michele Guindani, Kassie Fronczyk, Volkan Cevher, and Charles Sestok IV for all their efforts.

I have had the privilege of interacting with many wonderful people throughout graduate school who have made the journey so much more enjoyable. To this end, many thanks to Lisa, Eva, Ivy, Tom, Div, Ryan N., Hannah, Leticia, Ryan G., Corina, Matthew, Eric, Sam, Marcel, Achal, Liz, Lee, Denise, Mark, Chin, Stephen, Jason, and Marco.
Last, my unending gratitude for my family who have been so supportive through the many ups and downs that graduate school has brought. In particular my mother Gay and grandmother Bonnie who have influenced me more than they will ever realize. Finally to my little Hannah and Janelle for being a constant source of inspiration and happiness.
## CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>ii</td>
</tr>
<tr>
<td>Acknowledgements</td>
<td>iv</td>
</tr>
<tr>
<td><strong>1 Introduction</strong></td>
<td>1</td>
</tr>
<tr>
<td>1.1 Overview</td>
<td>1</td>
</tr>
<tr>
<td>1.2 A Note on Notation</td>
<td>3</td>
</tr>
<tr>
<td>1.3 The Learning Analytics Challenge</td>
<td>4</td>
</tr>
<tr>
<td>1.4 The SPARFA model for LA</td>
<td>7</td>
</tr>
<tr>
<td>1.5 Outline</td>
<td>7</td>
</tr>
<tr>
<td><strong>2 SPARFA: SPARse Factor Analysis for Learning Analytics</strong></td>
<td>8</td>
</tr>
<tr>
<td>2.1 Introduction</td>
<td>9</td>
</tr>
<tr>
<td>2.2 Statistical Model for Learning and Content Analytics</td>
<td>10</td>
</tr>
<tr>
<td>2.3 SPARFA-B: Bayesian Sparse Factor Analysis</td>
<td>19</td>
</tr>
<tr>
<td>2.4 Tag Analysis: Post-Processing to Interpret the Estimated Concepts</td>
<td>25</td>
</tr>
<tr>
<td>2.5 Experiments</td>
<td>28</td>
</tr>
<tr>
<td>2.6 Conclusions</td>
<td>48</td>
</tr>
<tr>
<td><strong>3 Extensions of SPARFA</strong></td>
<td>50</td>
</tr>
<tr>
<td>3.1 Introduction</td>
<td>51</td>
</tr>
<tr>
<td>3.2 Bayesian Model</td>
<td>53</td>
</tr>
<tr>
<td>3.3 Posterior Inference</td>
<td>58</td>
</tr>
<tr>
<td>3.4 Applications</td>
<td>61</td>
</tr>
<tr>
<td>3.5 Conclusions</td>
<td>74</td>
</tr>
</tbody>
</table>
## 4 Learning Analytics Model with Automated Learner Clustering

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1 Introduction</td>
<td>76</td>
</tr>
<tr>
<td>4.2 Model</td>
<td>78</td>
</tr>
<tr>
<td>4.3 Algorithm</td>
<td>80</td>
</tr>
<tr>
<td>4.4 Experiments</td>
<td>83</td>
</tr>
<tr>
<td>4.5 Conclusions</td>
<td>87</td>
</tr>
</tbody>
</table>

## 5 Collaboration-Type Identification

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1 Introduction</td>
<td>88</td>
</tr>
<tr>
<td>5.2 Statistical approaches for Learning Analytics</td>
<td>93</td>
</tr>
<tr>
<td>5.3 Statistical models for real-world collaborative behavior</td>
<td>95</td>
</tr>
<tr>
<td>5.4 Algorithms for Collaboration-Type Identification</td>
<td>100</td>
</tr>
<tr>
<td>5.5 Experiments</td>
<td>107</td>
</tr>
<tr>
<td>5.6 Conclusions</td>
<td>115</td>
</tr>
</tbody>
</table>

## Appendices

<table>
<thead>
<tr>
<th>Appendix</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A Proof of Theorem 1</td>
<td>121</td>
</tr>
<tr>
<td>B Implementation details for non-parametric SPARFA</td>
<td>124</td>
</tr>
<tr>
<td>C MCMC steps for Bayesian Rasch approach to Learning Analytics</td>
<td>128</td>
</tr>
<tr>
<td>D Numerical evaluation of (5.1)</td>
<td>130</td>
</tr>
</tbody>
</table>

References | 133 |
List of Figures

2.1 Graphical Depiction of the SPARFA Framework ........................ 11
2.2 SPARFA decomposition of STEMscopes data with tag post-processing 13
2.3 SPARFA synthetic experiments: variation in problem size ............. 33
2.4 SPARFA synthetic experiments: variation in percentage of observed entries ........................................ 34
2.5 SPARFA synthetic experiments: variation in sparsity level .......... 35
2.6 SPARFA synthetic experiments: variation in link function .......... 36
2.7 SPARFA decomposition on the Signal Processing dataset with tag post-processing ........................................ 37
2.8 Knowledge estimates for STEMscopes dataset ....................... 40
2.9 SPARFA decomposition on Mechanical Turk algebra exam with tag post-processing ........................................ 44

3.1 Non-parametric SPARFA experiments: variations in percentage of observed entries ........................................ 64
3.2 Mechanical Turk: posterior mean of $K$ and $\psi_j$ ................. 67
3.3 Mechanical Turk: posterior mean of $H$ and $W$ ..................... 67
3.4 Mechanical Turk: bipartite decomposition ............................. 68
3.5 STEMscopes: posterior mean of $K$ and $\psi_j$ ...................... 70
3.6 STEMscopes: bipartite decomposition ............................... 71
3.7 Signal Processing course: posterior mean of $K$ and $\psi_j$ ....... 72
3.8 Signal Processing course: bipartite decomposition .................. 73

5.1 Block diagram for collaboration–type identification .................. 91
5.2 Graphical model for collaborative model selection (CMS) . . . . . . . 104
5.3 Collaboration identification: synthetic data comparisons . . . . . . 110
5.4 Collaboration–type identification: synthetic data experiments for CMS 111
5.5 Collaboration–type identification: ELEC301 homework groups . . 117
5.6 Collaboration–type identification: ELEC301 final exam . . . . . . . 118
5.7 Collaboration–type identification: ELEC220 final exam . . . . . . . 119
## List of Tables

2.1 Signal processing: tag knowledge of Learner 1. 38
2.2 Signal processing: tag knowledge of all learners 38
2.3 STEMscopes: comparison of SPARFA methods over questions 42
2.4 Mechanical Turk: learner responses and concept decomposition 45
2.5 Mechanical Turk: estimated concept mastery 46
2.6 SPARFA synthetic experiments: imputation accuracy for various methods 47

3.1 Non-parametric SPARFA synthetic experiments: performance comparison with parametric SPARFA 62
3.2 Non-parametric SPARFA experiments: variation in size for heteroskedastic ordinal data 62
3.3 Non-parametric SPARFA synthetic experiments: variation in percentage of observed entries 63
3.4 Non-parametric SPARFA synthetic experiments: variation in learner precision 65
3.5 Non-parametric SPARFA real-world experiments: STEMscopes imputation accuracy comparison with parametric SPARFA 70

4.1 Clustered SPARFA synthetic experiments: average classification error 84
4.2 Clustered SPARFA synthetic experiments: $E_\theta$ vs $L$ 84
4.3 Clustered SPARFA real-world experiments: comparison with non-clustered model 85
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.4</td>
<td>Clustered SPARFA real-world experiments: recovered $L$ for various</td>
<td>86</td>
</tr>
<tr>
<td></td>
<td>datasets</td>
<td></td>
</tr>
<tr>
<td>5.1</td>
<td>Likelihood table for the independence model $\mathcal{M}_1$</td>
<td>97</td>
</tr>
<tr>
<td>5.2</td>
<td>Likelihood table for the parasitic copy model $\mathcal{M}_2$</td>
<td>97</td>
</tr>
<tr>
<td>5.3</td>
<td>Likelihood table for the dominance model $\mathcal{M}_3$</td>
<td>97</td>
</tr>
<tr>
<td>5.4</td>
<td>Likelihood table for the collaborative OR model $\mathcal{M}_4$</td>
<td>97</td>
</tr>
</tbody>
</table>
1.1 Overview

Textbooks, lectures, and homework assignments were the answer to the main educational challenges of the 19th century, but they are the main bottleneck of the 21st century. Today’s textbooks are static, linearly organized, time-consuming to develop, soon out-of-date, and expensive. Lectures remain a primarily passive experience of copying down what an instructor says and writes on a board (or projects on a screen). Homework assignments that are not graded for weeks provide poor feedback to learners (e.g., students) on their learning progress. Even more importantly, today’s courses provide only a “one-size-fits-all” learning experience that does not cater to the background, interests, and goals of individual learners.

1.1.1 The Promise of Personalized Learning

We envision a world where access to high-quality, personally tailored educational experiences is affordable to all of the world’s learners. The key is to integrate textbooks, lectures, and homework assignments into a personalized learning system (PLS)
that closes the learning feedback loop by (i) continuously monitoring and analyzing learner interactions with learning resources in order to assess their learning progress and (ii) providing timely remediation, enrichment, or practice based on that analysis. See [1], [2], [3], [4], [5], and [6] for various visions and examples.

Some progress has been made over the past few decades on personalized learning; see, for example, the sizable literature on intelligent tutoring systems discussed in [7]. To date, the lion’s share of fielded, intelligent tutors have been rule-based systems that are hard-coded by domain experts to give learners feedback for pre-defined scenarios (e.g., [8], [9], [10], and [11]). The specificity of such systems is counterbalanced by their high development cost in terms of both time and money, which has limited their scalability and impact in practice.

In a fresh direction, recent progress has been made on applying machine learning algorithms to mine learner interaction data and educational content (see the overview articles by [12] and [13]). In contrast to rule-based approaches, machine learning-based PLSs promise to be rapid and inexpensive to deploy, which will enhance their scalability and impact. Indeed, the dawning age of “big data” provides new opportunities to build PLSs based on data rather than rules. We conceptualize the architecture of a generic machine learning-based PLS to have three interlocking components:

- **Learning analytics**: Algorithms that estimate what each learner does and does not understand based on data obtained from tracking their interactions with learning content.

- **Content analytics**: Algorithms that organize learning content such as text, video, simulations, questions, and feedback hints.

- **Scheduling**: Algorithms that use the results of learning and content analytics to
suggest to each learner at each moment what they should be doing in order to maximize their learning outcomes, in effect closing the learning feedback loop.

1.2 A Note on Notation

While we will make every effort to follow standard practices regarding notation we include this section as a concise reference.

**Vectors and Matrices**: We will denote all matrices using bold capital letters (e.g., \( W, Z, C \)) and use bold lowercase letters to denote vectors (e.g., \( w, z, c \)). Often, we will discuss vectors corresponding to certain rows or columns of matrices. We use the subscript notation to denote a vector corresponding to a particular column of a matrix (e.g., \( c_j \) is the \( j^{th} \) column of the matrix \( C \)). We employ an overbar on vectors to denote a column vector whose elements are taken from the row of another matrix (e.g., \( \bar{w}_i \) is the \( i^{th} \) row of the matrix \( W \), transposed into a column vector). To refer to an individual element of a matrix use regular (Roman) capitals letters followed by a subscript (e.g., \( Z_{i,j} \) is the element of \( Z \) in the \( i^{th} \) row and \( j^{th} \) column). For entries of a vector, we use regular lower case letters followed by a subscript (e.g., \( c_j \) is the \( j^{th} \) element of the vector \( c \).

**Probability and Statistics**: We use the notation \( P(X) \) to refer to the probability mass/density of a random variable \( X \). We will be explicitly about whether such variables are discrete are continuous when it is not immediately obvious from the context. We will further make explicit use of \( \Phi : \mathbb{R} \mapsto [0,1] \) to denote a probabilistic link functions. All other notation will follow standard conventions.
1.3 The Learning Analytics Challenge

In this thesis, we will develop novel algorithms for learning analytics (LA) and content analytics (CA). LA places models on both learners and questions and uses these models to i) estimate learner competence and ii) predict future learner success. Identifying learner enables a system to identify learner strengths and weaknesses automatically from data. It can then use this information for a variety of tasks. For example, LA can quickly inform learners (or their instructors) when they have sufficiently mastered certain material, need to improve on it, or possible refresh older material. Using this information, a PLS can then recommend the appropriate steps for a learner to take in order to maximize their learning goals. LA information can be used to predict future test scores which can be crucial for instructors wishing to maximize their students’ potential for success. As a final example, LA can be used to test whether learners are working independently or collaboratively with their peers. This information can be used to identifying cheating in educational contexts.

To be useful for large scale PLS, LA needs to be powerful, flexible, and interpretable. We discuss each of these needs below:

- **Deployable**: LA must be ready to deploy in new educational contexts with little to no human intervention.

- **Powerful**: LA must be able to powerfully model learners and questions such that it can accurately predict future learning performance.

- **Flexible**: LA must be able to seamlessly integrate information across multiple educational domains.

- **Interpretable**: LA must be human interpretable. That is, the output of an LA algorithm should be readily understood by learners, educators, and course content authors.
There exists a wide body of prior literature for LA. To date, however, none of this prior work satisfies all four of our essential requirements. We discuss these methods below.

**The Rasch Model:** The Rasch model [14, 15] assumes simply that learner $j$ can be modeled adequately with a single real-valued ability parameter $c_j$, with large positive values denoting strong ability and large negative values denoting weak ability. The $i^{th}$ question is modeled similarly by a single real-valued intrinsic difficulty parameter $\mu_i$, with large positive values denoting difficult questions and large negative values denoting very easy questions. The probability of success for learner $j$ on question $i$ is given by:

$$P_{i,j} = \Phi(c_j - \mu_i),$$

where $\Phi(\cdot)$ denote a link function (e.g., probit or logistic link) with maps the real-valued difference into a probability in $[0, 1]$.

The Rasch model is both simple to deploy and easily interpretable. However, it is not powerful enough to adequately model many real-world educational scenarios. As an example, a learner studying chemistry would undoubtedly encounter many different concepts. The learner may be very proficient at some, but also weak in others. The Rasch model is simply unable to accurately model such a scenario, making it very weak for real-world PLS.

**Multidimensional Item-Response Theory:** Multidimensional IRT (MIRT) is a generalization of the Rasch model that involves some number of additional parameters. Typically, they include the addition of a discrimination parameter $w_i$ for question $i$. The probability of success is then given by:

$$P_{i,j} = \Phi(w_i(c_j - \mu_i)).$$
Other variants of MIRT include additional variables for guessing (important on multiple choice tests). While the MIRT model does improve over the simple Rasch model, it still fails to integrate information across multiple educational domains and is hence, not powerful enough for real-world PLS.

**Collaborative Filtering:** A number of sophisticated methods based on collaborative filtering (CF) have been recently proposed, including low-rank models [16, 17], factor analysis (FA) [18, 19], and clustering-based methods [20]. CF techniques are incredibly powerful and can often have a very high success rate at predicting future interactions between learners and items. They are also quite flexible, and often require little training to deploy in contexts with items of mixed type (e.g., Netflix, Amazon rating system). A primary draw back of these models is the lack of interpretability in the results. Low rank and factor analysis models, for example, attempt to find a vector space representation for users and items (define these) that attempt to make sense out of the data. A typically formulation of the interaction between user $j$ and item $i$ is given by:

$$Y_{i,j} = \Phi(\mathbf{w}_i^T \mathbf{c}_j)$$

where the vector $w_i$ relates the qualities of item $i$ in a mathematical form while $c_j$ relates the qualities of the user. Stacking the vectors together produces a matrix product $\mathbf{WC}$ that is assumed to be of low-rank. The vast majority of the models in the literature assume real-valued output data such that the link function $\Phi$ is assumed to the identity operator. Some recent work [16] extends the model for use with more useful link functions. While these models can have excellent predictive performance, there is typically no structural constraints placed on $\mathbf{W}$ and $\mathbf{C}$ and, as a result, the factors are difficult if not impossible to interpret, making them a poor choice of LA.
1.4 The SPARFA model for LA

To address the shortcomings of current methods and provide a superior framework for LA, we will propose SPARFA (short for SPARse Factor Analysis) which consists of a statistical model and a fully Bayesian algorithm for fitting the SPARFA model to educational data. SPARFA meets all four of the required LA objectives. Concretely, SPARFA is i) readily deployable in diverse educational contexts, ii) powerful in that it can accurately model learners and predict their future successes, iii) flexible in that it can handle questions relating to various educational domains and iv) interpretable in that its output can be understood by a human.

1.5 Outline

The remainder of this thesis is organized as follows: In Chapter 2 we detail the SPARFA model and Bayesian SPARFA (SPARFA-B) algorithm. In Chapter 3 we detail Bayesian non-parametric extensions to SPARFA that enable flexible modeling of the number of latent variables in a statistically principled manner as well as methods for clustering similar learners together automatically. Following this, we discuss collaboration-type identification, an important application of SPARFA that searches for pairs of learners in a course that are collaborating on questions. We offer concluding remarks in Chapter 5.
SPARFA: SPARse Factor Analysis for Learning Analytics
2.1 Introduction

The SPARFA methodology, first proposed in [21], is a powerful approach to LA that overcomes many of the challenges of traditional LA methods. In this chapter we detail the SPARFA statistical model and derive the SPARFA-B algorithm which uses a fully Bayesian approach to fit the SPARFA model to data. We will then verify the utility of SPARFA on a number of synthetic and real-world datasets.
2.2 Statistical Model for Learning and Content Analytics

Here we detail the SPARFA statistical model for LA, which was first introduced in [21]. This model posits that the probability that a learner provides the correct response to a given question in terms of three factors: their knowledge of the underlying concepts, the concepts involved in each question, and each question’s intrinsic difficulty.

Figure 2.1 provides a graphical depiction of our approach. As shown in Figure 2.1(a), we are provided with data relating to the correctness of the learners’ responses to a collection of questions. We encode these graded responses in a “gradebook,” a source of information commonly used in the context of classical test theory [22]. Specifically, the “gradebook” is a matrix with entry $Y_{i,j} = 1$ or 0 depending on whether learner $j$ answers question $i$ correctly or incorrectly, respectively. Question marks correspond to incomplete data due to unanswered or unassigned questions. Working left-to-right in Figure 2.1(b), we assume that the collection of questions (rectangles) is related to a small number of abstract concepts (circles) by a bipartite graph, where the edge weight $W_{i,k}$ indicates the degree to which question $i$ involves concept $k$. We also assume that question $i$ has intrinsic difficulty $\mu_i$. Denoting learner $j$’s knowledge of concept $k$ by $C_{k,j}$, we calculate the probabilities that the learners answer the questions correctly in terms of $WC + M$, where $W$ and $C$ are matrix versions of $W_{i,k}$ and $C_{k,j}$, respectively, and $M$ is a matrix containing the intrinsic question difficulty $\mu_i$ on row $i$. We transform the probability of a correct answer to an actual 1/0 correctness via a standard probit or logit link function (see [23]).

Armed with this model and given incomplete observations of the graded learner–question responses $Y_{i,j}$, our goal is to estimate the factors $W$, $C$, and $M$. Such
Graded learner–question responses.

Inferred question–concept association graph.

Figure 2.1: (a) The SPARFA framework processes a (potentially incomplete) binary-valued dataset of graded learner–question responses to (b) estimate the underlying questions-concept association graph and the abstract conceptual knowledge of each learner (illustrated here by smiley faces for learner \( j = 3 \), the column in (a) selected by the red dashed box).

A factor-analysis problem is ill-posed in general, especially when each learner answers only a small subset of the collection of questions (see [24] for a factor analysis overview). Our first key observation that enables a well-posed solution is the fact that typical educational domains of interest involve only a small number of key concepts (i.e., we have \( K \ll N, Q \) in Fig. 2.1). Consequently, \( W \) becomes a tall, narrow \( Q \times K \) matrix that relates the questions to a small set of abstract concepts, while \( C \) becomes a short, wide \( K \times N \) matrix that relates learner knowledge to that same small set of abstract concepts. Note that the concepts are “abstract” in that they will be estimated from the data rather than dictated by a subject matter expert. Our second key observation is that each question involves only a small subset of the abstract concepts. Consequently, the matrix \( W \) is sparsely populated. Our third observation is that the entries of \( W \) should be non-negative, since we postulate that having strong concept knowledge should never hurt a learner’s chances to correctly answer questions. This constraint on \( W \) ensures that large positive values in \( C \) represent strong knowledge of the associated abstract concepts, which is crucial for a PLS to generate human-interpretable feedback to learners on their strengths and weaknesses.

Leveraging these observations, we propose a fully Bayesian method, SPARFA-
B, which produces posterior distributions of each parameter of interest. Since the concepts are abstract mathematical quantities estimated by SPARFA-B, we develop a post-processing step in Sec. 2.4 to facilitate interpretation of the estimated latent concepts by associating user-defined tags for each question with each abstract concept.

In Section 2.5, we report on a range of experiments with a variety of synthetic and real-world data that demonstrate the wealth of information provided by the estimates of W, C and M. As an example, Fig. 2.2 provides the results for a dataset collected from learners using STEMscopes [25], a science curriculum platform. The dataset consists of 145 Grade 8 learners from a single school district answering a manually tagged set of 80 questions on Earth science; only 13.5% of all graded learner–question responses were observed. We applied the SPARFA-B algorithm to retrieve the factors W, C, and M using 5 latent concepts. The resulting sparse matrix W is displayed as a bipartite graph in Fig. 2.2(a); circles denote the abstract concepts and boxes denote questions. Each question box is labeled with its estimated intrinsic difficulty $\mu_i$, with large positive values denoting easy questions. Links between the concept and question nodes represent the active (non-zero) entries of W, with thicker links denoting larger values $W_{i,k}$. Unconnected questions are those for which no concept explained the learners’ answer pattern; such questions typically have either very low or very high intrinsic difficulty, resulting in nearly all learners answering them correctly or incorrectly. The tags provided in Fig. 2.2(b) enable human-readable interpretability of the estimated abstract concepts.

We envision a range of potential learning and content analytics applications for the SPARFA framework that go far beyond the standard practice of merely forming column sums of the “gradebook” matrix (with entries $Y_{i,j}$) to arrive at a final scalar numerical score for each learner (which is then often further quantized to a letter grade on a 5-point scale). Each column of the estimated C matrix can be interpreted
(a) Inferred question–concept association graph.

<table>
<thead>
<tr>
<th>Concept 1</th>
<th>Concept 2</th>
<th>Concept 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Changes to land (45%)</td>
<td>Evidence of the past (74%)</td>
<td>Alternative energy (76%)</td>
</tr>
<tr>
<td>Properties of soil (28%)</td>
<td>Mixtures and solutions (14%)</td>
<td>Environmental changes (19%)</td>
</tr>
<tr>
<td>Uses of energy (27%)</td>
<td>Environmental changes (12%)</td>
<td>Changes from heat (5%)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Concept 4</th>
<th>Concept 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties of soil (77%)</td>
<td>Formulation of fossil fuels (54%)</td>
</tr>
<tr>
<td>Environmental changes (17%)</td>
<td>Mixtures and solutions (28%)</td>
</tr>
<tr>
<td>Classifying matter (6%)</td>
<td>Uses of energy (18%)</td>
</tr>
</tbody>
</table>

(b) Most important tags and relative weights for the estimated concepts.

Figure 2.2: (a) Sparse question–concept association graph and (b) most important tags associated with each concept for Grade 8 Earth science with $N = 135$ learners answering $Q = 80$ questions. Only 13.5% of all graded learner–question responses were observed.
as a measure of the corresponding learner’s knowledge about the abstract concepts. Low values indicate concepts ripe for remediation, while high values indicate concepts ripe for enrichment. The sparse graph stemming from the estimated $W$ matrix automatically groups questions into similar types based on their concept association; this graph makes it straightforward to find a set of questions similar to a given target question. Finally, the estimated $M$ matrix (with entries $\mu_i$ on each row) provides an estimate of each question’s intrinsic difficulty. This property enables an instructor to assign questions in an orderly fashion as well as to prune out potentially problematic questions that are either too hard, too easy, too confusing, or unrelated to the concepts underlying the collection of questions.

Our approach to learning and content analytics is based on a new statistical model that encodes the probability that a learner will answer a given question correctly in terms of three factors: (i) the learner’s knowledge of a set of latent, abstract concepts, (ii) how the question is related to each concept, and (iii) the intrinsic difficulty of the question.

### 2.2.1 Model for Graded Learner Response Data

Let $N$ denote the total number of learners, $Q$ the total number of questions, and $K$ the number of latent abstract concepts. We define $C_{k,j}$ as the concept knowledge of learner $j$ on concept $k$, with large positive values of $C_{k,j}$ corresponding to a better chance of success on questions related to concept $k$. Stack these values into the column vector $c_j \in \mathbb{R}^K$, $j \in \{1, \ldots, N\}$ and the $K \times N$ matrix $C = [c_1, \ldots, c_N]$. We further define $W_{i,k}$ as the question-concept association of question $i$ with respect to concept $k$, with larger values denoting stronger involvement of the concept. Stack these values into the column vector $\bar{w}_i \in \mathbb{R}^K$, $i \in \{1, \ldots, Q\}$ and the $Q \times K$ matrix $W = [\bar{w}_1, \ldots, \bar{w}_Q]^T$. Finally, we define the scalar $\mu_i \in \mathbb{R}$ as the intrinsic difficulty
of question $i$, with larger values representing easier questions. Stack these values into the column vector $\mu$ and form the $Q \times N$ matrix $M = \mu \mathbf{1}_{1 \times N}$ as the product of $\mu = [\mu_1, \ldots, \mu_Q]^T$ with the $N$-dimensional all-ones row vector $\mathbf{1}_{1 \times N}$.

Given these definitions, we propose the following model for the binary-valued graded response variable $Y_{i,j} \in \{0, 1\}$ for learner $j$ on question $i$, with 1 representing a correct response and 0 an incorrect response:

$$Z_{i,j} = \mathbf{w}_i^T \mathbf{c}_j + \mu_i, \quad \forall i, j,$$

$$Y_{i,j} \sim \text{Ber}(\Phi(Z_{i,j})), \quad (i, j) \in \Omega_{\text{obs}}. \quad (2.1)$$

Here, $\text{Ber}(z)$ designates a Bernoulli distribution with success probability $z$, and $\Phi(z)$ denotes an inverse link function$^1$ that maps a real value $z$ to the success probability of a binary random variable. Thus, the slack variable $\Phi(Z_{i,j}) \in [0, 1]$ governs the probability of learner $j$ answering question $i$ correctly.

The set $\Omega_{\text{obs}} \subseteq \{1, \ldots, Q\} \times \{1, \ldots, N\}$ in (2.1) contains the indices associated with the observed graded learner response data. Hence, our framework is able to handle the case of incomplete or missing data (e.g., when the learners do not answer all of the questions)$^2$. Stack the values $Y_{i,j}$ and $Z_{i,j}$ into the $Q \times N$ matrices $Y$ and $Z$, respectively. We can conveniently rewrite (2.1) in matrix form as

$$Y_{i,j} \sim \text{Ber}(\Phi(Z_{i,j})), \quad (i, j) \in \Omega_{\text{obs}} \quad \text{with} \quad Z = WC + M. \quad (2.2)$$

In this paper, we focus on the two most commonly used link functions in the

---

$^1$Inverse link functions are often called response functions in the generalized linear models literature (see, e.g., [26]).

$^2$Two common situations lead to missing learner response data. First, a learner might not attempt a question because it was not assigned or available to them. In this case, we simply exclude their response from $\Omega_{\text{obs}}$. Second, a learner might not attempt a question because it was assigned to them but was too difficult. In this case, we treat their response as incorrect, as is typical in standard testing settings.
machine learning literature. The \textit{inverse probit} function is defined as

\[
\Phi_{\text{pro}}(x) = \int_{-\infty}^{x} \mathcal{N}(t) \, dt = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-t^2/2} \, dt, \quad (2.3)
\]

where \( \mathcal{N}(t) = \frac{1}{\sqrt{2\pi}} e^{-t^2/2} \) is the probability density function (PDF) of the standard normal distribution (with mean zero and variance one). The \textit{inverse logit} link function is defined as

\[
\Phi_{\text{log}}(x) = \frac{1}{1 + e^{-x}}. \quad (2.4)
\]

As we noted in the Introduction, \( C, W, \) and \( \mu \) (or equivalently, \( M \)) have natural interpretations in real education settings. Column \( j \) of \( C \) can be interpreted as a measure of learner \( j \)'s knowledge about the abstract concepts, with larger \( C_{k,j} \) values implying more knowledge. The non-zero entries in \( W \) can be used to visualize the connectivity between concepts and questions (see Fig. 2.1(b) for an example), with larger \( W_{i,k} \) values implying stronger ties between question \( i \) and concept \( k \). The values of \( \mu \) contains estimates of each question’s intrinsic difficulty.

2.2.2 Joint Estimation of Concept Knowledge and Question–Concept Association

Given a (possibly partially observed) matrix of graded learner response data \( Y \), we aim to estimate the learner concept knowledge matrix \( C \), the question–concept association matrix \( W \), and the question intrinsic difficulty vector \( \mu \). In practice, the latent factors \( W \) and \( C \), and the vector \( \mu \) will contain many more unknowns than we have observations in \( Y \); hence, estimating \( W, C, \) and \( \mu \) is, in general, an ill-posed inverse problem. The situation is further exacerbated if many entries in \( Y \) are unobserved.
To regularize this inverse problem, prevent over-fitting, improve identifiability,\textsuperscript{1} and enhance interpretability of the entries in $\mathbf{C}$ and $\mathbf{W}$, we appeal to the following three observations regarding education that are reasonable for typical exam, homework, and practice questions at all levels. We will exploit these observations extensively in the sequel as fundamental assumptions:

(A1) \textit{Low-dimensionality:} The number of latent, abstract concepts $K$ is small relative to both the number of learners $N$ and the number of questions $Q$. This implies that the questions are redundant and that the learners’ graded responses live in a low-dimensional space. The parameter $K$ dictates the concept \textit{granularity}. Small $K$ extracts just a few general, broad concepts, whereas large $K$ extracts more specific and detailed concepts.\textsuperscript{2}

(A2) \textit{Sparsity:} Each question should be associated with only a small subset of the concepts in the domain of the course/assessment. In other words, we assume that the matrix $\mathbf{W}$ is sparsely populated, i.e., contains mostly zero entries.

(A3) \textit{Non-negativity:} A learner’s knowledge of a given concept does not negatively affect their probability of correctly answering a given question, i.e., knowledge of a concept is not “harmful.” In other words, the entries of $\mathbf{W}$ are non-negative, which provides a natural interpretation for the entries in $\mathbf{C}$: Large values $C_{k,j}$ indicate strong knowledge of the corresponding concept, whereas negative values indicate weak knowledge.

In practice, $N$ can be larger than $Q$ and vice versa, and hence, we do not impose any additional assumptions on their values. Assumptions (A2) and (A3) impose

\textsuperscript{1}If $\mathbf{Z} = \mathbf{WC}$, then for any orthonormal matrix $\mathbf{H}$ with $\mathbf{H}^T \mathbf{H} = \mathbf{I}$, we have $\mathbf{Z} = \mathbf{WH}^T \mathbf{HC} = \mathbf{WHC}$. Hence, the estimation of $\mathbf{W}$ and $\mathbf{C}$ is, in general, non-unique up to a unitary matrix rotation.

\textsuperscript{2}Standard techniques like cross-validation \cite{26} can be used to select $K$. We provide the corresponding details in Sec. 2.5.3. An alternative approach utilizing Bayesian non-parametrics to estimate $K$ directly from data is presented in Chapter 2.
sparsity and non-negativity constraints on $W$. Since these assumptions are likely to be violated under arbitrary unitary transforms of the factors, they help alleviate several well-known identifiability problems that arise in factor analysis.

We will refer to the problem of estimating $W$, $C$, and $\mu$, given the observations $Y$, under the assumptions (A1)–(A3) as the *SPARse Factor Analysis* (SPARFA) problem. In Section 2.3, we introduce SPARFA-B, a Bayesian method that produces full posterior estimates of the quantities of interest.
2.3 SPARFA-B: Bayesian Sparse Factor Analysis

SPARFA-B solves the SPARFA problem using a Bayesian method based on Markov chain Monte-Carlo (MCMC) sampling. By utilizing a Bayesian framework, SPARFA-B computes full posterior distributions for $W, C,$ and $\mu$ instead of simple point estimates.

SPARFA-B has several notable benefits in the context of learning and content analytics. First, the full posterior distributions enable the computation of informative quantities such as credible intervals and posterior modes for all parameters of interest. Second, since MCMC methods explore the full posterior space, they are not subject to being trapped indefinitely in local minima, which is possible with approaches based on convex optimization. Third, the hyperparameters used in Bayesian methods generally have intuitive meanings, in contrary to the regularization parameters of optimization-based methods. These hyperparameters can also be specially chosen to incorporate additional prior information about the problem.

2.3.1 Problem Formulation

As discussed in Sec. 2.2.2, we require the matrix $W$ to be both sparse (A2) and non-negative (A3). We enforce these assumptions through the following prior distributions that are a variant of the well-studied spike-slab model [27] adapted for non-negative factor loadings:

$$W_{i,k} \sim r_k \text{Exp}(\lambda_k) + (1 - r_k) \delta_0, \quad \lambda_k \sim G(a, \beta), \quad \text{and} \quad r_k \sim \text{Beta}(e, f). \quad (2.5)$$

Here, $\text{Exp}(x|\lambda) \sim \lambda e^{-\lambda x}, x \geq 0,$ and $G(a|\alpha, \beta) \sim \frac{\beta^\alpha x^{\alpha-1} e^{-\beta x}}{\Gamma(\alpha)}, x \geq 0,$ $\delta_0$ is the Dirac delta function, and $\alpha, \beta, e, f$ are hyperparameters. The model (2.5) uses the latent random variable $r_k$ to control the sparsity via the hyperparameters $e$ and $f$. This set
of priors induces a conjugate form on the posterior that enables efficient sampling. We note that both the exponential rate parameters $\lambda_k$ as well as the inclusion probabilities $r_k$ are grouped per factor. The remaining priors used in the proposed Bayesian model are summarized as

$$c_j \sim \mathcal{N}(0, V), \quad V \sim IW(V_0, h), \quad \text{and} \quad \mu_i \sim \mathcal{N}(\mu_0, v_\mu),$$

(2.6)

where $V_0, h, \mu_0$, and $v_\mu$ are hyperparameters.

### 2.3.2 The SPARFA-B Algorithm

We obtain posterior distribution estimates for the parameters of interest through an MCMC method based on the Gibbs’ sampler. To implement this, we must derive the conditional posteriors for each of the parameters of interest. We note again that the graded learner-response matrix $Y$ will not be fully observed, in general. Thus, our sampling method must be equipped to handle missing data.

The majority of the posterior distributions follow from standard results in Bayesian analysis and will not be derived in detail here. The exception is the posterior distribution of $W_{i,k} \forall i, k$. The spike-slab model that enforces sparsity in $W$ requires first sampling $W_{i,k} \neq 0 | Z, C, \mu$ and then sampling $W_{i,k} | Z, C, \mu$, for all $W_{i,k} \neq 0$. These posterior distributions differ from previous results in the literature due to our assumption of an exponential (rather than a normal) prior on $W_{i,k}$. We next derive these two results in detail.

#### 2.3.2.1 Derivation of Posterior Distribution of $W_{i,k}$

We seek both the probability that an entry $W_{i,k}$ is active (non-zero) and the distribution of $W_{i,k}$ when active given our observations. The following theorem states the
Theorem 1 (Posterior distributions for $\mathbf{W}$). For all $i = 1, \ldots, Q$ and all $k = 1, \ldots, K$, the posterior sampling results for $W_{i,k} = 0 | \mathbf{Z}, \mathbf{C}, \mu$ and $W_{i,k} | \mathbf{Z}, \mathbf{C}, \mu, W_{i,k} \neq 0$ are given by

$$
\hat{R}_{i,k} = p(W_{i,k} = 0 | \mathbf{Z}, \mathbf{C}, \mu) = \frac{\mathcal{N}(\tilde{M}_{i,k}, \tilde{S}_{i,k}, \lambda_k)}{\mathcal{N}(\tilde{M}_{i,k}, \tilde{S}_{i,k}, \lambda_k)} (1-r_k),
$$

$$
W_{i,k} | \mathbf{Z}, \mathbf{C}, \mu, W_{i,k} \neq 0 \sim \mathcal{N}(\tilde{M}_{i,k}, \tilde{S}_{i,k}, \lambda_k),
$$

$$
\tilde{M}_{i,k} = \sum_{\{j:(i,j) \in \Omega_{obs}\}}((Z_{i,j} - \mu_i) - \sum_{k' \neq k} W_{i,k'} C_{k',j}) C_{k,j},
$$

$$
\tilde{S}_{i,k} = \left( \sum_{\{j:(i,j) \in \Omega_{obs}\}} C_{k,j}^2 \right)^{-1},
$$

where $\mathcal{N}(x|m,s,\lambda) = \frac{e^{\lambda m - \lambda^2 s^2/2}}{\sqrt{2\pi s^2 \Phi(m - \lambda s)}} e^{-(x-m)^2/2s - \lambda m}$ represents a rectified normal distribution (see [28]).

### 2.3.2.2 Sampling Methodology

SPARFA-B carries out the following MCMC steps to compute posterior distributions for all parameters of interest:

1. For all $(i,j) \in \Omega_{obs}$, draw $Z_{i,j} \sim \mathcal{N}((\mathbf{W}\mathbf{C})_{i,j} + \mu_i, 1)$, truncating below 0 if $Y_{i,j} = 1$, and truncating above 0 if $Y_{i,j} = 0$.

2. For all $i = 1, \ldots, Q$, draw $\mu_i \sim \mathcal{N}(m_i, v)$ with $v = (v_\mu^{-1} + n')^{-1}$, $m_i = \mu_0 + v \sum_{\{j:(i,j) \in \Omega_{obs}\}} (Z_{i,j} - \tilde{\mathbf{w}}_i^T \mathbf{c}_j)$, and $n'$ the number of learners responding to question $i$.

3. For all $j = 1, \ldots, N$, draw $\mathbf{c}_j \sim \mathcal{N}(\mathbf{m}_j, \mathbf{M}_j)$ with $\mathbf{M}_j = (\mathbf{V}^{-1} + \tilde{\mathbf{W}}^T \tilde{\mathbf{W}})^{-1}$, and $\mathbf{m}_j = \mathbf{M}_j \tilde{\mathbf{W}}^T (\mathbf{z}_j - \tilde{\mu})$. The notation $\tilde{\cdot}$ denotes the restriction of the vector or matrix to the set of rows $i : (i,j) \in \Omega_{obs}$. 
4. Draw $\mathbf{V} \sim IW(\mathbf{V}_0 + \mathbf{CC}^T, N + h)$.

5. For all $i = 1, \ldots, Q$ and $k = 1, \ldots, K$, draw $W_{i,k} \sim \mathcal{N}(\hat{R}_{i,k} \mathbf{N}^r(\hat{M}_{i,k}, \hat{S}_{i,k}) + (1 - \hat{R}_{i,k})\delta_0$, where $\hat{R}_{i,k}, \hat{M}_{i,k},$ and $\hat{S}_{i,k}$ are as stated in Thm. 1.

6. For all $k = 1, \ldots, K$, let $b_k$ define the number of active (i.e., non-zero) entries of $\mathbf{\bar{w}}_k$. Draw $\lambda_k \sim Ga(\alpha + b_k, \beta + \sum_{i=1}^{Q} W_{i,k})$.

7. For all $k = 1, \ldots, K$, draw $r_k \sim \text{Beta}(e + b_k, f + Q - b_k)$, with $b_k$ defined as in Step 6.

2.3.3 Algorithmic Details and Improvements for SPARFA-B

Here we discuss some several practical issues for efficiently implementing SPARFA-B, selecting the hyperparameters, and techniques for easy visualization of the SPARFA-B results.

2.3.3.1 Improving Computational Efficiency

The Gibbs sampling scheme of SPARFA-B enables efficient implementation in several ways. First, draws from the truncated normal in Step 1 of Section 2.3.2.2 are decoupled from one another, allowing them to be performed independently and, potentially, in parallel. Second, sampling of the elements in each column of $\mathbf{W}$ can be carried out in parallel by computing the relevant factors of Step 5 in matrix form. Since $K \ll Q, N$ by assumption (A1), the relevant parameters are recomputed only a relatively small number of times. One taxing computation is the calculation of the covariance matrix $\mathbf{M}_j$ for each $j = 1, \ldots, N$ in Step 3. This computation is necessary, since we do not constrain each learner to answer the same set of questions which, in turn, changes the nature of the covariance calculation for each individual learner. For
data sets where all learners answer the same set of questions, this covariance matrix is the same for all learners and, hence, can be carried out once per MCMC iteration.

2.3.3.2 Parameter Selection

The selection of the hyperparameters is performed at the discretion of the user. As is typical for Bayesian methods, non-informative (broad) hyperparameters can be used to avoid biasing results and to allow for adequate exploration of the posterior space. Tighter hyperparameters can be used when additional side information is available. For example, prior information from subject matter experts might indicate which concepts are related to which questions or might indicate the intrinsic difficulty of the questions.

2.3.3.3 Post-Processing for Data Visualization

As discussed above, the generation of posterior statistics is one of the primary advantages of SPARFA-B. However, for many tasks, such as visualization of the retrieved knowledge base, it is often convenient to post-process the output of SPARFA-B to obtain point estimates for each parameter. For many Bayesian methods, simply computing the posterior mean is often sufficient. This is the case for most parameters computed by SPARFA-B, including $C$ and $\mu$. The posterior mean of $W$, however, is generally non-sparse, since the MCMC will generally explore the possibility of including each entry of $W$. Nevertheless, we can easily generate a sparse $W$ by examining the posterior mean of the inclusion statistics contained in $\tilde{R}_{i,k}$, $\forall i,k$. Concretely, if the posterior mean of $\tilde{R}_{i,k}$ is small, then we set the corresponding entry of $W_{i,k}$ to zero. Otherwise, we set $W_{i,k}$ to its posterior mean. We will make use of this method throughout the experiments presented in Sec. 2.5.
2.3.4 Related Work on Bayesian Sparse Factor Analysis

Sparsity models for Bayesian factor analysis have been well-explored in the statistical literature [19, 27]. One popular avenue for promoting sparsity is to place a prior on the variance of each component in $\mathbf{W}$ (see, e.g., [19], [29], and [18]). In such a model, large variance values indicate active components, while small variance values indicate inactive components. Another approach is to model active and inactive components directly using a form of a spike-slab model due to [27] and used in [30], [31], and [32]:

$$W_{i,k} \sim r_k \mathcal{N}(0, v_k) + (1 - r_k) \delta_0, \quad v_k \sim IG(\alpha, \beta), \quad \text{and} \quad r_k \sim Beta(e, f).$$

The approach employed in (2.5) utilizes a spike-slab prior with an exponential distribution, rather than a normal distribution, for the active components of $\mathbf{W}$. We chose this prior for several reasons: First, it enforces the non-negativity assumption (A3). Second, it induces a posterior distribution that can be both computed in closed form and sampled efficiently. Third, its tail is slightly heavier than that of a standard normal distribution, which improves the exploration of quantities further away from zero.

A sparse factor analysis model with non-negativity constraints that is related to the one proposed here was discussed in [33], although their methodology is quite different from ours. Specifically, they impose non-negativity on the (dense) matrix $\mathbf{C}$ rather than on the sparse factor loading matrix $\mathbf{W}$. Furthermore, they enforce non-negativity using a truncated normal$^1$ rather than an exponential prior.

$^1$One could alternatively employ a truncated normal distribution on the support $[0, \infty)$ for the active entries in $\mathbf{W}$. In experiments with this model, we found a slight, though noticeable, improvement in prediction performance on real-data experiments using the exponential prior. We will however, consider the use of the truncated normal prior in Chapters 3 and 4.
2.4 Tag Analysis: Post-Processing to Interpret the Estimated Concepts

We have developed SPARFA-B to estimate $W$, $C$, and $\mu$ (or equivalently, $M$) in (2.2) given the partial binary observations in $Y$. Both $W$ and $C$ encode a small number of latent concepts. As we initially noted, the concepts are "abstract" in that they are estimated from the data rather than dictated by a subject matter expert. In this section we develop a principled post-processing approach to interpret the meaning of the abstract concepts after they have been estimated from learner responses, which is important if our results are to be usable for learning analytics and content analytics in practice. Our approach applies when the questions come with a set of user-generated “tags” or “labels” that describe in a free-form manner what ideas underlie each question.

We develop a post-processing algorithm for the estimated matrices $W$ and $C$ that estimates the association between the latent concepts and the user-generated tags, enabling concepts to be interpreted as a “bag of tags.” Additionally, we show how to extract a personalized tag knowledge profile for each learner. The efficacy of our tag-analysis framework will be demonstrated in the real-world experiments in Sec. 2.5.2.

2.4.1 Incorporating Question–Tag Information

Suppose that a set of tags has been generated for each question that represent the topic(s) or theme(s) of each question. The tags could be generated by the course instructors, subject matter experts, learners, or, more broadly, by crowd-sourcing. In general, the tags provide a redundant representation of the true knowledge components, i.e., concepts are associated to a “bag of tags.”
Assume that there is a total number of $M$ tags associated with the $Q$ questions. We form a $Q \times M$ matrix $T$, where each column of $T$ is associated to one of the $M$ pre-defined tags. We set $T_{i,m} = 1$ if tag $m \in \{1, \ldots, M\}$ is present in question $i$ and 0 otherwise. Now, we postulate that the question association matrix $W$ extracted by SPARFA can be further factorized as $W = TA$, where $A$ is an $M \times K$ matrix representing the tags-to-concept mapping. This leads to the following additional assumptions:

(A4) *Non-negativity:* The matrix $A$ is non-negative. This increases the interpretability of the result, since concepts should not be negatively correlated with any tags, in general.

(A5) *Sparsity:* Each column of $A$ is sparse. This ensures that the estimated concepts relate to only a few tags.

2.4.2 Estimating the Concept–Tag Associations and Learner–Tag Knowledge

The assumptions (A4) and (A5) enable us to extract $A$ using $\ell_1$-norm regularized non-negative least-squares as described in [26] and [34]. Specifically, to obtain each column $a_k$ of $A$, $k = 1, \ldots, K$, we solve the following convex optimization problem, a non-negative variant of *basis pursuit denoising*:

\[
(BPDN_+) \quad \minimize_{a_k : A_m,k \geq 0 \forall m} \frac{1}{2} \|w_k - Ta_k\|_2^2 + \eta \|a_k\|_1.
\]

Here, $w_k$ represents the $k^{th}$ column of $W$, and the parameter $\eta$ controls the sparsity level of the solution $a_k$.

We propose a first-order method derived from the FISTA framework in [35] to solve (BPDN$_+$). The algorithm consists of two steps: A gradient step with respect
to the $\ell_2$-norm penalty function, and a projection step with respect to the $\ell_1$-norm regularizer subject to the non-negative constraints on $a_k$. By solving (BPDN$_+$) for $k = 1, \ldots, K$, and building $A = [a_1, \ldots, a_K]$, we can (i) assign tags to each concept based on the non-zero entries in $A$ and (ii) estimate a tag-knowledge profile for each learner.

2.4.2.1 Associating Tags to Each Concept

Using the concept–tag association matrix $A$ we can directly associate tags to each concept estimated by SPARFA. We first normalize the entries in $a_k$ such that they sum to one. With this normalization, we can then calculate percentages that show the proportion of each tag that contributes to concept $k$ corresponding to the non-zero entries of $a_k$. This concept tagging method typically will assign multiple tags to each concept, thus, enabling one to identify the coarse meaning of each concept (see Sec. 2.5.2 for examples using real-world data).

2.4.2.2 Learner Tag Knowledge Profiles

Using the concept–tag association matrix $A$, we can assess each learner’s knowledge of each tag. To this end, we form an $M \times N$ matrix $U = AC$, where the $U_{m,j}$ characterizes the knowledge of learner $j$ of tag $m$. This information could be used, for example, by a PLS to automatically inform each learner which tags they have strong knowledge of and which tags they do not. Course instructors can use the information contained in $U$ to extract measures representing the knowledge of all learners on a given tag, e.g., to identify the tags for which the entire class lacks strong knowledge. This information would enable the course instructor to select future learning content that deals with those specific tags. A real-world example demonstrating the efficacy of this framework is shown below in Sec. 2.5.2.1.
2.5 Experiments

In this section, we validate SPARFA-B on both synthetic and real-world educational data sets. First, using synthetic data, we validate that the algorithms can accurately estimate the underlying factors from binary-valued observations and characterize their performance under different circumstances. Specifically, we benchmark the factor estimation performance of SPARFA-B against a maximum likelihood approach to the SPARFA model based on convex optimization dubbed SPARFA-M [21], as well as a variant of the well-established K-SVD algorithm [36] used in dictionary-learning applications. Second, using real-world graded learner-response data we demonstrate the efficacy of SPARFA-B for learning and content analytics. Specifically, we showcase how the estimated learner concept knowledge, question–concept association, and intrinsic question difficulty can support machine learning-based personalized learning. Finally, we demonstrate the ability of SPARFA-B to impute missing data and compare its performance against existing LA methods.

2.5.1 Synthetic Data Experiments

We first characterize the estimation performance of SPARFA-M and SPARFA-B using synthetic test data generated from a known ground truth model. We generate instances of $W$, $C$, and $\mu$ under pre-defined distributions and then generate the binary-valued observations $Y$ according to (2.2).

Our report on the synthetic experiments is organized as follows. In Sec. 2.5.1.1, we outline the SPARFA-M algorithm, which is an alternative to SPARFA-B that relies on bi-convex optimization to fit the SPARFA parameters to data. Next, in Sec. 2.5.1.2, we outline K-SVD+, a variant of the well-established K-SVD dictionary-learning (DL) algorithm originally proposed in [36]. In Sec. 2.5.1.3 we detail the performance metrics. We compare SPARFA-B, SPARFA-M, and K-SVD+ as we vary the problem size.
and number of concepts (Sec. 2.5.1.4), observation incompleteness (Sec. 2.5.1.5), and the sparsity of $W$ (Sec. 2.5.1.6). In the above-referenced experiments, we simulate the observation matrix $Y$ via the inverse probit link function and use only the probit variant of SPARFA-M in order to make a fair comparison with SPARFA-B. In a real-world situation, however, the link function is generally unknown. In Sec. 2.5.1.7 we conduct model-mismatch experiments, where we generate data from one link function but analyze assuming the other.

In all synthetic experiments, we average the results of all performance measures over 25 Monte-Carlo trials, for each instance of the model parameters we control.

2.5.1.1 Baseline Algorithm 1: SPARFA-M

A maximum likelihood estimator for the SPARFA model parameters was first proposed in [21]. This approach, dubbed SPARFA-M, uses bi-convex optimization to solve for $W$ and $C$ by first initializing both parameters at random. It then holds $W$ fixed and performs an optimization step on $C$. After this, $C$ is held fixed and the optimization step is performed on $W$. The primary disadvantage of this approach is that, because of the bi-convex nature of the SPARFA problem, SPARFA-M cannot guarantee convergence to the optimal solution. Rather, it will converge only to some local optima. The advantage of SPARFA-M, however, is speed as SPARFA-M can typically converge very quickly to one of the local optimal points.

2.5.1.2 Baseline Algorithm 2: K-SVD$_+$

Since we are not aware of any existing algorithms to solve (2.2) subject to the assumptions (A1)–(A3), we deploy a novel baseline algorithm based on the well-known K-SVD algorithm of [36], which is widely used in various dictionary learning settings but ignores the inverse probit or logit link functions. Since the standard K-SVD al-
gorithm also ignores the non-negativity constraint used in the SPARFA model, we develop a variant of the non-negative K-SVD algorithm proposed in [37] that we refer to as K-SVD+. In the sparse coding stage of K-SVD+, we use the non-negative variant of orthogonal matching pursuit (OMP) outlined in [38]; that is, we enforce the non-negativity constraint by iteratively picking the entry corresponding to the maximum inner product without taking its absolute value. We also solve a non-negative least-squares problem to determine the residual error for the next iteration. In the dictionary update stage of K-SVD+, we use a variant of the rank-one approximation algorithm detailed in [37, Figure 4], where we impose non-negativity on the elements in $\mathbf{W}$ but not on the elements of $\mathbf{C}$.

K-SVD+ has as input parameters the sparsity level of each row of $\mathbf{W}$. In what follows, we provide K-SVD+ with the known ground truth for the number of non-zero components in order to obtain its best-possible performance. This will favor K-SVD+ over both SPARFA algorithms, since, in practice, such oracle information is not available.

2.5.1.3 Performance Measures

In each simulation, we evaluate the performance of SPARFA-B, SPARFA-M, and K-SVD+ by comparing the fidelity of the estimates $\hat{\mathbf{W}}$, $\hat{\mathbf{C}}$, and $\hat{\boldsymbol{\mu}}$ to the ground truth $\mathbf{W}$, $\mathbf{C}$, and $\boldsymbol{\mu}$. Performance evaluation is complicated by the facts that (i) SPARFA-B outputs posterior distributions rather than simple point estimates of the parameters and (ii) factor-analysis methods are generally susceptible to permutation of the latent factors. We address the first concern by post-processing the output of SPARFA-B to obtain point estimates for $\mathbf{W}$, $\mathbf{C}$, and $\boldsymbol{\mu}$ as detailed in Sec. 2.3.3.3 using $\hat{R}_{i,k} < 0.35$ for the threshold value. We address the second concern by normalizing the columns of $\mathbf{W}$, $\hat{\mathbf{W}}$ and the rows of $\mathbf{C}$, $\hat{\mathbf{C}}$ to unit $\ell_2$-norm, permuting the columns of $\hat{\mathbf{W}}$ and
\( \hat{\mathbf{C}} \) to best match the ground truth, and then compare \( \mathbf{W} \) and \( \mathbf{C} \) with the estimates \( \hat{\mathbf{W}} \) and \( \hat{\mathbf{C}} \). We also compute the Hamming distance between the support set of \( \mathbf{W} \) and that of the (column-permuted) estimate \( \hat{\mathbf{W}} \). To summarize, the performance measures used in the sequel are

\[
E_{\mathbf{W}} = \| \mathbf{W} - \hat{\mathbf{W}} \|_F^2 / \| \mathbf{W} \|_F^2, \\
E_{\mathbf{C}} = \| \mathbf{C} - \hat{\mathbf{C}} \|_F^2 / \| \mathbf{C} \|_F^2, \\
E_{\mu} = \| \mu - \hat{\mu} \|_2^2 / \| \mu \|_2^2, \\
E_{\mathbf{H}} = \| \mathbf{H} - \hat{\mathbf{H}} \|_F^2 / \| \mathbf{H} \|_F^2,
\]

where \( \mathbf{H} \in \{0, 1\}^{Q \times K} \) with \( H_{i,k} = 1 \) if \( W_{i,k} > 0 \) and \( H_{i,k} = 0 \) otherwise. The \( Q \times K \) matrix \( \hat{\mathbf{H}} \) is defined analogously using \( \hat{\mathbf{W}} \).

### 2.5.1.4 Impact of Problem Size and Number of Concepts

In this experiment, we study the performance of SPARFA-B, SPARFA-M, and KSVD+ as we vary the number of learners \( N \), the number of questions \( Q \), and the number of concepts \( K \).

**Experimental setup** We vary the number of learners \( N \in \{50, 100, 200\} \), the number of questions \( Q \in \{50, 100, 200\} \), and the number of concepts \( K \in \{5, 10\} \). For each combination of \((N, Q, K)\), we generate \( \mathbf{W}, \mathbf{C}, \mu, \) and \( \mathbf{Y} \) according to (2.5) and (2.6) with \( v_\mu = 1, \lambda_k = 2/3 \ \forall k, \) and \( \mathbf{V}_0 = \mathbf{I}_K \). For each instance, we choose the number of non-zero entries in each row of \( \mathbf{W} \) as \( DU(1, 3) \) where \( DU(a, b) \) denotes the discrete uniform distribution in the range \( a \) to \( b \). For each trial, we run the probit version of SPARFA-M, SPARFA-B, and K-SVD+ to obtain the estimates \( \hat{\mathbf{W}}, \hat{\mathbf{C}}, \hat{\mu}, \) and calculate \( \hat{\mathbf{H}} \). For all of the synthetic experiments with SPARFA-M, we set the regularization parameters \( \gamma = 0.1 \) and select \( \lambda \) using the BIC [26]. For SPARFA-B, we set the hyperparameters to \( h = K + 1, v_\mu = 1, \alpha = 1, \beta = 1.5, e = 1, \) and \( f = 1.5; \) moreover, we burn-in the MCMC for 30,000 iterations and take output samples over
the next 30,000 iterations.

**Results and discussion**  Fig. 2.3 shows box-and-whisker plots for the three algorithms and the four performance measures. We observe that the performance of all of the algorithms generally improves as the problem size increases. Moreover, SPARFA-B has superior performance for $E_W$, $E_C$, and $E_\mu$. We furthermore see that both SPARFA-B and SPARFA-M outperform K-SVD$^+$ on $E_W$, $E_C$, and especially $E_\mu$. K-SVD$^+$ performs very well in terms of $E_H$ (slightly better than both SPARFA-M and SPARFA-B) due to the fact that we provide it with the oracle sparsity level, which is, of course, not available in practice. SPARFA-B’s improved estimation accuracy over SPARFA-M comes at the price of significantly higher computational complexity. For example, for $N = Q = 200$ and $K = 5$, SPARFA-B requires roughly 10 minutes on a 3.2 GHz quad-core desktop PC, while SPARFA-M and K-SVD$^+$ require only 6 s.

**2.5.1.5 Impact of the Number of Incomplete Observations**

In this experiment, we study the impact of the number of observations in $Y$ on the performance of the probit version of SPARFA-M, SPARFA-B, and K-SVD$^+$.

**Experimental setup**  We set $N = Q = 100$, $K = 5$, and all other parameters as in Sec. 2.5.1.4. We then vary the percentage $P_{\text{obs}}$ of entries in $Y$ that are observed as 100%, 80%, 60%, 40%, and 20%. The locations of missing entries are generated i.i.d. and uniformly over the entire matrix.

**Results and discussion**  Fig. 2.4 shows that the estimation performance of all methods degrades gracefully as the percentage of missing observations increases. Again, SPARFA-B outperforms the other algorithms on $E_W$, $E_C$, and $E_\mu$. K-SVD$^+$ performs worse than both SPARFA algorithms except on $E_H$, where it achieves com-
Figure 2.3: Performance comparison of SPARFA-M, SPARFA-B, and K-SVD \( + \) for different problem sizes \( Q \times N \) and number of concepts \( K \). The performance naturally improves as the problem size increases, while both SPARFA algorithms outperform K-SVD \( + \). M denotes SPARFA-M, B denotes SPARFA-B, and K denotes KSVD \( + \).
Figure 2.4: Performance comparison of SPARFA-M, SPARFA-B, and K-SVD+ for different percentages of observed entries in Y. The performance degrades gracefully as the number of observations decreases, while the SPARFA algorithms outperform K-SVD+.

parable performance. We conclude that SPARFA-M and SPARFA-B can both reliably estimate the underlying factors, even in cases of highly incomplete data with SPARFA-B having the best performance.

2.5.1.6 Impact of Sparsity Level

In this experiment, we study the impact of the sparsity level in W on the performance of the probit version of SPARFA-M, SPARFA-B, and K-SVD+.

**Experimental setup**  We choose the active entries of W i.i.d. Ber(q) and vary $q \in \{0.2, 0.4, 0.6, 0.8\}$ to control the number of non-zero entries in each row of W. All other parameters are set as in Sec. 2.5.1.4. This data generation method allows for scenarios in which some rows of W contain no active entries as well as all active entries. We set the hyperparameters for SPARFA-B to $h = K + 1 = 6$, $v_\mu = 1$, and $e = 1$, and $f = 1.5$. For $q = 0.2$ we set $\alpha = 2$ and $\beta = 5$. For $q = 0.8$ we set $\alpha = 5$ and $\beta = 2$. For all other cases, we set $\alpha = \beta = 2$.

**Results and discussion**  Fig. 2.5 shows that sparser W lead to lower estimation errors. This demonstrates that SPARFA-B outperforms SPARFA-M and K-SVD+ across all metrics. The performance of K-SVD+ is worse than both SPARFA algorithms except on the support estimation error $E_{ WH}$, which is due to the fact that K-SVD+ is aware of the oracle sparsity level.
2.5.1.7 Impact of Model Mismatch

In this experiment, we examine the impact of model mismatch by using a link function for estimation that does not match the true link function from which the data is generated.

**Experimental setup** We fix $N = Q = 100$ and $K = 5$, and set all other parameters as in Sec. 2.5.1.4. Then, for each generated instance of $W$, $C$ and $\mu$, we generate $Y_{\text{pro}}$ and $Y_{\text{log}}$ according to both the inverse probit link and the inverse logit link, respectively. We then run SPARFA-M (both the probit and logit variants), SPARFA-B (which uses only the probit link function), and K-SVD$_+$ on both $Y_{\text{pro}}$ and $Y_{\text{log}}$.

**Results and discussion** Fig. 2.6 shows that model mismatch does not severely affect $E_W$, $E_C$, and $E_H$ for both SPARFA-M and SPARFA-B. However, due to the difference in the functional forms between the probit and logit link functions, model mismatch does lead to an increase in $E_\mu$ for both SPARFA algorithms. We also see that K-SVD$_+$ performs worse than both SPARFA methods, since it ignores the link function. We note, however, that even with the incorrect model, SPARFA-B still outperforms SPARFA-M working under the correct model.
2.5.2 Real Data Experiments

We next test the SPARFA-B algorithms on three real-world educational datasets. In what follows, we select the hyperparameters for SPARFA-B to be largely non-informative.

2.5.2.1 Undergraduate DSP course

Dataset We analyze a very small dataset consisting of $N = 15$ learners answering $Q = 44$ questions taken from the final exam of an introductory course on digital signal processing (DSP) taught at Rice University in Fall 2011 [39]. There is no missing data in the matrix $Y$.

Analysis We estimate $W$, $C$, and $\mu$ from $Y$ assuming $K = 5$ concepts to achieve a concept granularity that matches the complexity of the analyzed dataset. Since the questions had been manually tagged by the course instructor, we deploy the tag-analysis approach proposed in Sec. 2.4. Specifically, we form a $44 \times 12$ matrix $T$ using the $M = 12$ available tags and estimate the $12 \times 5$ concept–tag association matrix $A$ in order to interpret the meaning of each retrieved concept. For each concept, we only show the top 3 tags and their relative contributions. We also compute the $12 \times 15$ learner tag knowledge profile matrix $U$. 

Figure 2.6: Performance comparison of SPARFA-M, SPARFA-B, and K-SVD$+$ with probit/logit model mismatch; $M_P$ and $M_L$ indicate probit and logit SPARFA-M, respectively. In the left/right halves of each box plot, we generate $Y$ according to the inverse probit/logit link functions. The performance degrades only slightly with mismatch, while both SPARFA algorithms outperform K-SVD$+$. 

(a) Question–concept association graph. Circles correspond to concepts and rectangles to questions; the values in each rectangle corresponds to that question’s intrinsic difficulty.

<table>
<thead>
<tr>
<th>Concept 1</th>
<th>Concept 2</th>
<th>Concept 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency response</td>
<td>Fourier transform (46%)</td>
<td>z-transform (66%)</td>
</tr>
<tr>
<td>Sampling rate</td>
<td>Laplace transform (36%)</td>
<td>Pole/zero plot (22%)</td>
</tr>
<tr>
<td>Aliasing</td>
<td>z-transform (24%)</td>
<td>Laplace transform (12%)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Concept 4</th>
<th>Concept 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fourier transform</td>
<td>Impulse response (74%)</td>
</tr>
<tr>
<td>Systems/circuits</td>
<td>Transfer function (15%)</td>
</tr>
<tr>
<td>Transfer function</td>
<td>Fourier transform (11%)</td>
</tr>
</tbody>
</table>

(b) Most important tags and relative weights for the estimated concepts.

Figure 2.7: (a) Question–concept association graph and (b) most important tags associated with each concept for an undergraduate DSP course with $N = 15$ learners answering $Q = 44$ questions.
Table 2.1: Selected tag knowledge of Learner 1.

<table>
<thead>
<tr>
<th>z-transform</th>
<th>Impulse response</th>
<th>Transfer function</th>
<th>Fourier transform</th>
<th>Laplace transform</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.09</td>
<td>-1.80</td>
<td>-0.50</td>
<td>0.99</td>
<td>-0.77</td>
</tr>
</tbody>
</table>

Table 2.2: Average tag knowledge of all learners.

<table>
<thead>
<tr>
<th>z-transform</th>
<th>Impulse response</th>
<th>Transfer function</th>
<th>Fourier transform</th>
<th>Laplace transform</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.04</td>
<td>-0.03</td>
<td>-0.10</td>
<td>0.11</td>
<td>0.03</td>
</tr>
</tbody>
</table>

Results and discussion  

Fig. 2.7(a) visualizes the estimated question–concept association matrix $\hat{W}$ as a bipartite graph consisting of question and concept nodes. In the graph, circles represent the estimated concepts and squares represent questions, with thicker edges indicating stronger question–concept associations (i.e., larger entries $\hat{W}_{i,k}$). Questions are also labeled with their estimated intrinsic difficulty $\mu_i$, with larger positive values of $\mu_i$ indicating easier questions. Note that ten questions are not linked to any concept. All $Q = 15$ learners answered these questions correctly; as a result nothing can be estimated about their underlying concept structure. Fig. 2.7(b) provides the concept–tag association (top 3 tags) for each of the 5 estimated concepts.

Tbl. 2.1 provides the posterior mean of Learner 1’s knowledge of the various tags relative to other learners. Large positive values mean that Learner 1 has strong knowledge of the tag, while large negative values indicate a deficiency in knowledge of the tag. Tbl. 2.2 shows the average tag knowledge of the entire class, computed by averaging the entries of each row in the learner tag knowledge matrix $U$ as described in Sec. 2.4. Tbl. 2.1 indicates that Learner 1 has particularly weak knowledges of the tag “Impulse response.” Armed with this information, a PLS could automatically suggest remediation about this concept to Learner 1. Tbl. 2.2 indicates that the

---

1To avoid the scaling identifiability problem that is typical in factor analysis, we normalize each row of $C$ to unit $\ell_2$-norm and scale each column of $W$ accordingly prior to visualizing the bipartite graph. This enables us to compare the strength of question–concept associations across different concepts.
entire class has (on average) weak knowledge of the tag “Transfer function.” With this information, a PLS could suggest to the class instructor that they provide remediation about this concept to the entire class.

2.5.2.2 Grade 8 science course

Dataset The STEMscopes dataset was introduced in Sec. 2.2. There is substantial missing data in the matrix $Y$, with only 13.5% of its entries observed.

Analysis We compare the results of SPARFA-M and SPARFA-B on this data set to highlight the pros and cons of each approach. For both algorithms, we select $K = 5$ concepts. For SPARFA-B, we fix reasonably broad (non-informative) values for all hyperparameters. For $\mu_0$ we calculate the average rate of correct answers $p_s$ on observed graded responses of all learners to all questions and use $\mu_0 = \Phi_{pro}^{-1}(p_s)$. The variance $\nu_\mu$ is left sufficiently broad to enable adequate exploration of the intrinsic difficulty for each questions. Point estimates of $W$, $C$ and $\mu$ are generated from the SPARFA-B posterior distributions using the methods described in Sec. 2.3.3.3. Specifically, an entry $\hat{W}_{i,k}$ that has a corresponding active probability $\hat{R}_{i,k} < 0.55$ is thresholded to 0. Otherwise, we set $\hat{W}_{i,k}$ to its posterior mean. On a 3.2 GHz quad-core desktop PC, SPARFA-M converged to its final estimates in 4 s, while SPARFA-B required 10 minutes.

Results and discussion Both SPARFA-M and SPARFA-B deliver comparable factorizations. The estimated question–concept association graph for SPARFA-B is shown in Fig. 2.2(a), with the accompanying concept–tag association in Fig. 2.2(b). Again we see a sparse relationship between questions and concepts. The few outlier questions that are not associated with any concept are generally those questions with very low intrinsic difficulty or those questions with very few responses.
Figure 2.8: Concept 5 knowledge estimates generated by SPARFA-B for the STEMscopes data for a randomly selected subset of learners. The box-whisker plot shows the posterior variance of the MCMC samples, with each box-whisker plot corresponding to a different learner in the dataset. Anonymized learner IDs are shown on the bottom, while the number of relevant questions answered by each learner answered is indicated on the top of the plot.

One advantage of SPARFA-B over SPARFA-M is its ability to provide not only point estimates of the parameters of interest but also reliability information for those estimates. This reliability information can be useful for decision making, since it enables one to tailor actions according to the associated uncertainty. If there is considerable uncertainty regarding learner mastery of a particular concept, for example, it may be a more appropriate use of time of the learner to ask additional questions that reduce the uncertainty, rather than assigning new material for which the learner may not be adequately prepared.

We demonstrate the utility of SPARFA-B’s posterior distribution information on the learner concept knowledge matrix $C$. Fig. 2.8 shows box-whisker plots of the MCMC output samples over 30,000 iterations (after a burn-in period of 30,000 iterations) for a set of learners for Concept 5. Each box-whisker plot corresponds to the posterior distribution for a different learner. These plots enable us to visualize both the posterior mean and variance associated with the concept knowledge estimates $\hat{c}_j$. As one would expect, the estimation variance tends to decrease as the number of
answered questions increases (shown in the top portion of Fig. 2.8).

The exact set of questions answered by a learner also affects the posterior variance of our estimate, as different questions convey different levels of information regarding a learner’s concept mastery. An example of this phenomenon is observed by comparing Learners 7 and 28. Each of these two learners answered 20 questions and had a nearly equal number of correct answers (16 and 17, respectively). A conventional analysis that looked only at the percentage of correct answers would conclude that both learners have similar concept mastery. However, the actual set of questions answered by each learner is not the same, due to their respective instructors assigning different questions. While SPARFA-B finds a similar posterior mean for Learner 7 and Learner 28, it finds very different posterior variances, with considerably more variance for Learner 28. The SPARFA-B posterior samples shed additional light on the situation at hand. Most of the questions answered by Learner 28 are deemed easy (defined as having intrinsic difficulties \( \hat{\mu}_i \) larger than one). Moreover, the remaining, more difficult questions answered by Learner 28 show stronger affinity to concepts other than Concept 5. In contrast, roughly half of the questions answered by Learner 7 are deemed hard and all of these questions have stronger affinity to Concept 5. Thus, the questions answered by Learner 28 convey only weak information about the knowledge of Concept 5, while those answered by Learner 7 convey strong information. Thus, we cannot determine from Learner 28’s responses whether they have mastered Concept 5 well or not. Such SPARFA-B posterior data would enable a PLS to quickly assess this scenario and tailor the presentation of future questions to Learner 28—in this case, presenting more difficult questions related to Concept 5 would reduce the estimation variance on their concept knowledge and allow a PLS to better plan future educational tasks for this particular learner.

Second, we demonstrate the utility of SPARFA-B’s posterior distribution infor-
Table 2.3: Comparison of SPARFA-M and SPARFA-B for three selected questions and the $K = 5$ estimated concepts in the STEMscopes dataset. For SPARFA-M, the labels “Yes” and “No” indicate whether a particular concept was detected in the question. For SPARFA-B, we show the posterior inclusion probability (in percent), which indicates the percentage of iterations in which a particular concept was sampled.

<table>
<thead>
<tr>
<th>Question</th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>C5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q3 (27 responses)</td>
<td>M</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>B</td>
<td>94%</td>
<td>36%</td>
<td>48%</td>
<td>18%</td>
<td>80%</td>
</tr>
<tr>
<td>Q56 (5 responses)</td>
<td>M</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>B</td>
<td>0%</td>
<td>30%</td>
<td>26%</td>
<td>31%</td>
<td>31%</td>
</tr>
<tr>
<td>Q72 (6 responses)</td>
<td>M</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>B</td>
<td>61%</td>
<td>34%</td>
<td>29%</td>
<td>36%</td>
<td>58%</td>
</tr>
</tbody>
</table>

Accurate estimation of $\mathbf{W}$ enables course instructors and content authors to validate the extent to which problems measure knowledge across various concepts. In general, there is a strong degree of commonality between the results of SPARFA-M and SPARFA-B, especially as the number of learners answering a question grow. We present some illustrative examples of support estimation on $\mathbf{W}$ for both SPARFA algorithms in Tbl. 2.3. We use the labels “Yes”/“No” to indicate inclusion of a concept by SPARFA-M and show the posterior inclusion probabilities for each concept by SPARFA-B. Here, both SPARFA-M and SPARFA-B agree strongly on both Question 3 and Question 56. Question 72 is answered by only 6 learners, and SPARFA-M discovers a link between this question and Concept 5. SPARFA-B proposes Concept 5 in 58% of all MCMC iterations, but also Concept 1 in 60% of all MCMC iterations. Furthermore, the proposals of Concept 1 and Concept 5 are nearly mutually exclusive; in most iterations only one of the two concepts is proposed, but both are rarely proposed jointly. This behavior implies that SPARFA-B has found two competing models that explain the data associated with Question 72. To resolve this ambiguity, a PLS would need to gather more learner responses.
2.5.2.3 Algebra Test Administered on Amazon Mechanical Turk

For a final demonstration of the capabilities of SPARFA, we analyze a dataset from a high school algebra test carried on Amazon Mechanical Turk, a crowd-sourcing marketplace [40].

**Dataset** The dataset consists of $N = 99$ learners answering $Q = 34$ questions covering topics such as geometry, equation solving, and visualizing function graphs. Question labels are drawn from a set of $M = 10$ tags. The dataset is fully populated, with no missing entries.

**Analysis** We estimate $\mathbf{W}$, $\mathbf{C}$, and $\mu$ from the fully populated $34 \times 99$ binary-valued matrix $\mathbf{Y}$ assuming $K = 5$ concepts. We deploy the tag-analysis approach proposed in Sec. 2.4 to interpret each concept. Additionally, we calculate the likelihoods of the responses using (2.1) and the estimates $\hat{\mathbf{W}}$, $\hat{\mathbf{C}}$ and $\hat{\mu}$. The results from SPARFA are summarized in Fig. 2.9. We detail the results of our analysis for Questions 19–26 in Tbl. 2.4 and for Learner 1 in Tbl. 2.5.

**Results and discussion** With the aid of SPARFA, we can analyze the strengths and weaknesses of each learner’s concept knowledge both individually and relative to other users. We can also detect outlier responses that are due to guessing or carelessness. The values in the estimated concept knowledge matrix $\hat{\mathbf{C}}$ measure each learner’s concept knowledge relative to all other learners. The estimated intrinsic difficulties of the questions $\hat{\mu}$ provide a relative measure that summarizes how all users perform on each question.

Let us now consider an example in detail; see Tbl. 2.4 and Tbl. 2.5. Learner 1 incorrectly answered Questions 21 and 26 (see Tbl. 2.4), which involve Concepts 1 and 2. Their knowledge of these concepts is not heavily penalized, however (see
(a) Question–concept association graph.

<table>
<thead>
<tr>
<th>Concept 1</th>
<th>Concept 2</th>
<th>Concept 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fractions (57%)</td>
<td>Plotting functions (64%)</td>
<td>Geometry (63%)</td>
</tr>
<tr>
<td>Solving equations (42%)</td>
<td>System of equations (27%)</td>
<td>Simplifying expressions (27%)</td>
</tr>
<tr>
<td>Arithmetic (1%)</td>
<td>Simplifying expressions (9%)</td>
<td>Trigonometry (10%)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Concept 4</th>
<th>Concept 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simplifying expressions (64%)</td>
<td>Trigonometry (53%)</td>
</tr>
<tr>
<td>Trigonometry (21%)</td>
<td>Slope (40%)</td>
</tr>
<tr>
<td>Plotting Functions (15%)</td>
<td>Solving equations (7%)</td>
</tr>
</tbody>
</table>

(b) Most important tags and relative weights for the estimated concepts.

Figure 2.9: (a) Question–concept association graph and (b) most important tags associated with each concept for a high-school algebra test carried out on Amazon Mechanical Turk with $N = 99$ users answering $Q = 34$ questions.
Table 2.4: Graded responses and their underlying concepts for Learner 1 (1 designates a correct response and 0 an incorrect response).

<table>
<thead>
<tr>
<th>Question number</th>
<th>19</th>
<th>20</th>
<th>21</th>
<th>22</th>
<th>23</th>
<th>24</th>
<th>25</th>
<th>26</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learner’s graded response $Y_{i,j}$</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Correct answer likelihood $p(Y_{i,j} = 1</td>
<td>\bar{w}_i, c_j, \mu_i)$</td>
<td>0.79</td>
<td>0.71</td>
<td>0.11</td>
<td><strong>0.21</strong></td>
<td>0.93</td>
<td>0.23</td>
<td>0.43</td>
</tr>
<tr>
<td>Underlying concepts</td>
<td>1</td>
<td>1, 5</td>
<td>1</td>
<td>2, 3, 4</td>
<td>3, 5</td>
<td>2, 4</td>
<td>1, 4</td>
<td>2, 4</td>
</tr>
<tr>
<td>Intrinsic difficulty $\mu_i$</td>
<td>−1.42</td>
<td>−0.46</td>
<td>−0.67</td>
<td>0.27</td>
<td>0.79</td>
<td>0.56</td>
<td>1.40</td>
<td>−0.81</td>
</tr>
</tbody>
</table>

Tbl. 2.5), due to the high intrinsic difficulty of these two questions, which means that most other users also incorrectly answered them. User 1 also incorrectly answered Questions 24 and 25, which involve Concepts 2 and 4. Their knowledge of these concepts is penalized, due to the low intrinsic difficulty of these two questions, which means that most other users correctly answered them. Finally, Learner 1 correctly answered Questions 19 and 20, which involve Concepts 1 and 5. Their knowledge of these concepts is boosted, due to the high intrinsic difficulty of these two questions.

SPARFA can also be used to identify each user’s individual strengths and weaknesses. Continuing the example, Learner 1 needs to improve their knowledge of Concept 4 (associated with the tags “Simplifying expressions”, “Trigonometry,” and “Plotting functions”) significantly, while their deficiencies on Concepts 2 and 3 are relatively minor.

Finally, by investigating the likelihoods of the graded responses, we can detect outlier responses, which would enables a PLS to detect guessing and cheating. By inspecting the concept knowledge of Learner 1 in Tbl. 2.5, we can identify insufficient knowledge of Concept 4. Hence, Learner 1’s correct answer to Question 22 is likely due to a random guess, since the predicted likelihood of providing the correct answer is estimated at only 0.21.
Table 2.5: Estimated concept knowledge for Learner 1.

<table>
<thead>
<tr>
<th>Concept number</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Concept knowledge</td>
<td>0.46</td>
<td>−0.35</td>
<td>0.72</td>
<td>−1.67</td>
<td>0.61</td>
</tr>
</tbody>
</table>

2.5.3 Predicting Unobserved Learner Responses

Thus far, our real-world examples have been concerned with understanding the relationship between concepts, learners, and questions given the learner response data $Y$. Another goal of LA is to be able to predict future observations based on the outcome of the learner on other related questions. An advantage of the the Bayesian SPARFA framework is that we can impute missing data by introducing an additional sampling step to the MCMC procedure described in Sec. 2.3. By integrating out the latent variable $Z$ in (2.2) we can see that this sampling step is given by:

$$Y_{i,j} | c_j, \bar{w}_i, \mu_i \sim \text{Ber}(\bar{w}_i^T c_j + \mu_i), \forall i, j \in \Omega_{\text{obs}}^c. \quad (2.7)$$

There are several advantages to taking this approach to missing values. First, performing imputation from within the MCMC leads to superior performance as opposed to, say, basing imputation on the posterior means of each of the parameters. Secondly, the imputed values of $Y$ can then be used in the remainder of the MCMC steps as if they were observed data. This fully Bayesian approach to missing data is computationally advantageous. As a concrete example, the computation of the covariance matrix $M_j = (V^{-1} + \bar{W}^T \bar{W})^{-1}$ is taxing when each user answers a different set of questions. Under the missing value imputation scheme, however, this matrix is the same for each user (as is the case when $Y$ is fully observed) leading to easier computation.

We now consider the imputation accuracy of SPARFA-B against two other techniques for LA. The first is the CF-IRT approach of [41]. We additionally consider
<table>
<thead>
<tr>
<th>Method</th>
<th>MTurk</th>
<th>Sig. Proc.</th>
<th>STEMscopes</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPARFA-B</td>
<td>0.1939</td>
<td>0.1513</td>
<td>0.2046</td>
</tr>
<tr>
<td>CF-IRT</td>
<td>0.1951</td>
<td>0.1555</td>
<td>0.2144</td>
</tr>
<tr>
<td>Rasch</td>
<td>0.2029</td>
<td>0.1572</td>
<td>0.2108</td>
</tr>
</tbody>
</table>

Table 2.6: Imputation accuracy for SPARFA-B, CF-IRT, and the Bayesian Rasch model for three educational datasets. The SPARFA has the best performance in all cases.

a fully Bayesian variant of the Rasch model [14] assuming simple Normal priors on both the learner ability and question difficulty parameters. We will compare the performance of these methods across multiple datasets. For each experiments, we choose 20% of the entries of \( Y \) and remove them from the observed ground truth. We then deploy each of the three methods and estimate the missing values. We then compare against the ground truth and compute the fraction of mislabeled entries (lower values imply better performance, with 0 implying perfect recovery). We repeat this experiment over 50 randomized trials and display our results in Table 2.6, where we note that SPARFA achieves the best performance of all three methods. Most importantly, SPARFA does this while ensuring interpretability of the final result.
2.6 Conclusions

In this chapter, we have formulated the SPARFA model for learning and content analytics, which is based on a new statistical model that encodes the probability that a learner will answer a given question correctly in terms of three factors: (i) the learner’s knowledge of a set of latent concepts, (ii) how the question related to each concept, and (iii) the intrinsic difficulty of the question. We have further proposed the SPARFA-B algorithm to estimate the above three factors given incomplete observations of graded learner question responses. We have also introduced a novel method for incorporating user-defined tags on questions to facilitate the interpretability of the estimated factors. Experiments with both synthetic and real world education datasets have demonstrated both the efficacy and robustness of the SPARFA model as well as the utility of SPARFA-B over both its optimization-based counterpart and other state-of-the-art methods for LA.

The quantities estimated by SPARFA can be used directly in a range of PLS functions. For instance, we can identify the knowledge level of learners on particular concepts and diagnose why a given learner has incorrectly answered a particular question or type of question. Moreover, we can discover the hidden relationships among questions and latent concepts, which is useful for identifying questions that do and do not aid in measuring a learner’s conceptual knowledge. Outlier responses that are either due to guessing or cheating can also be detected. In concert, these functions can enable a PLS to generate personalized feedback and recommendation of study materials, thereby enhancing overall learning efficiency.

Before closing, we would like to point out a connection between SPARFA and dictionary learning that is of independent interest. This connection can be seen by noting that (2.2) for both the probit and inverse logit functions is statistically
equivalent to (see [23]):

\[ Y_{i,j} = \text{sign}(W_C + M + N)_{i,j}, \quad (i, j) \in \Omega_{\text{obs}}, \]

where \text{sign}(\cdot) denotes the entry-wise sign function and the entries of \( N \) are i.i.d. and drawn from either a standard Gaussian or standard logistic distribution. Hence, estimating \( W, C, \) and \( M \) (or equivalently, \( \mu \)) is equivalent to learning a (possibly overcomplete) dictionary from the data \( Y \). The key departures from the dictionary-learning literature [36,42] and algorithm variants capable of handling missing observations [43] are the binary-valued observations and the non-negativity constraint on \( W \).

Note that the proposed SPARFA algorithms can be applied to a wide range of applications beyond education, including the analysis of survey data, voting patterns, gene expression, and signal recovery from noisy 1-bit compressive measurements [44–46].
Chapter 3

Extensions of SPARFA
3.1 Introduction

In the previous chapter, we introduced and developed the SPARFA model for LA as well as the SPARFA-B algorithm that fits the SPARFA model to a learner response data matrix $Y$. We demonstrated the efficacy of this model on both synthetic and real-world educational data. In particular, we showed that SPARFA outperformed other state-of-the-art methods for LA.

We note that the model presented in Chapter 2 possesses some number of limitations. First, it was developed only for the case of binary data. While we noted that extensions to real-valued data were trivial, it is less clear how to extend SPARFA for use with other common data types.

Secondly, the model proposed previously assumed that learner consistency was homoskedastic. Concretely, we assumed that, conditioned on the latent variable $Z_{i,j}$:

$$Y_{i,j} \sim Ber(\Phi(Z_{i,j})), \quad (i, j) \in \Omega_{\text{obs}}. \quad (3.1)$$

In this model, each learner is assumed to have the same reliability. To see this more clearly, we can rewire the second line of (3.1) as:

$$P(Y_{i,j} = 1) = P(\Phi(Z_{i,j} + \varepsilon_{i,j}) > 0), \varepsilon \sim \mathcal{N}(0,1) \quad (3.2)$$

The models of (3.1) and (3.2) can be shown to be equivalent (see, for example, [27] for details). The second representation, however, shows that each has an expected performance defined by $Z_{i,j}$ and equal noise variance. In real educational scenarios, we would expect different learners to possess different degrees of internal reliability. By modeling this variability directly, we can hope to provide superior modeling for each learner.

Finally, we assumed that the number of latent concepts in the educational dataset
is known and fixed to a specific value $K$. While it may be quite reasonable to assume this values in some scenarios, and possible to learn it via cross validation in others, this is a limitation on the flexibility of the SPARFA model.

In this chapter, we propose extensions to the SPARFA model to address each of these limitations. First, this extended model can be used for the case of ordinal (ordered categorical) data $Y$. Second, it employs a heteroskedastic model on learner reliability that allows to extract more information about a learner given their responses to questions. Finally, it employs a non-parametric prior based on the Indian Buffet Process (IBP) [47–49] on the number of latent concepts $K$ such that we can sample the number of latent concepts directly as part of the MCMC. Each of these extensions is fully Bayesian, again enabling us to learn full posterior distributions over each parameter of interest. These extensions to SPARFA were first proposed in [50].

The remainder of this chapter is organized as follows. Details regarding the fully Bayesian model and prior distributions are given in Sec. 3.2. Sec. 3.3 presents our method for performing posterior inference and analysis. Sec. 3.4 presents a simulation study and results from experimental data. We conclude in Sec. 3.5.
3.2 Bayesian Model

We start by presenting the latent factor multinomial probit model. We also discuss prior distributions for all of the parameters and the imputation of missing data.

3.2.1 Latent factor probit model

Consider ordinal data from several subjects on a number of variables. For illustration, we investigate graded answers to a number of assessment items (questions) by a number of learners. A common approach to model such data is multinomial probit regression, where probability of success is captured through the use of the normal cumulative distribution function. As before, denote the response variable, \( Y_{ij} \), for subject (learner) \( j = 1, \ldots, N \) on variable (question) \( i = 1, \ldots, Q \). Initially, let us examine the binary heteroskedastic case, where \( Y_{ij} \) can take values 0 or 1. Using data augmentation [51,52] we introduce a latent, continuous random variable, \( Z_{i,j} \), of individual \( j \) for variable \( j \) such that \( Y_{ij} = 1 \) if \( Z_{i,j} > 0 \), and 0 otherwise. Under the probit model, we assume that

\[
P(Y_{ij} = 1) = \Phi(Z_{i,j}; 0, \psi_j^{-1}),
\]

where the inverse probit function includes a subject-specific variance, \( \psi_j^{-1} \).

As before, we have the relationship:

\[
Z_{i,j} = \mathbf{w}_i^T \mathbf{c}_i + \mu_j, \quad \forall i, j.
\]

In the general setting, the latent factor probit regression model can handle ordered, polychotomous data. Here, the response, \( Y_{i,j} \), takes one of \( D \) values, coded as 1, \ldots, \( D \).
Then, we consider a latent variable $Z_{i,j}$ and posit that

$$Y_{i,j} = d \text{ if } Z_{i,j} \in (\xi_{d-1}, \xi_d],$$

(3.5)

where $\{\xi_0, \ldots, \xi_D\}$ is an ordered set of real valued cutoff points, $-\infty = \xi_0 < \xi_1 < \xi_2 < \cdots < \xi_{D-1} < \xi_D = \infty$. The standard probit model utilized throughout Chapter 2 is a simplification of (3.5) with $D = 2$ and $\xi_1 = 0$.

### 3.2.2 Infinite factor models via the Indian Buffet Process

Previously, we fixed the number of latent factors to $K$. We now detail how to use a Bayesian non-parametric prior such that $K$ can be estimated directly from the data. This prior will further allow us to maintain the sparsity and non-negativity properties that are fundamental to SPARFA.

We follow the approach of [49] to develop a nonparametric infinite factor model and break the latent features matrix into a binary matrix $H$ indicating which concepts are present for each variable and a second matrix $W$ providing the value of the association between factor and variable. That is, we enforce $W = H \odot W$, where $\odot$ denotes the Hadamard (element-wise) matrix product. Given a truncated normal prior for the non-zero elements of $W$, we can represent this relationship as

$$W_{i,k} \sim H_{i,k} \mathcal{N}^+(0, \tau_k^{-1}) + (1 - H_{i,k}) \delta_0,$$

where $\mathcal{N}^+(0, \tau_k^{-1})$ is a normal distribution with mean 0 and factor-specific precision $\tau_k$ truncated below at 0 and $\delta_0$ is a point mass at 0. Naturally, we could use the exponential prior $\text{Exp}(\lambda_k)$ as well.

As $H$ is unknown, it requires a prior distribution. We allow the data to determine the number of factors and feature selection via the Indian Buffet process (IBP).
The IBP is a stochastic process defining a probability distribution over sparse binary matrices with a finite number of rows (here, $Q$) and an unbounded number of columns [47, 53]. This prior provides a means to learn the binary matrix without fixing the number of factors.

Assume we have a finite number of columns, $K$. We say that feature $k$ affects the $i^{th}$ row of $H$ if $H_{i,k} = 1$. Each dimension includes feature $k$ independently with probability $\pi_k$, and can include multiple features. We place a Bernoulli distribution on each $H_{i,k}$

$$p(H | \pi) = \prod_{k=1}^{K} \prod_{i=1}^{Q} p(H_{i,k} | \pi_k) = \prod_{k=1}^{K} \pi_k^{m_k} (1 - \pi_k)^{Q - m_k},$$

with $m_k = \sum_{j=1}^{Q} h_{i,k}$, the number of rows influenced by the $k^{th}$ factor. We then define a beta prior on the $\pi_k$,

$$\pi_k \mid \alpha \sim \text{Beta} \left( \frac{\alpha}{K}, 1 \right). \quad (3.6)$$

Marginalizing over the $\pi_k$ and taking the limit of $K \to \infty$, we obtain

$$p(H | \alpha) = \frac{\alpha^{K+} e^{-\alpha H_Q} K^+ \prod_{k=1}^{K} (Q - m_k)! (m_k - 1)!}{\prod_{h=1}^{2D-1} K_b! Q!},$$

where $H_Q = \sum_{i=1}^{Q} \frac{1}{i}$ is the $Q^{th}$ harmonic number, $K_+$ is the number of columns where $m_k > 0$, and $K_b$ is the number of columns with pattern $b$.

An alternative representation of the IBP is characterized via a theoretical buffet with a possibly infinite number of dishes. The first customer chooses a number of dishes according to a Poisson($\alpha$). The $i^{th}$ subsequent customer samples previously sampled dishes with probability $m_k / i$, where $m_k$ is the number of customers who have already sampled dish $k$. Then he considers new dishes according to a Poisson($\alpha / i$).
Looking at the last customer, the probability \( H_{i,k} = 1 \) given \( H_{-i,k} \) is \( m_{-i,k}/Q \), where \( m_{-i,k} = \sum_{s \neq i} H_{s,k} \). Thus, the parameter \( \alpha \) in (3.6) controls the number of features per dimension as well as the total number of features.

### 3.2.3 Prior distributions for model parameters

We complete the specifications of the model by assuming computationally convenient prior distributions on the remaining parameters of interest. The model can then be fully summarized as follows

\[
P(Y_{i,j} = 1) = \Phi(Z_{i,j}; 0, \psi_j^{-1}),
\]

\[
\xi \propto 1_{\{\xi \in \mathcal{D}\}}, \quad \mathcal{D} = \{-\infty = \xi_0 < \xi_1 < \cdots < \xi_{D-1} < \xi_D = \infty\},
\]

\[
\psi_j \sim Ga(a_\psi, b_\psi),
\]

\[
c_j \sim \mathcal{N}(0, V),
\]

\[
V \sim IW(V_0, \nu),
\]

\[
\mu_i \sim \mathcal{N}(m_0, v_\mu),
\]

\[
\tau_k \sim Ga(a_\tau, b_\tau),
\]

\[
W_{i,k} | \tau_k, H_{i,k} \sim H_{i,k} \mathcal{N}_+^+(0, \tau_k^{-1}) + (1 - H_{i,k}) \delta_0,
\]

\[
\mathbf{H} \sim IBP(\alpha),
\]

\[
\alpha \sim Ga(a_\alpha, b_\alpha),
\]

for all \( j = 1, \ldots, N \) and \( i = 1, \ldots, Q \). Here, \((a_\psi, b_\psi, V_0, \nu, m_0, v_\mu, a_\tau, b_\tau, a_\alpha, b_\alpha)\) are fixed hyperparameters.
3.2.4 Imputing Missing values

As was the case with the parametric SPARFA-B algorithm, we can handle missing values through Bayesian variable imputation [54,55] as ignoring the unobserved data can lead to potential biases. Instead, we take a random imputation approach to handle the missing data and incorporate our uncertainty about the unobserved data. In essence, we estimate the probable responses for a given learner’s missing values based on the answers we do observe. By doing this, we avoid complex estimation algorithms because, conditioned on our estimated responses, $Y$ is now considered to be completely observed.
3.3 Posterior Inference

3.3.1 Markov Chain Monte Carlo algorithm

We employ a Markov Chain Monte Carlo (MCMC) algorithm to obtain samples from the joint posterior distribution of the model parameters. We outline the algorithm below; details of the full sampling procedure are found in App. B.

For each iteration:

1. Impute any missing values in $Y$.
2. Update $Z$ from the truncated normal full conditional.
3. Update $\mu$ from the normal full conditional.
4. For each $i$ in $1, \ldots, Q$
   (a) Update each $(H_{i,k}, W_{i,k}), k = 1, \ldots, K$, marginally for $H_{i,k}$ then $W_{i,k} | H_{i,k}$.
   (b) Propose the addition of $k_i$ new factors with a Metropolis-Hastings step.
5. Update each $c_j$ from the $K$-variate normal full conditional.
6. Propose new cutoff values $\xi$ through a Metropolis-Hastings step, if applicable.
7. Update the precision parameters $\{\tau_k\}$ and $\{\psi_j\}$ and the IBP parameter $\alpha$ from their respective gamma full conditionals.

3.3.2 Identifiability

It is well known that both ordinal data and factor analysis models suffer from several identifiability issues $[56, 57]$. First, identifiability problems arise under certain scalings and shiftings of the latent parameters. In our method, for example, one can shift
the cutoff positions $\xi$ by some constant while simultaneously shifting the intercept parameters $\mu$ by the same constant without affecting the overall likelihood. Additionally, one can arbitrarily scale the factor loadings $W$ while inversely scaling the factor scores $C$ by the same amount. We follow the lead of [56] and mitigate many of these difficulties by imposing proper priors on the latent factors as well constraining the first cutoff position $\xi_1$ to 0. We additionally constrain the first user precision $\psi_1$ to 1.

A more serious concern in many applied contexts is that factor analysis models are unidentifiable under any permutation of the latent factors [57]. Concretely, one can jointly permute the factors of $W$ and $C$ without affecting the overall likelihood. This is commonly referred to as “label-switching”. If not mitigated properly, the label switching problem can severely complicate posterior analysis. Some prior work [58] exists on overcoming the label switching problem for factor models that involve operating directly on the factors during the MCMC execution. We have found, however, that these techniques are unsatisfactory for our application. Instead, we recommend a processing step performed after running the MCMC to mitigate label switching. This step is inspired, in part, by the mixture model literature [59]. Let $W^t$, $C^t$, and $\mu^t$ denote the $t^{th}$ samples from the MCMC. We first compute the posterior probability $p(W|W^t, C^t, \mu^t)$ and then select the iteration $t_{\text{max}}$ that maximizes this probability. We then permute the factors $W^t, C^t$ obtained over all iterations $t \neq t_{\text{max}}$ to best match $W^{t_{\text{max}}}, C^{t_{\text{max}}}$. Performing this step essentially aligns the posterior samples to a common reference, enabling more meaningful posterior analysis (such as the computation of posterior means).
3.3.3 Posterior analysis

At each iteration of the algorithm, the number of active features, say $K^+$, can change. For comparable results, we base posterior inferences only on output from iterations where $K^+$ is equal to the posterior mode. Given the draws from the joint posterior distributions of all parameters with $K^+$ fixed at the posterior mode, we obtain inference for many objectives of interest.

For example, estimates for the mean and quantiles of each cell in $H$, $W$, and $C$ are easily calculated. The posterior mean of $H$ provides estimates of the posterior probability of inclusion of the particular dimension for any given factor. Consequently, we estimate the factor loadings in $W$ by thresholding the posterior probability of inclusion. If the probability of factor $k$ for question $i$ is less than a specified value, the corresponding $W_{i,k}$ is set to 0, otherwise the posterior mean is used.
3.4 Applications

Here we assess the performance of non-parametric SPARFA on simulated data as well as on three real-world educational data sets.

3.4.1 Simulation study

We first use simulated data sets generated under various settings. Our experiments proceed by generating data, running our MCMC method to estimate posterior distributions for all model parameters, and then evaluating their fidelity with respect to the known ground truth. Concretely, we generate model parameters according to (3.7), and synthetic data $Y$ according to (3.4) and (3.5). We then estimate posterior distributions for all model parameters using the MCMC method described in Sec. 3.3.1 and post-process our factors as described in Sec. 3.3.2 to alleviate the label switching problem. Following the post-processing step, we compute the posterior means for all model parameters as detailed in Section 3.3.3. Performance is measured using the normalized Frobenius loss first defined in Sec. 2.5.1.3 and also used in [60]. Restated concisely, for an arbitrary matrix $S$ and estimate $\hat{S}$, our metric is defined as

$$E_S = \frac{||S - \hat{S}||_F^2}{||S||_F^2}. \quad (3.8)$$

We note that, in all of our synthetic experiments, the posterior mode of $K$ always matched the known ground truth.

3.4.1.1 Accuracy vs. size

We first measure performance as a function of the size of the input data, $Y$. We first consider the binary ($D = 2$) case. We consider fully observed $Y$ and problems of size $Q \times N$, with $Q = N = N_0 \in \{50, 75, 100\}$, and with $K = 3$. We compare our method
Table 3.1: Synthetic data, accuracy vs size for our method (IBP) and a parametric (Para) method for binary, homoskedastic data with $K = 3$ latent factors. Results show the mean (std. err.) for each metric over 50 trials.

<table>
<thead>
<tr>
<th>$N_0$</th>
<th>$E_W$</th>
<th>$E_C$</th>
<th>$E_\mu$</th>
</tr>
</thead>
<tbody>
<tr>
<td>50 (IBP)</td>
<td>0.27 (0.11)</td>
<td>0.32 (0.13)</td>
<td>0.09 (0.02)</td>
</tr>
<tr>
<td>50 (Para)</td>
<td>0.22 (0.1)</td>
<td>0.29 (0.13)</td>
<td>0.10 (0.03)</td>
</tr>
<tr>
<td>75 (IBP)</td>
<td>0.12 (0.03)</td>
<td>0.16 (0.03)</td>
<td>0.06 (0.02)</td>
</tr>
<tr>
<td>75 (Para)</td>
<td>0.11 (0.02)</td>
<td>0.15 (0.03)</td>
<td>0.06 (0.02)</td>
</tr>
<tr>
<td>100 (IBP)</td>
<td>0.08 (0.02)</td>
<td>0.12 (0.02)</td>
<td>0.05 (0.01)</td>
</tr>
<tr>
<td>100 (Para)</td>
<td>0.08 (0.01)</td>
<td>0.12 (0.02)</td>
<td>0.05 (0.01)</td>
</tr>
</tbody>
</table>

Table 3.2: Synthetic data, accuracy vs size for heteroskedastic ordinal data with $C = 5$ and $K = 5$ latent factors. Results show the mean (std. err.) for each metric over 50 trials.

<table>
<thead>
<tr>
<th>$N_0$</th>
<th>$E_W$</th>
<th>$E_C$</th>
<th>$E_\mu$</th>
<th>$E_\psi$</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>0.66 (0.10)</td>
<td>1.21 (0.17)</td>
<td>0.127 (0.04)</td>
<td>0.21 (0.06)</td>
</tr>
<tr>
<td>75</td>
<td>0.27 (0.14)</td>
<td>0.50 (0.26)</td>
<td>0.07 (0.02)</td>
<td>0.13 (0.04)</td>
</tr>
<tr>
<td>100</td>
<td>0.07 (0.01)</td>
<td>0.13 (0.02)</td>
<td>0.06 (0.02)</td>
<td>0.08 (0.02)</td>
</tr>
</tbody>
</table>

against the parametric SPARFA method, with $K$ set to the true value in order to give it the best possible performance. As the parametric model was developed for only the homoskedastic case, we set $\psi_j = 1$ $\forall j$ when generating our data. For each data size, we run 50 Monte Carlo trials and compute the accuracy metric (3.8) for $W$, $C$, and $\mu$. We compile the mean and standard deviation of the error metrics across all trials and display our results in Table 3.1. Here we see that our method, despite additionally having to learn the true value of $K$, produces results which are comparable to parametric SPARFA.

We next consider a more challenging setting with heteroskedastic ordinal data. Here, we generate data with $D = 5$ categories and with $K = 5$ latent factors. We generate $\psi_j$ according to (4.1) with $\alpha_\psi = \beta_\psi = 5$ $\forall j$. Our results are taken over 50 trials and are displayed in Table 3.2. Here, the complexity of the data makes recovery difficult for the smallest data set, though performance improves dramatically with only small increases in the size of the data.
Table 3.3: Synthetic data, accuracy vs percent observed data for the binary data case with $K = 3$. Results show the mean (std. err.) for each metric over 50 trials.

<table>
<thead>
<tr>
<th>$p_{\text{obs}}$</th>
<th>$E_{\mathbf{W}}$</th>
<th>$E_{C}$</th>
<th>$E_{\mu}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>0.75 (0.09)</td>
<td>0.94 (0.09)</td>
<td>0.13 (0.02)</td>
</tr>
<tr>
<td>0.6</td>
<td>0.47 (0.11)</td>
<td>0.60 (0.17)</td>
<td>0.10 (0.02)</td>
</tr>
<tr>
<td>0.7</td>
<td>0.25 (0.08)</td>
<td>0.30 (0.12)</td>
<td>0.09 (0.02)</td>
</tr>
<tr>
<td>0.8</td>
<td>0.14 (0.04)</td>
<td>0.17 (0.03)</td>
<td>0.07 (0.01)</td>
</tr>
<tr>
<td>0.9</td>
<td>0.10 (0.02)</td>
<td>0.13 (0.01)</td>
<td>0.06 (0.01)</td>
</tr>
</tbody>
</table>

3.4.1.2 Accuracy vs. missing Data

We now measure performance in the case where $\mathbf{Y}$ is only partially observed. Concretely, we randomly remove entries in $\mathbf{Y}$ with probability $1 - p_{\text{obs}}$ ($p_{\text{obs}} = 1$ corresponds to fully observed data). We then impute these missing values via the MCMC as described in Sec. 3.3.1.

For brevity, we examine only the binary homoskedastic case with $N = Q = 100, K = 3$. We vary $p_{\text{obs}} \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$. For each value of $p_{\text{obs}}$ we run 50 Monte Carlo trials and compute the accuracy metrics (3.8) for $\mathbf{W}, \mathbf{C}$ and $\mu$. The results are displayed in Table 3.3, where we note that we still have acceptable estimation of the true model with only 70\% of $\mathbf{Y}$ observed.

We further examine the predictive power of the fully Bayesian imputation of missing data. To do this, we calculate the posterior mode for all of the missing entries in $\mathbf{Y}$, compare against the true label, and compute the percent of entries correctly labeled. A summary of this metric is presented in Fig. 3.1. Here, we still achieve excellent predictive performance for missing entries even with only 60\% of the data is available.

3.4.1.3 Accuracy vs. user precision

As a final synthetic experiment we consider the estimation accuracy as a function of the heteroskedastic precision term $\psi_j$. To evaluate this, we fix $\alpha_0 = \alpha_{\psi} = \beta_{\psi}$ and vary
Figure 3.1: Synthetic data, predictive accuracy across different observation probabilities.
Table 3.4: Synthetic data, accuracy vs user precision. Results show the mean (std. err.) for each metric over 50 trials.

<table>
<thead>
<tr>
<th>$\alpha_0$</th>
<th>$E_W$</th>
<th>$E_C$</th>
<th>$E_\mu$</th>
<th>$E_\psi$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.27 (0.12)</td>
<td>0.48 (0.22)</td>
<td>0.08 (0.02)</td>
<td>0.20 (0.07)</td>
</tr>
<tr>
<td>2</td>
<td>0.23 (0.10)</td>
<td>0.40 (0.21)</td>
<td>0.07 (0.02)</td>
<td>0.16 (0.04)</td>
</tr>
<tr>
<td>5</td>
<td>0.15 (0.08)</td>
<td>0.30 (0.12)</td>
<td>0.09 (0.02)</td>
<td>0.12 (0.02)</td>
</tr>
<tr>
<td>100</td>
<td>0.08 (0.03)</td>
<td>0.13 (0.02)</td>
<td>0.05 (0.02)</td>
<td>0.05 (0.01)</td>
</tr>
</tbody>
</table>

$\alpha_0 \in \{1, 2, 5, 100\}$. As $\alpha_0$ grows larger, the data behaves more like a homoskedastic system. We run 50 Monte Carlo trials and compute the accuracy metrics (3.8) for $W, C, \mu$, and $\psi$. The results are displayed in Table 3.4. Here we see that the algorithm still yields reasonable results even for the cases of small $\alpha_0$, where the user precision can be quite low.

### 3.4.2 Educational data

We now turn to real educational data for learning and content analytics. In each case, we examine the factors estimated by our method and what these factors reveal about the data.

#### 3.4.2.1 Mechanical Turk math test

We first examine the Amazon Mechanical Turk exam first introduced in Sec. 2.5.2.3 in the binary case. The data consists of 99 participants responding to 34 questions. Responses are now graded on an ordinal scale from 1 to 4 by a single grader. Each participant responded to all questions, and so there is no missing data. We apply our method to this data set with relatively broad priors and infer posterior distributions for all factors of interest. We display histograms of the posterior of $K$ as well as the means of the $\psi_j$ variables in Figure 3.2. Our method finds that $K = 5$ concepts are sufficient to achieve a good fit to the data. Participant reliability for this data possesses many modes, essentially clustering learners into different reliability clusters.
Such information is valuable to educators for evaluating learner competence.

Next, we examine the posterior distribution of $\mathbf{W}$. In Figure 3.3 we graphically display both the posterior inclusion probability of $\mathbf{H}$ for the factor loadings along with the posterior mean of $\mathbf{W}$ computed via the post-processing step described in Sec. 3.3.3. We additionally display the post-processed $\mathbf{W}$ matrix as a bi-partite graph linking questions to concepts in Figure 3.4, with thicker lines denoting stronger links. Here, the five concepts are all well represented in the graph. We note that a small number of the questions given in the exam are not connected to any latent concept. The majority of these questions were either answered correctly or incorrectly by all participants, making it impossible to learn any meaningful structure. We further note that the first concept connects to most questions in the dataset. This concept can be best explained as relating to the basic mathematical nature of all questions on the test. Each test-taker, in turn, has a intrinsic latent ability in general mathematics that governs a large part of their success on all questions. The remaining concepts relate to more nuanced aspects of the exam – such as factorizing a polynomial or differentiating a function.

### 3.4.2.2 STEMscopes earth science

Next we examine the STEMscopes dataset first introduced in Sec. 2.5.2.2. The data consists of 85 learners from a single school district answering 145 questions. The data is graded on a binary (right/wrong) scale. This data set is especially challenging as it is highly incomplete; only 13.5% of the total question-answer pairs are observed. This is due to instructors being free to assign whichever problems they wished to a particular learner, leading to an overall lack of uniformity across the district.

We estimate posterior distributions for all parameters of interest using our method. Histograms showing the posterior distribution of $K$ and posterior means of $\psi_j$ are
Figure 3.2: Mechanical Turk results. Left: Histogram of $K$. Right: Histogram of the posterior mean of $\psi_j$.

Figure 3.3: Mechanical Turk math test results. Left: Graphical representation of the posterior mean of $H$. Right: Graphical representation of the posterior mean of $W$. 
Figure 3.4: Mechanical Turk results. Bipartite graph of the inferred $W$ matrix. Circles denote concepts while squares denote questions. Links denote relations between questions and concepts, with thicker lines denoting stronger links. The value inside each square denotes the intrinsic difficulty parameter $\mu_i$. 
shown in Fig. 3.5. Our method generally infers between between $K = 4$ and $K = 5$ concepts. Learner reliability is unimodal, with the vast majority of learners falling into a large mode centered at $\psi_j \leq 1$. We show the connectivity graph of $W$ in Fig. 3.6, where questions determined to be unconnected to any latent concept have been removed from the graph. The majority of the unconnected questions in this example are those for which we have too few responses to reliably estimate any latent structure. There are, however, five instances of questions with both a large number of responses as well significant diversity in the answer pattern. Here, our method declares that the dynamics of learner-response patterns for these questions are controlled only by the intrinsic difficulty. Our approach enables the automatic flagging of such questions, which can then be reviewed by educational content authors.

Because our data is binary, we can perform a final experiment and compare the predictive performance of our algorithm against the parametric SPARFA method. We do this by randomly removing 20% of the available entries of $Y$ and then applying our method to jointly learn the latent model factors while imputing the missing entries as detailed in Section 3.2.4. We repeat this experiment 50 times, selecting a new set of entries to remove for each experiment.

Let $Y_{i,j}^t$ denote the sample at iteration $t$ for all $(i, j) \notin \Omega_{\text{obs}}$. We use the following two metrics to evaluate performance

$$\ell_{\text{mode}} = E_{i,j} \left[ E_t[Y_{i,j}^t] - Y_{i,j} \right], \quad \ell_2 = E_{i,j} \left[ (E_t[Y_{i,j}^t] - Y_{i,j})^2 \right].$$

The first metric corresponds to the expected loss using the posterior mode of the estimates, whereas the second corresponds to the squared loss. We display the mean and standard deviation of these metrics in Table 3.5. We see that our method, despite having to contend with learning the true number of latent factors from the data, had nearly equal performance with respect to the parametric model given the optimal
Figure 3.5: STEMscopes results. Left: Histogram for the posterior values of $K$. Right: Histogram of the posterior mean of the $\psi_j$ terms.

Table 3.5: Real world data, imputation accuracy for non-parametric (IBP) SPARFA and parametric (Para) SPARFA. Results show the mean (std. err.) for each metric over 50 trials.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>$\ell_{\text{mode}}$ (std. err.)</th>
<th>$\ell_2$ (std. err.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>STEM (IBP)</td>
<td>0.2019 (0.02)</td>
<td>0.1540 (0.01)</td>
</tr>
<tr>
<td>STEM (Para)</td>
<td>0.2046 (0.02)</td>
<td>0.1497 (0.01)</td>
</tr>
<tr>
<td>Sig. Proc. (IBP)</td>
<td>0.1720 (0.03)</td>
<td>0.1282 (0.02)</td>
</tr>
<tr>
<td>Sig. Proc. (Para)</td>
<td>0.1567 (0.03)</td>
<td>0.1119 (0.02)</td>
</tr>
</tbody>
</table>

3.4.2.3 Undergraduate signal processing course

Last, we examine data obtained from a final exam given as part of undergraduate signal processing course, which was first introduced in Sec. 2.5.2.1. The dataset consists of 15 learners answering 44 questions. Each learner answered all questions, and so there is no missing data.

We infer all latent factors using our non-parametric MCMC approach with rela-
Figure 3.6: STEMscopes Results. Bipartite graph of the inferred $W$ matrix. Circles denote concepts while squares denote questions. Links denote relations between questions and concepts, with thicker lines denoting stronger links. The value inside each square denotes the intrinsic difficulty parameter $\mu_i$. 
Figure 3.7: Signal processing course results. Left: Histogram for the posterior values of $K$. Right: Histogram of the posterior mean of the $\psi_j$ terms.

tively vague priors. Histograms of the inferred value of $K$ and of the posterior means of $\psi_j$ are shown in Fig. 3.7. Originally, we assumed $K = 5$ latent concepts for this data set [21]. In contrast, our method finds a much more parsimonious model requiring only $K = 2$ latent concepts. Learner precisions for this data set are bimodal— with roughly half of the learners clustering as “less reliable” ($\psi_j < 1$) while the other half show stronger reliability ($\psi_j \geq 1$). A graphical representation of the inferred $W$ matrix is displayed in Figure 3.8, where we note that for this data set one concept tends to dominate most of the questions, with the second concept appearing more sparsely. This is likely due to the small size of the data set.

We again perform a data imputation comparison between our method and the parametric SPARFA method. As before, we randomly delete 20% of the entries in $Y$ and impute the missing values as part of the MCMC. A comparison of the imputation performance of the two methods is shown in Table 3.5, where we note that our method performs nearly as well as the parametric method tuned to the optimal value of $K$. 
Figure 3.8: Signal processing course results. Bipartite graph of the inferred $W$ matrix. Circles denote concepts while squares denote questions. Links denote relations between questions and concepts, with thicker lines denoting stronger links. The value inside each square denotes the intrinsic difficulty parameter $\mu_i$. 
3.5 Conclusions

We have proposed a non-parametric, heteroskedastic, probit variant of SPARFA. This model enables a number of important tasks related to personalized learning, such as automatically inferring the number of latent concepts in an educational data set, learning associations between questions and latent concepts in a course, estimating the level of mastery that each learner has attained with respect to each concept, and modeling the reliability of each learner. We have demonstrated the utility of our approach on both synthetic and real-world educational datasets and shown that performance is comparable to parametric SPARFA despite the fact that our model must contend with learning additional parameters.
Chapter 4

Learning Analytics Model with Automated Learner Clustering
4.1 Introduction

We now examine a variant of the SPARFA algorithm in which learners are automatically clustered by their similarity to one another. This clustering has powerful implications for education. First, explicitly clustering similar learners allows us to “borrow strength” from other learners in making statistical inferences about another learner. Second, this clustering can be useful to course instructors or a fully automated PLS to understand these similarities when making instructional decisions. As a concrete example, in a given class there may be learners who struggle with the same concept. By clustering these learners, an instructor (or PLS) may be able to provide the same diagnosis and remediation for the groups conceptual misunderstanding [61]. This clustering could be useful are in recommending study groups or understanding different ways in which learners can fail to master certain material.

We additionally note that automatic clustering of users is a very important problem that finds applications in fields such as medicine [62], psychology [63], and genetics [64].

In this chapter we derive a statistical model and algorithm for clustered SPARFA. This model employs two non-parametric priors that warrant superior modeling flexibility, interpretability, and applicability. First, our model employs a beta-bernoulli process to learn the number of latent variables from the data itself [47, 53] as in Chapter 3. As before, this allows our model to avoid problems with over/under fitting common in factor analysis. Second, our model employs a Dirichlet Process (DP) prior that enables us to automatically cluster similar users and adapt the number of groups as required by the data.

The remainder of this chapter is organized as follows. In Sec. 4.2 we develop our statistical model for our non-parametric clustering factor model. In Sec. 4.3 we show how to fit our model using Markov Chain Monte-Carlo (MCMC) methods.
showcase the performance of our method on both synthetic and real-world educational data in Sec. 4.4. We conclude in Sec. 4.5.
4.2 Model

Grouping the users with similar latent factors provides insight on the anatomy of the population. We assume the latent factors arise from a mixture of normal densities. We may consider a finite mixture of $L$ normals, where each component has a $K$-variate normal distribution with mean $\theta_l$ and covariance matrix $I_K$:

$$p(c_j \mid \theta) = \sum_{l=1}^{L} \pi_l N_K(c_j \mid \theta_l, I_K),$$

with $\{\pi_l\}$ having a conjugate Dirichlet prior. One major assumption for this model is that each vector of latent factors arises from one of the $L$ mixture components, which has a distinct mean to capture the distribution of the factors assigned to it. However, the choice of the number of distinct components is not necessarily apparent.

The possibly infinite dimensional models involving Dirichlet process (DP) priors [65] are the most widely used alternative to finite mixture models.

There are several definitions of the Dirichlet process. For illustrative purposes, we explore the Chinese restaurant process definition, which highlights the clustering mechanism of the process. The Chinese restaurant model is based on idea that there is a restaurant with an infinite number of tables. At each table there are an infinite number of seats. The first customer arrives and sits down at an open table. The second customer then arrives and selects the table that the first customer is currently sitting with probability $\frac{1}{\beta+1}$ or selects a new table with probability $\frac{\beta}{\beta+1}$. This continues on to the $i^{th}$ customer, who selects a table that a current customer is sitting with probability $\frac{n_m}{\beta+i-1}$, where $n_m$ is the number of customers at table $m$. For each table, model parameters are generated from a base distribution $G_0$.

We place a DP prior on the distribution of the $c_j$, with a $K$-variate normal as the base distribution. That is, each $c_j$ is indexed by a clustering variable, $\theta_j$, that is deter-
mined through the Chinese restaurant process. This results in inference for the latent factors and the number of underlying groups within the collection of users. In clustering the latent factors, each user in a cluster has the same vector of latent factors. The Rasch type parameter, $\alpha_j$, is added to give some variation for each user, seeing as the clustering of latent factors provides only a few distinct vectors of latent factors.

The remainder of the SPARFA model, including the latent question–concept matrix $W$ and intrinsic difficulty parroters $\mu$ remain as before. We can treat $W$ parametrically as in Chapter 2, or non-parametric as in Chapter 3. For the sake of clarity we will discuss the more general case with a non-parametric IBP prior placed on $W$. The reduction to the simpler case of a parametric $W$ with known value of $K$ is straightforward.

2.4 Complete model

Given the continuous latent variable defined in 3.4, and assume the nonparametric priors described above, the full model is written as follows

$$
P(Y_{i,j} = 1) = \Phi(Z_{i,j};0,1), \quad Z_{i,j} = \bar{w}_i^T c_{\theta_j} + \mu_i + \alpha_j
$$

$$
\{\theta_j\} \sim \text{CRP}(\beta), \quad c_j \mid P \sim P
$$

$$
P \sim \text{DP}(\beta, G_0), \quad \mu_i \sim \mathcal{N}(m, v)
$$

$$
\alpha_j \sim \mathcal{N}(m, v), \quad w_{ik} \mid \tau_k, H_{i,k} \sim H_{i,k}\mathcal{N}^+(0, \tau_k^{-1}) + (1 - H_{i,k}) \delta_0
$$

$$
H \sim \text{BBP}(\alpha), \quad \alpha \sim \text{Ga}(a, b), \quad \tau_k \sim \text{Ga}(a, b)
$$

for all $j = 1, \ldots, N$ and $i = 1, \ldots, Q$. Here, $G_0 = \mathcal{N}_K(0, I_K)$ and $(m, v, m, v, a, b, a, b)$ are fixed hyperparameters.
4.3 Algorithm

As the model given in (4.1) is intractable, we require a computational algorithm to simulate draws from the joint posterior of the model parameters. Sec. 4.3.1 provides a brief outline of the procedure. Details regarding the use of the posterior draws for inference are found in Sec. 4.3.2.

4.3.1 Markov chain Monte Carlo algorithm

We employ a Markov Chain Monte Carlo (MCMC) algorithm to obtain samples from the joint posterior distribution of the model parameters. We outline the algorithm below; details of most of the full sampling procedure can be found in [50].

For each iteration:

1. Impute any missing values in $Y$.

2. Update $Z$ from the truncated normal full conditional.

3. Update $\mu$ from the normal full conditional.

4. For each $i$ in $1, \ldots, Q$
   
   (a) Update each $(H_{i,k}, W_{i,k}), k = 1, \ldots, K$, marginally for $H_{i,k}$ then $W_{i,k} | H_{i,k}$.

   (b) Propose the addition of $k_i$ new factors with a Metropolis-Hastings step.

5. Update each $c_j$ from the $K$-variate normal full conditional.

6. Propose new cutoff values $\xi$ through a Metropolis-Hastings step, if applicable.

7. Update the precision parameters $\{\tau_k\}$ and the IBP parameter $\alpha$ from their respective gamma full conditionals.
The full conditional updates for \( c_i \) are done via a multi-step process which we now outline. We do this in three steps.

First, we sample \( C \) based on newly sampled \( W \). This is done by first removing any rows of \( C \) that pertained to columns of \( W \) that were removed during the previous iteration. Next, we divide \( C \) into a set \( C_{\text{old}} \) consisting of elements of \( C \) that were active presently and a set \( C_{\text{new}} \) consisting of the elements \( C \) pertaining to the newly added columns of \( W \) such that \( C = \{C_{\text{old}}, C_{\text{new}}\} \). We then sample \( C_{\text{new}}|Z, C_{\text{old}}, W, \mu \) from the full conditional for each set of \( i \) in the same cluster.

Second, we sample \( c_j | c_{-j}, \cdot \), for \( j = 1, \ldots, N \) under the DP prior. This is done sequentially for each \( c_j \). We sample \( c_j = c_{\theta_j} \) with probability proportional to \( n_{\theta_j} \cdot P(z_j|c_j = c_{\theta_j}, \cdot) \) and is drawn from the full conditional posterior \( P(c_j|\cdot) \) with probability proportional to \( \beta \cdot P(z_j) \), where \( P(z_j) \) is the marginal likelihood of the \( j^{th} \) column of \( Z \) defined by:

\[
P(z_j) = \int_{c_j} P(Z|c_j, W, \mu)\pi(c_j)dc_j.
\]

Finally we perform a reshuffling step on \( C \) by drawing from the full conditional for each cluster. While this step is not strictly necessary, it can significantly improve the overall convergence speed of the MCMC.

4.3.2 Posterior Inference

At each iteration of the algorithm, the number of active features, say \( K^+ \), can change. For comparable results, we base posterior inferences only on output from iterations where \( K^+ \) is equal to the posterior mode. Given the draws from the joint posterior distributions of all parameters with \( K^+ \) fixed at the posterior mode, we obtain inference for many objectives of interest.
For example, estimates for the mean and quantiles of each cell in \( Z, W, \) and \( C \) are easily calculated. The posterior mean of \( Z \) provides estimates of the posterior probability of inclusion of the particular dimension for any given factor. Consequently, we estimate the factor loadings in \( W \) by thresholding the posterior probability of inclusion. If the probability of factor \( k \) for question \( i \) is less than a specified value, the corresponding \( W_{i,k} \) is set to 0, otherwise the posterior mean is used.
4.4 Experiments

We now develop the performance of our method on both synthetic and real-world data.

4.1 Synthetic Data

We first validate our method on synthetic data. The IBP aspects of our method were sufficiently verified in Chapter 3. Because of this as well as space constraints, we will focus exclusively on the clustering of factors in by our method.

To do this, we generate synthetic data with a fixed number of latent concepts $K$ and fixed number of clusters $L$. We then deploy the algorithm of Sec. 4.3.1 and perform posterior inference as described in Sec. 4.3.2. We evaluate performance using two metrics. First, we examine the residual error between the true $\theta$ and its posterior estimate $\hat{\theta}$. We define this error as:

$$E_\theta = \frac{||\theta - \hat{\theta}||_F^2}{||\theta||_F^2}.$$  

(4.2)

We additionally examine the misclassification rate of the clustering properties of our algorithm. To do this, we examine the estimated $\theta^t$ produced by the MCMC at iteration $t$. We then create a correspondence between the clusters present in $\theta^t$ with the ground truth matrix $\theta$. This is done by sequentially choosing the cluster in $\theta^t$ that optimally matches with the corresponding cluster in $\theta$. We then remove these clusters from $\theta^t$ and $\theta$ until exhausting either $\theta^t$ or $\theta$. We then compute the misclassification error under this optimal alignment. We average this misclassification over all MCMC trials.

In our experiments we set $K = 3$ and iterate over $L \in \{2, 5, 10\}$. We repeat each experiment over 50 randomized trials. We display our results in Tables 3.1 and 3.2. Naturally, the performance improves as the problem size $N_0 = N = Q$ increases or as
the number of clusters decrease. We are able to handle a 10-cluster problem well at size \( N_0 = 200 \).

### 4.2 Real-World Educational Data

We finally examine the performance of our method on several real-world educational datasets. In each example, we have \( N \) learners who have answered up to \( Q \) questions (possibly with missing values). For each dataset we remove 20\% of the observed values in \( Y \) and deploy the method of Sec. 4.3.2. As before, let the set \( \Omega_{\text{obs}}^c \) define the set of values that we have removed for testing purposes. We compare against the non-clustering IBP SPARFA method developed in Chapter 3. We will compare two metrics. First, the average log-likelihood taken averaged across MCMC iterations. Second, we will examine the imputation accuracy of each method. We will compute the imputation accuracy by finding the posterior mode \( Y_{i,j}^{\text{mode}} \) for all \( (i,j) \)
The simulation results for each of these datasets are displayed in Table 3.3. We see that our method achieves comparable likelihood and imputation accuracy to the standard IBP factor model method. This is impressive, as our method must is both has additional constraints on $C$ and also must contend with the clustering of the latent factors.

We finally examine the number of clusters found by our method for each of the 6 educational datasets under consideration. These results are displayed in Table 3.4, where we see that a reasonably number of clusters are found for each dataset, with each cluster generally containing between 6 and 7 learners. Recall that this clustering is important in educational applications as it allows us to diagnose and
propose remediation for learners as a group, which can ultimately lead to better educational outcomes.

<table>
<thead>
<tr>
<th>Data (Size)</th>
<th>$L$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECE3077 (89 × 42)</td>
<td>3.8 (.38)</td>
</tr>
<tr>
<td>ECE220 (203 × 97)</td>
<td>51.2 (2.6)</td>
</tr>
<tr>
<td>ELEC301A (44 × 15)</td>
<td>2.5 (.07)</td>
</tr>
<tr>
<td>ELEC301B (143 × 41)</td>
<td>7.0 (1.1)</td>
</tr>
<tr>
<td>Mech Turk (34 × 99)</td>
<td>8.42 (2.3)</td>
</tr>
<tr>
<td>STEM14g (80 × 145)</td>
<td>1.97 (0.6)</td>
</tr>
</tbody>
</table>

Table 4.4: The number of clusters $L$ uncovered by our method for each dataset.
4.5 Conclusions

We have proposed a variance of SPARFA that automatic clusters learners based on similar concept mastery. We have demonstrated this method on both synthetic and real-world data. In the latter case, we have shown that the additional structure imposed by our model does not hurt the overall inference and imputation accuracy of our method when compared to a state-of-the-art non-parametric method for factor analysis, while additionally being able to cluster similar users automatically.
Chapter 5

Collaboration-Type Identification

5.1 Introduction

5.1.1 Today’s challenges in identifying collaboration

A well-known challenge for educators is identifying collaboration among learners (or students) in a course, test, or exam [66,67]. This task is important for a number of reasons. The first and most obvious reason is that there are many educational scenarios in which collaboration is prohibited and considered a form of cheating. Identifying collaboration, in this instance, is important for maintaining fairness and academic integrity in a course. The second reason is that collaboration among learners complicates the accurate evaluation of a learner’s true level of competence. If, for example, a group of learners work together on a set of homework problems, then it is difficult to evaluate the competence of each individual learner as opposed to the competence of the group as a whole. This aspect is especially important in scenarios where learners are simply copying the responses of a single peer. In such a scenario, a series of correct answers among collaborative group members could lead to the conclusion that all learners have mastered the material when, in reality, only one
learner in the group achieved proficiency.

Manually identifying collaboration among learners is difficult enough when the class size is moderately small, say 20–30 learners, where an instructor may have a reasonable knowledge about the aptitudes and habits of each particular learner. The problem is exacerbated as the class size increases to university-level classes with hundreds of learners. In the setting of online education, such as massive open online courses (MOOCs), a manual identification of learner collaboration (or cheating-through-collaboration) becomes infeasible, as potentially thousands of learners may be enrolled in a course, without ever having face-to-face interaction with an instructor [68].

5.1.2 Automated collaboration identification

An alternative to manually identifying learners that collaborate is to rely on statistical methods that sift through learner response data automatically. Such data-driven methods look for patterns in learner answer data in order to identify potential collaborations. A naïve approach for automated identification of collaboration in educational datasets, such as multiple-choice tests, would consist of simply comparing the answer patterns between all pairs of learners and flagging learner pairs that exhibit a high degree of similarity. This approach, however, is prone to fail, as it ignores the aptitude of the individual learners, as well as the intrinsic difficulty of each test item or question [67,69].

In order to improve on such a naïve approach, several authors have examined statistically principled methods for collaboration detection [67,70]. These methods typically involve two steps. First, they estimate the probability that each learner will provide the correct response to each question by fitting models to both learners and questions. Second, they examine the actual answers provided by learners and
compute a statistical measure on how likely the learner response patterns are to have arisen by chance. While such methods for collaboration identification have led to promising results, they possess a number of limitations:

- The first limitation of prior work in statistical collaboration detection is the overwhelming focus on multiple-choice testing. While multiple-choice exams are a fact of life in many settings, they are very limiting. For example, creating useful multiple choice questions is non-trivial and requires careful thought and planning [71, 72]; this is especially true when creating effective wrong answers (lures) [73]. Additionally, the type of knowledge that can be tested on multiple choice exams is quite limited. This is especially true in fields such as STEM (science, technology, engineering, and mathematics) as well as economics [74]. Hence, automated collaboration identification methods should be able to analyze more general forms of learner response data.

- The second limitation is the use of simplistic models for how collaborative behavior between learners manifests in learner response data. The method of [67], for example, proposed the combination of point-estimates of the learner’s success probability (which are estimated directly from multiple-choice test results) and a basic model on the number of correspondences that should arise between learners based on these success probabilities. However, this method does not take into account the variety of complex ways that learners could collaborate, ranging from simple copying to symbiotic collaboration. By employing a variety of models for different types of collaborative scenarios, one could hope to improve overall identification performance as well as provide valuable information to educators.
Figure 5.1: Block diagram for our proposed methodology for collaboration-type identification. The methodology consists of (i) learning analytics (Sec. 5.2) that model the success probabilities between learners and questions from learner response data, (ii) collaboration models (Sec. 5.3) for various types of real-world collaborative behavior, and (iii) collaboration detection algorithms (Sec. 5.4) that jointly identify collaboration and classify it according to one of the collaboration models. The collaboration graph summarizes the result of the collaboration detection algorithm graphically. In this example, the collaboration graph depicts collaboration on a final exam for an undergraduate electrical engineering course. Collaboration was detected among three groups of learners. In two cases, collaboration is classified as symbiotic (denoted by solid, dark blue lines). In the other case, collaboration was classified as parasitic copying (denoted by the dashed, green line). Further details of this real-world application example are given in Sec. 5.5.

- The third limitation is due to the poor methods employed for Learning Analytics. By employing a more sophisticated LA method, such as SPARFA, one could hope to improve the identification of collaboration in educational datasets.

5.1.3 Contributions

This chapter develops a novel methodology for collaboration-type identification, first proposed in [75], which jointly identifies which learners collaborated and classifies the type of collaboration employed. A block diagram of our methodology is shown in
Fig. 5.1. Our approach overcomes the limitations of existing approaches described in Sec. 5.1.2. Concretely, we make the following four contributions, each one corresponding to one of the four blocks shown in Fig. 5.1.

- **Generic learner response data:** Our methodology relies only on simple right/wrong response data as opposed to multiple-choice responses (which usually contain multiple options per question). This response model enables our approach to be applied to a much broader range of educational datasets than existing methods.

- **Improved learning analytics:** Our methodology utilizes SPARFA to deliver superior LA over state-of-the-art methods. We note, however, that the algorithms employed are not tied to any particular LA method. In fact, any LA method that estimates success probabilities for each learner–question pair can be utilized. Furthermore, the LA method used in combination with our approach can either provide point estimates or full posterior distributions of the success probabilities.

- **Improved models for collaboration type:** Our methodology proposes four novel models for describing collaboration in real-world educational scenarios. By employing these models, our methodology provides superior performance and increased flexibility in representing real-world collaborative behavior.

- **Novel algorithms for collaboration-type identification:** Our methodology provides two novel algorithms for collaboration-type identification that fuse LA and collaboration models. These algorithms have superior performance compared to state-of-the-art algorithms for detecting collaboration in educational datasets.
5.1.4 Organization of this chapter

The remainder of this chapter is devoted to detailing our methodology for collaboration-type identification depicted in Fig. 5.1. Although we will use SPARFA exclusively in our experiments, our method is generic in that it can be used with many different LA methods. In Sec. 5.2, we provide a brief review about the restrictions on these LA methods. Although we will use SPARFA exclusively in our simulations, many LA methods can be used in our framework. In Sec. 5.3, we develop probabilistic models for various types of collaborative behavior between pairs of learners. In Sec. 5.4, we develop two novel algorithms for collaboration-type identification that make direct use of LA and collaboration models to search for likely pairs of learners engaged in collaboration. To demonstrate the efficacy of the proposed methodology, we validate our algorithms on both synthetic and real-world educational data in Sec. 5.5. We conclude in Sec. 5.6. The computational details of our methods are relegated to Appendices C and D.

5.2 Statistical approaches for Learning Analytics

As discussed above, naïve methods for collaboration identification that simply compare the pattern of right/wrong learner responses are prone to fail because they do not take into account the ability of each learner and the difficulty of each question. We use the term learning analytics (LA) to refer to methods that estimate the probability that a given learner will be successful on a given question. LA is typically accomplished by specifying models on both the learner abilities and the question difficulties. By fusing these, one can formulate a statistical model and develop cor-
responding algorithms for estimating the success probability of a given learner for each question. Recent approaches to LA enable us to distinguish scenarios where two learners have highly similar response patterns due to active collaboration as opposed to simply having similar abilities or a set of very easy/hard questions, where learners jointly succeed/fail with high probability.

The collaboration identification methodology developed in this work is generic in that one may use an arbitrary LA algorithm as long as it provides an estimated success probability for each learner on each question. We have seen how SPARFA provides this via the latent variable $\Phi(Z)$. SPARFA, however, is not the only way of achieving this.

As a concrete example, consider the Rasch model [14, 15], a simple yet powerful approach to LA.

This model assumes that each learner can be adequately characterized by a single latent ability parameter, $c_j \in \mathbb{R}, j = 1, \ldots, N$. Large positive values of $c_i$ indicate strong abilities, while large negative values indicate weak ability. Questions are also modeled by a single parameter $\mu_i \in \mathbb{R}, i = 1, \ldots, Q$, with large positive values indicating easy questions and large negative values indicating difficult questions. By defining the slack variables

$$Z_{i,j} = c_j + \mu_i, \ \forall i, j,$$

the Rasch approach expresses the probability of user $i$ answering question $j$ correctly (with $Y_{i,j} = 1$) or incorrectly (with $Y_{i,j} = 0$) using

$$Y_{i,j} \sim Ber(\Phi(Z_{i,j})), \ \forall i, j.$$
The conventional Rasch model deploys the inverse logistic link function defined as

\[ \Phi_{\log}(x) = \frac{\exp(x)}{1 + \exp(x)}. \]

Alternatively, one can use the inverse probit link function defined as

\[ \Phi_{\pro}(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{t^2}{2}\right) dt. \]

An advantage of the inverse probit link function (over the inverse logistic link) is that, when coupled with suitable prior probability distributions (i.e., Gaussian distributions) for each parameter, it enables efficient Markov chain Monte–Carlo (MCMC) methods based on Gibbs’ sampling [76]. We will employ our earlier notation such that \( \Phi(x) = \Phi_{\pro}(x) \).

MCMC methods enable us to sample from the posterior distribution of each Rasch parameter of interest in a computationally tractable manner. Among these parameters is the latent success probability \( p_{i,j} = \Phi(Z_{i,j}) \) which denotes the probability of user \( j \) correctly responding to question \( i \). Such a Rasch MCMC sampler will produce a series of samples from the posterior distribution of \( p_{i,j} \), which will be useful when developing the collaboration-type detection algorithms in Sec. 5.4. We reserve the treatment of the full sampling details of the Rasch MCMC sampler for Appendix C.

5.3 Statistical models for real-world collaborative behavior

Learners in real-world educational settings typically use a variety of strategies for providing responses to questions. In many scenarios, learners simply work independently (i.e., without any collaboration). In other cases, weaker learners may simply copy the
responses of a stronger classmate. In yet other scenarios, learners may work together collaboratively such that every learner within the group both participates and benefits. Other times, learners may defer to one trusted learner’s answer, regardless of whether or not the trusted learner is actually correct. The fact that learners may collaborate on only a subset of questions further complicates automated collaboration identification.

By explicitly modeling collaboration type, one could hope to both provide valuable information regarding collaboration as well as to improve detection of collaborating learners. To this end, we propose four statistical collaboration models that capture a range of different scenarios. We use the notation $M_m$ for $m = 1, \ldots, 4$ to refer to each model. We express our models probabilistically for a given pair of learners, i.e., learner $u$ and learner $v$, and model the joint probability distribution of observing the set of answers $(Y_{i,u}, Y_{i,v})$. This joint distribution naturally depends first on the prior success probabilities $p_{i,u}$ and $p_{i,v}$ of both learners. In practice, these probabilities can be estimated via an LA approach such as the Bayesian Rasch model or SPARFA model. All models (with the exception of the independence model) are parameterized by a scalar variable $\varepsilon_1 \in [0, 1]$, which characterizes the probability that two learners will choose to collaborate on a given question. This parametrization enables us to capture the fact that learners might only collaborate on a subset of all $Q$ questions.

Additionally, two of the collaboration-type models we propose will utilize a second parameter, $\varepsilon_2 \in [0, 1]$; the meaning of this parameter is model specific and will be explained when applicable. To simplify notation, we will use the following definitions $\bar{\varepsilon}_1 = 1 - \varepsilon_1$ and $\bar{\varepsilon}_2 = 1 - \varepsilon_2$, as well as $\bar{p}_{i,u} = 1 - p_{i,u}$ and $\bar{p}_{i,v} = 1 - p_{i,v}$. 
### 5.3.1 Collaborative Models

**Independence model** $\mathcal{M}_1$  
Under the independence model, a pair of learners is not collaborating. Instead, each learner answers the assigned questions independently. Hence, there are no parameters $\varepsilon$ for this model. The probability of observing any answer sequence for two learners working independently is simply given by the product of the individual prior probabilities. For example, the graded response pair $(1, 0)$ is achieved if learner $u$ provides a correct response to the $i^{th}$ question, while learner $v$ provides an incorrect response. This case occurs with probability $p_{i,u} \bar{p}_{i,v}$. The likelihoods for each of the four possible observed set of responses under the independence model for a given question are given in Tbl. 5.1.

**Parasitic model** $\mathcal{M}_2$  
Under the parasitic model of collaboration, only one of the two learners under consideration attempts to solve the question while the other learner simply copies the solution. The parasitic model is a two-parameter model with parameters $\varepsilon_1$ and $\varepsilon_2$. The first parameter $\varepsilon_1$ models the rate of collaboration, with a value of $\varepsilon_1 = 1$ denoting that the learner pair collaborates on every question; $\varepsilon_1 = 0$
denotes that the learners will never collaborate (thus, collapsing to the independence model). The second parameter \( \varepsilon_2 \) denotes to the probability that each learner will be selected to answer the question. A value of \( \varepsilon_2 = 0 \) denotes that learner \( u \) will always be the one selected to solve the question, while \( \varepsilon_2 = 1 \) denotes that learner \( v \) will always be the one selected. For example, observing the graded response pair \((0, 0)\) occurs in the event that (i) both learners do not collaborate on the question and both provide incorrect responses independently or (ii) both learners are collaborating on the question and that the learner chosen to solve the question does so incorrectly. The probability of this event is given by \( \bar{p}_{i,u} \bar{p}_{i,v} \bar{v}_1 + \varepsilon_1 (\bar{p}_{i,u} \bar{v}_2 + \bar{p}_{i,u} \bar{v}_1) \). The likelihood table for each of the four possible observed set of responses under the parasitic model is given in Tbl. 5.2.

**Dominance model \( M_3 \)** Under the dominance model, each learner works a question independently, after which each pair of learners discusses which of the two answers will be used. Under this model, each of the two learners attempts to convince the other to accept their response. The parameter \( \varepsilon_1 \) denotes the probability that the pair will collaborate on a given question (analogous to the parasitic model), while the second parameter \( \varepsilon_2 \) denotes the probability that learner \( u \) will convince learner \( v \) to adopt their response. For example, \( \varepsilon_2 = 1 \) implies that learner \( u \) will always convince learner \( v \), while \( \varepsilon_2 = 0 \) indicates the opposite scenario. Under this model, observing the graded response pair \((0, 0)\) occurs in either the event that (i) both learners get the incorrect response (regardless of which learner dominates) or (ii) only one learner produces an incorrect responses, but convinces the other learner to accept the response. The probability of this event is given by \( \bar{p}_{i,u} \bar{p}_{i,v} + \bar{p}_{i,u} p_{i,v} \bar{v}_1 \varepsilon_2 + p_{i,u} \bar{p}_{i,v} \bar{v}_1 \varepsilon_2 \). The likelihood table for each of the four possible observed set of responses under the dominance model is given in Tbl. 5.3.
**OR model $\mathcal{M}_4$** Under the OR model, each learner may not be able to provide the correct response to a given question. However, they can identify the correct response if at least one of them is able to provide it. Thus, the learner pair will jointly provide the correct response if at least one of the learners succeeds. The name of this model derives from the Boolean OR function, which is 1 if either one or both inputs to the function are 1 and 0 otherwise. This model only uses a single parameter $\varepsilon_1$, which denotes the probability that the pair will collaborate on a given question (analogous to the Models $\mathcal{M}_2$ and $\mathcal{M}_3$). As an example, the graded response pair (1, 1) occurs if (i) both learners produce the correct response (regardless of whether or not they are collaborating on the question) or (ii) only one learner produces the correct response and the pair is actively collaborating on the given question. This probability of this scenario is given by $p_{i,u}p_{i,v} + \tilde{p}_{i,u}p_{i,v}\varepsilon_1 + p_{i,u}\tilde{p}_{i,v}\varepsilon_1$. The likelihood table for each of the four possible observed sets of responses under the OR model for question $i$ is given in Tbl. 5.4.

### 5.3.2 Discussion of pairwise collaboration models

Many more models can be developed to emulate various collaboration types. Such new models can be easily integrated into our methodology. We further note a number of correspondences that exist between the collaboration models detailed above. For example, the models $\mathcal{M}_2$, $\mathcal{M}_3$, and $\mathcal{M}_4$ are equivalent to $\mathcal{M}_1$ whenever $\varepsilon_1 = 0$, i.e., with the collaboration probability equal to zero. Further, models $\mathcal{M}_2$ and $\mathcal{M}_4$ are equivalent under the same value of $\varepsilon_1$ and whenever $\varepsilon_2$ is either 0 or 1.

One limitation of the collaboration models proposed above is that we have entirely decoupled the collaboration rate parameter $\varepsilon_1$ from the success probabilities $p_{i,u}$ and $p_{i,v}$. In real educational scenarios, learners might choose to collaborate when they perceive a large potential benefit. Learners may, for example, be less likely to collabor-
orate on questions that they are likely to answer correctly (i.e., $p_{i,u}, p_{i,v}$ are large) and more likely to collaborate on questions that they are likely to answer incorrectly (i.e., $p_{i,u}, p_{i,v}$ are small). The development of such collaboration models is an interesting topic for future work.

5.4 Algorithms for Collaboration-Type Identification

We now develop two novel algorithms for pairwise collaboration-type identification. Both algorithms jointly utilize learner–response data, an LA method, and a set of collaboration models to jointly detect and classify different types of collaboration in educational datasets (recall Fig. 5.1).

The first algorithm, referred to as *sequential hypothesis testing* (SHT), uses a Bayesian hypothesis test first introduced in [77]. This algorithm examines the joint answer sequence of a pair of learners and evaluates the likelihood that such patterns would arise independently (under model $M_1$) or under one of the other collaboration model ($M_2, M_3,$ or $M_4$). The second algorithm, referred to as *collaborative model selection* (CMS), uses Bayesian model selection [78] in order to jointly compute posterior distributions on the probability of learner response data arising under various collaboration models.

5.4.1 Sequence hypothesis testing (SHT)

SHT compares two hypotheses. The first hypothesis $H_1$ corresponds to the case where learner $u$ and $v$ collaborate under a pre-defined collaboration-type model $M_m, m \neq 1,$ given the LA parameters. The second hypothesis $H_2$ of SHT assumes that the number of agreements between the graded responses of learner $u$ and $v$ are a result of the
independence model $\mathcal{M}_1$, given the LA parameters.

### 5.4.1.1 Collaboration hypothesis

We start by defining the first hypothesis $H_1$, which models the situation of observing the given pair of graded responses sequences for learner $u$ and $v$ under the chosen collaboration model $\mathcal{M}_m$, $m \neq 1$. Note that the SHT method can be utilized with any of the collaborative models introduced in Sec. 5.3. The model proposed here relies on the individual probabilities $p_{i,u}$ and $p_{i,v}$, which are the probabilities of learner $u$ and $v$ succeeding in question $i$ given the LA parameters. For ease of exposition, we will proceed with our derivation for a two-parameter model such as $\mathcal{M}_2$ or $\mathcal{M}_3$ with parameters $\varepsilon_1$ and $\varepsilon_2$; the reduction to a single parameter model (such as the OR model) or the extension to a model with additional parameters is straightforward.

Assuming uniform priors on $\varepsilon_1$ and $\varepsilon_2$ over the range $[0,1]$ our collaboration hypothesis for a given model $\mathcal{M}_m$ is simply given by

$$P(H_1 | \mathcal{M}_m) = \frac{1}{\mathcal{Z}} \prod_{i=1}^{Q} P(Y_{i,u}, Y_{i,v} | p_{i,u}, p_{i,v}, \varepsilon_1, \varepsilon_2, \mathcal{M}_m) d\varepsilon_1 d\varepsilon_2,$$  \hspace{1cm} (5.1)

which corresponds to the probability of observing the pair of sequences of graded responses for all $Q$ questions under the collaboration model $\mathcal{M}_m$. The quantity in (5.1) can be computed efficiently via convolution; we reserve the computation details for App. D.

### 5.4.1.2 Independence hypothesis

The probability of the second hypothesis $H_2$ for SHT corresponds to the probability of the observed pair of graded response sequences, given the success probabilities $p_{i,u}$
and $p_{i,v}$ obtained under the independence model $\mathcal{M}_1$, i.e.,

$$P(\mathcal{H}_2) = \prod_{i=1}^{Q} p_{i,u}^Y \bar{p}_{i,u}^{1-Y} p_{i,v}^Y \bar{p}_{i,v}^{1-Y}. \quad (5.2)$$

Given the probabilities (5.1) and (5.2) for the hypotheses $\mathcal{H}_1$ and $\mathcal{H}_2$, respectively, we can finally compute the log Bayes factor\(^1\) for SHT for a given pair of learners as follows:

$$LBF = \log \left( \frac{P(\mathcal{H}_1)}{P(\mathcal{H}_2)} \right). \quad (5.3)$$

A log Bayes factor greater than 0 indicates more evidence for the collaborative hypothesis (under the chosen model $\mathcal{M}_m$) than the independent hypothesis, while a log Bayes factor smaller than 0 indicates the reverse scenario. In general, however, a large value of the log Bayes factor is required when asserting that the evidence of collaboration is strong.

### 5.4.1.3 Discussion of SHT

The primary advantage of the SHT method is computational efficiency and flexibility. It can be used with simple point estimates of the learner success probabilities. Thus, it can be easily incorporated into classical approaches for LA, such as the standard (non-Bayesian) Rasch model [14] or item-response theory (IRT) [41]. When utilized in this way, the log Bayes factor needs only be computed once for each pair of learners, making it computationally very efficient.

SHT can also be used with a fully Bayesian LA approach (such as those detailed in Sec. 5.2 that provide full posterior distributions of the learner success probabilities). This is done by adding the computation of (5.3) as an additional sampling step of

\(^1\)Under a uniform prior, the log Bayes factor is called the log likelihood ratio (LLR) in the statistical signal processing community.
the MCMC sampler. Concretely, we compute (5.3) at each iteration of the MCMC sampler given the current estimates of $p_{i,u}$ and $p_{i,v}$, $\forall i, u, v$. The log Bayes factor can be equivalently converted to a posterior probability for each hypothesis, from which we can sample the hypotheses directly as part of the MCMC sampler. This approach has the advantage of improving the robustness of our inference over classical approaches, albeit at higher computational cost.

One restriction of our method is that SHT compares the independence model $M_1$ against exactly one other collaboration model $M_m$. One could, however, consider testing multiple models simultaneously by using a form of Bonferroni correction to control the family-wise error rate [79]. The approach proposed in the next section avoids such necessary corrections by means of Bayesian model selection.

5.4.2 Fully Bayesian Collaborative Model Selection

We now turn to a collaboration-type identification method based on Bayesian model selection [78]. This method allows us to jointly explore multiple collaboration models (types) and to extract the associated model parameters in an efficient way in order to find configurations that best explain the observed data. The result will provide estimates of the full posterior distributions for each collaboration model and each parameter of interest. We dub this method collaborative model selection (CMS).

5.4.2.1 Generative model for CMS

We first present the complete generative model for the pairwise collaborative model and state all necessary prior distributions. This will enable efficient MCMC sampling methods for estimating the relevant posterior distributions.

The full generative model is illustrated in Fig. 5.2 for the case of the SPARFA LA model (the equivalent Rasch-based model is obtained by removing the node $W$ and
replacing $C$ with the vector $c$). By symmetry of the proposed collaboration models, collaboration between each pair of $N$ learners can be specified with $D = \frac{1}{2}(N^2 - N)$ total models and corresponding sets of the associated model parameters. We will use the quantity $M_d$ to denote the random variable that indexes the collaboration model for learner pair $d$; the notation $\epsilon_d$ denotes the random vector of model parameters for learner pair $d$. For the collaborative model index $M_d$ we assume a discrete prior $\pi_{m,d}$ such that $\sum_{m=1}^{4} \pi_{m,d} = 1$ for all $d$. For the elements of the parameter vector $\epsilon_d$, we assume a Beta-distributed prior $Beta(\alpha_\epsilon, \beta_\epsilon)$. Generation of the latent variables in $Z$ is done for either the Rasch or SPARFA LA model. Finally, the observed learner–response matrix $Y$ for learner $u$ and $v$ is generated jointly as detailed in Sec. 5.3 given the model type index $M_d$ and the associated model parameters $\epsilon_d$.

### 5.4.2.2 MCMC sampling for collaboration type detection

Given the graded response data matrix $Y$ along with the prior distribution on $M_d$ and $\epsilon_d$, we wish to estimate the posterior distribution of each model index along with its respective parameters for each pair of learners $d = 1, \ldots, D$. Doing this will allow us to infer (i) which pairs of learners are collaborating, (ii) what type of collaborative model are they using, and (iii) how strong the evidence is for these assertions.

We use Bayesian model selection techniques [78] to efficiently search the space of possible models and model parameters for configurations that best explain the

![Graphical model for collaborative model selection (CMS).](image-url)
observed data $Y$. Full conditional posteriors for the models and model parameters, however, are not available in closed form, rendering Gibbs’ sampling infeasible. Thus, we make use of a suitable Metropolis-Hastings step [76]. Specifically, assume that at iteration $t$ of the MCMC sampler and for a specific pair of learners $d$, we have a model sample $M^t_d$ parametrized by $\epsilon^t_d$. The Metropolis-Hastings step proceeds by proposing a new model $M^{t+1}_d$ with parameters $\epsilon^{t+1}_d$ via some proposal distribution $q(M^{t+1}_d, \epsilon^{t+1}_d | M^t_d, \epsilon^t_d)$. We will utilize a proposal distribution of the following form:

$$q(M^{t+1}_d, \epsilon^{t+1}_d | M^t_d, \epsilon^t_d) = q_\epsilon(\epsilon^{t+1}_d | M^{t+1}_d, M^t_d, \epsilon^t_d) q_M(M^{t+1}_d | M^t_d, \epsilon^t_d)$$

$$= q_\epsilon(\epsilon^{t+1}_d | M^{t+1}_d, M^t_d, \epsilon^t_d) q_M(M^{t+1}_d | M^t_d).$$

In words, we (i) split the proposal into a model component and model parameters component and (ii) make use of a proposal for the model $M^t_d$ that is independent of the model parameters $\epsilon_d$. We implement this proposal in two steps: First, we propose $M^{t+1}_d \sim q_M(M^{t+1}_d | M^t_d)$. Note that there are many choices for this proposal; we will make use of the following simple one given by

$$p(M^{t+1}_d = M^{t+1} | M^t_d = M^t) = \begin{cases} 
\gamma, & \text{if } M^{t+1} = M^t \\
1 - \gamma, & \text{if } M^{t+1} \neq M^t.
\end{cases}$$

(5.4)

Here, $\gamma \in (0, 1)$ is a user-defined tuning parameter. In words, with probability $\gamma$ the MCMC sampler will retain the previous model; otherwise, one from the remaining $|\mathcal{M}| - 1$ models is proposed uniformly. The proposal for the parameters $\epsilon^{t+1}_d$ takes
the following form:

\[
q(\varepsilon_{d}^{t+1}|M_{d}^{t+1}, M_{d}^{t}, \varepsilon_{d}^{t}) = \begin{cases} 
\delta_0, & \text{if } M_{d}^{t+1} = M_1 \\
q_\varepsilon(\varepsilon_{d}^{t+1}|M_{d}^{t+1}, M_{d}^{t}, \varepsilon_{d}^{t}), & \text{if } M_{d}^{t+1} = M^t, \\
\pi_\varepsilon(\varepsilon|\alpha, \beta), & \text{otherwise},
\end{cases}
\]

(5.5)

where \(\delta_0\) corresponds to a point-mass at 0; the distribution \(q_\varepsilon\) corresponds to a random walk proposal on the interval [0, 1] defined by

\[
q_\varepsilon(a|b) = Beta(cb, c(1-b)),
\]

(5.6)

where \(c > 0\) is a tuning parameter. In words, the sampling of \(\varepsilon_{d}^{t+1}\) (i) is performed via a random walk when the model remains unchanged, (ii) is drawn directly from the prior \(\pi_\varepsilon(\varepsilon|\alpha, \beta)\) when a new model non-independent model is proposed, and (iii) is set to 0 when the model changes to the independence model (since this model has no parameters \(\varepsilon\), it is simply set to 0 for convenience). Note that it can be shown that the mean of the proposal distribution \(q_\varepsilon\) is simply \(b\) (the previous value used) while the variance is \(\frac{c^2(b-b^2)}{c^2+1}\), which tends to zero as \(b\) approaches either 0 or 1.

After proposing the new model \(\{M_{d}^{t+1}, \varepsilon_{d}^{t+1}\}\) via (5.4)–(5.6) we accept the proposal with a probability \(r\) as

\[
r = \min \left\{ 1, \frac{p(Y|M_{d}^{t+1}, \varepsilon_{d}^{t+1})\pi(M_{d}^{t+1})\pi(\varepsilon_{d}^{t+1}|\alpha, \beta)q_\varepsilon(\varepsilon_{d}^{t+1}|M_{d}^{t+1}, \varepsilon_{d}^{t}, q_{M}(M_{d}^{t+1}|M_{d}^{t}))}{p(Y|M_{d}^{t}, \varepsilon_{d}^{t})\pi(M_{d}^{t})\pi(\varepsilon_{d}^{t}|\alpha, \beta)q_\varepsilon(\varepsilon_{d}^{t+1}|M_{d}^{t+1}, M_{d}^{t}, \varepsilon_{d}^{t}, q_{M}(M_{d}^{t+1}|M_{d}^{t}))} \right\}.
\]

The accept/reject decision is computed individually for each learner pair \(d = 1, \ldots, D\).

### 5.4.2.3 Discussion of CMS

The primary advantage of CMS is that it can jointly search across all collaborative models for each pair of learners in the dataset. This comes at the price of additional
computational complexity, as a new set of models and model parameters must be proposed at each iteration of the MCMC.

We note that while our fully Bayesian method for collaboration detection uses the success probability matrix $\Phi(Z)$ when exploring new collaboration models, those models do not influence the sampling of $Z$ itself. This is due to the structure of the model we have proposed, as $Y$ separates the LA portion of the MCMC from the CMS portion. This assumption is similar to the work in [67] which computes success probabilities for each learner–question pair based only on the data $Y$ regardless of the evidence of collaboration.

The model depicted in Fig. 5.2 could be augmented in a way that enables us to propose a new posterior distribution for $Z$ where the current belief about the collaborative models will influence our beliefs about learner ability. For example, such a model could be accomplished by proposing an additional latent variable for the answer that a learner would have provided had they not been in collaboration; this would enable us to automatically temper our beliefs about learner ability in the event that we believe that they are involved in collaboration. We will leave such an approach for future work.

5.5 Experiments

We now validate the performance our proposed methodology. We first examine the identification capabilities using synthetic data with a known ground truth. Following this, we showcase the capabilities of our methods on several real-world educational datasets.
5.5.1 Synthetic experiments

We first validate the performance of our methodology using synthetic test data using both the SHT and CMS algorithms. We furthermore compare against two other methods. The first is the state-of-the-art method collaboration identification method developed in [67], which was designed specifically for handling responses to multiple choice exams, where one is interested in the specific option chosen by the pair of learners. Since we are interested in detecting collaboration given only binary (right/wrong) graded responses, we need to first modify the method [67] accordingly. Concretely, we set their term \( v_i \) indicating the number of wrong options for question \( i \) to 1, meaning that all wrong answers are treated equally. The remaining aspects of this method are left unchanged. The second method that we compare against is the agreement hypothesis testing method proposed in [77], which we will call AHT for short. This method utilizes a Bayesian hypothesis test similar to SHT that compares the likelihood of the independence model \( M_1 \) relative to a simple collaboration model in which two learners choose to agree in their answer patterns with some arbitrary probability \( \delta \).

We refer the interested reader to [77] for further details.

5.5.1.1 Performance metrics

In order to evaluate collaboration identification performance, we will examine how well the considered methods identify learners who collaborate relative to learners who work independently. Each of the four methods naturally outputs a collaboration metric related to the probability of collaboration between each pair of learners:

- **Bayesian Hypothesis Tests (AHT and SHT):** For each learner pair, the collaboration metric is given by the log Bayes' factor.
• **Bayesian Model Selection (CMS):** For each learner pair, we first threshold on the posterior probability that two learners worked under a collaborative model and then, we rank them according to the posterior mean of $\varepsilon_1$.

• **Wesolowsky’s Method:** For each learner pair, the collaboration metric is given by the Z-score (see [67] for the details).

By sorting the output metrics in ascending order, a course instructor can easily see which pairs of learners in a class are most likely engaging in collaborative behavior. To this end, let $\xi$ denote the output vector of pairwise metrics for each learner pair for a given algorithm. Sorting the entries of $\xi$ from smallest to largest, we can compute a normalized percentile ranking for each pair of learners. Let $I_d$ denote the index of learner pair $d$ in this sorted vector. The normalized percentile ranking is then given simply by

$$P_d = \frac{I_d}{D}, \ d = 1, \ldots, D, \quad (5.7)$$

with larger values of $P_d$ denoting higher likelihood of collaboration relative to the rest of the entire learner population.

### 5.5.1.2 Algorithm comparison on synthetic data

As a first synthetic experiment, we consider a class of $N = 30$ learners (with $D = 435$ unique learner pairs) answering $Q = 50$ questions. Learner abilities are initially generated via the SPARFA model. We select three pairs of learners who will work together collaboratively, one pair for each model $M_2$, $M_3$, and $M_4$ as defined in Sec. 5.2. Each pair has a per-question collaboration probability $\varepsilon_1 = 0.75$, while the value for $\varepsilon_2$ for each of the two-parameter models is set to 0 for simplicity. The answers
Figure 5.3: Normalized percentile ranking performance for all four collaboration methods with a synthetic dataset consisting of $N = 30$ learners and $Q = 50$ questions. Three learner pairs are engaged in collaboration, one for each of the collaborative models, with a per-question collaboration probability $\varepsilon_1 = 0.75$. Larger values indicate better identification performance. The CMS method achieves the best collaboration identification performance, followed by SHT, Wesolowsky’s method (denoted by “Wes.”), and AHT, respectively.

for the collaborating learners are generated according the appropriate collaboration model. The remainder of the learner pairs work independently, and their answers are generated according to the SPARFA model via (2.1). We then deploy CMS, SHT, Wesolowsky’s method (denoted by “Wes.” in Fig. 5.3), and AHT, and we compute the normalized percentile ranking for each learner pair according to (5.7). We repeat this experiment over 100 trials and present the normalized percentile ranking statistics for the collaborating learner pairs for each collaboration model and each algorithm as a box-whisker plot.

From Fig. 5.3, we see that CMS outperforms all other methods, both in the average and standard deviation of the normalized percentile ranking. CMS is followed by SHT, Wesolowsky’s method, and AHT. Since our proposed CMS method shows the best collaboration identification performance, we focus exclusively on this method in the remaining synthetic experiments.
Figure 5.4: Collaboration-type identification performance for the collaborative model selection (CMS) approach with a synthetic $N = 50$ learner dataset. (a) impact of varying collaboration probabilities $\varepsilon_1 \in \{0.25, 0.5, 0.75, 1.0\}$ for $Q = 50$ questions; (b) impact of numbers of questions for $Q \in \{25, 50, 75, 100\}$ with collaboration probability $\varepsilon_1 = 0.5$.

5.5.1.3 Performance evaluation for CMS over multiple parameters

We now examine the performance trends of the CMS method for a varying number of questions as well as the collaboration probability $\varepsilon_1$. First, we generate data for $N = 50$ learners with $Q = 50$ questions and sweep the collaboration probability $\varepsilon_1 \in \{0.25, 0.5, 0.75, 1.0\}$ and repeat this experiment over 100 trials. We again examine performance using the normalized percentile ranking (5.7) and display the results in Fig. 5.4(a). We can see that the performance is excellent for collaboration probabilities as low as $\varepsilon_1 = 0.5$, meaning that learners were expected to collaborate on only every-other question. Second, we fix $\varepsilon_1 = 0.5$ and sweep $Q \in \{25, 50, 75, 100\}$. The result is displayed in Fig. 5.4(b). We see that the proposed CMS method achieves excellent identification performance for $Q \geq 50$ questions.

5.5.2 Real-world experiments

We now turn to two real-world educational datasets. Specifically, we analyze datasets taken from undergraduate courses in electrical and computer engineering administered
on OpenStax Tutor.¹

5.5.2.1 Undergraduate signal processing course

We first identify collaboration on homework assignments in the course up to the first midterm examination. One interesting aspect of this course is that learners were encouraged to work together on all homework assignments, albeit with some restrictions. Concretely, each learner was assigned into a group of 2-to-4 learners with whom they were allowed to actively collaborate on homework assignments. Learners within each group were free to discuss each homework problem as well as its solution with any members of their group, though each learner was required to submit their own homework solutions for final grading. Learners were, however, not required to collaborate; the only restriction was that any collaboration with other learners was to be confined to the assigned homework group. Collaborating outside of the assigned homework group was considered cheating.

This particular setting presents an interesting test case for our method since we have a rough ground truth with which to compare our results. We examine the performance of CMS and the method of Wesolowsky on all homework assignments up to the first midterm exam; a total of \( Q = 50 \) questions and \( N = 38 \) learners. We further include all question–responses to the midterm (14 additional responses) in extracting the SPARFA parameters, though these questions were excluded from the collaboration detection algorithm. The data is especially challenging since learners were given ample time to solve and discuss homework problems. Because of this, most responses given on homework problems were correct. As a consequence, an extremely high degree of similarity between answer patterns is required for collaboration to be considered probable.

For CMS we posit collaborative connections between learner pairs for whom \( M_d \neq \)
1 (i.e., from which the independence model $M_1$ is excluded) in more than 90% of MCMC iterations and for whom the posterior mean of $\varepsilon_1$ was greater than 0.4. The $Z$-score threshold for Wesolowsky’s method was adjusted manually to provide the best match to the ground truth. We display the corresponding results in Fig. 5.5. Dotted red lines denote connections detected under Wesolowsky’s method. For the Bayesian model selection method, blue solid lines denote detections under symbiotic (OR) model, whereas dashed green lines show detections under the parasitic model.

Most collaborative types found using CMS for this dataset are of the OR type. An exception is the group $\{9, 10, 12\}$, for which the parasitic copying model was proposed most frequently. Examination of the answer pattern for these learners show that while the joint answer patterns for these learners are very similar on the homework assignments, Learners 10 and 12 perform poorly on the midterm relative to Learner 9. Thus, the algorithm assumes that their success in the homework is more a consequence of copying the responses of Learner 9 rather than because of their mastery of the subject material. We additionally note the collaborative connection between Learners 9 and 24 as well as between Learners 10 and 14. These connections arise due to high similarity in the homework answering patterns which are also quite different from the rest of the collaborative group. Interestingly, Wesolowsky’s method also found strong evidence of collaborations between Learners 9 and 24; this method, however, failed to reveal three of the intra-group collaborations found by our proposed CMS method. In the following, we omit further comparisons with Wesolowsky’s method for the sake of brevity.

As a second experiment with the same undergraduate signal processing course, we consider collaboration identification on the final exam, which was administered as a take-home test. During this examination, learners were instructed *not to collaborate* or discuss their results with any other learners in the class. The final exam consisted
of 24 questions. We deploy CMS using all questions in the course (a total of 147 questions) to extract the SPARFA parameters and search for collaboration only on the questions on the final exam. We jointly threshold on the posterior mean of $\varepsilon_1$ and the proportion of MCMC samples that indicated a non-independent collaborative model for each learner pair to arrive at the collaboration graph of Fig. 5.6. We find strong evidence of collaboration between the learner pair $\{14, 38\}$ under the symbiotic collaboration model. Interestingly, Learner 14 was also detected in the previous experiment as a learner working outside of his collaborative group on the course homework assignments. Both learners provided correct responses to each question on the final exam, although their previous performance in the course would lead one to expect otherwise. To prevent false accusations (see, e.g., [80] for a discussion on this matter), we examined their open form responses available in OpenStax Tutor and found a remarkable similarity in the text of their answers; this observation further strengthens our belief about their collaboration.

5.5.2.2 Final exam of an undergraduate computer engineering course

This course consists of 97 learners who completed the course answering a total of 203 questions, distributed over various homework assignments and three exams. We examine collaboration among learners in the final exam, which consists of 38 questions. As was the case with the signal processing final exam, the computer engineering final exam was administered as a take-home examination where learners were instructed not to collaborate with their peers. In order to extract the SPARFA parameters, we use all questions administered during the entire course and then use CMS to extract the posterior distributions for each pair of learners on the subset of questions corresponding to the final exam. We jointly threshold the posterior probability of non-independent collaboration as well as the posterior mean of $\varepsilon_1$. We display the result
in Fig. 5.7, where dashed green lines correspond to parasitic collaboration \( (\mathcal{M}_2) \) and solid blue lines denoting symbiotic collaboration \( (\mathcal{M}_4) \). Note that no collaborations were detected under the dominance model \( (\mathcal{M}_3) \).

All three groups of learners in Fig. 5.7 for whom we identify collaboration have identical answer patterns. The group \{10, 92, 62, 73\} provides the correct response to every question on the exam. The group \{5, 34, 90\} jointly miss only one question which is estimated by SPARFA to be in the mid to high range of difficulty. The group \{1, 88\} jointly miss the same two questions, one of these being found by SPARFA to exhibit low intrinsic difficulty. We manually inspected the open-form responses available in OpenStax Tutor of all learners in the identified groups to prevent false accusations. We found that there is some diversity in the responses provided by the group \{10, 62, 73, 92\}. This, coupled with the fact that each of the learners have only managed to perform slightly better than what SPARFA would predict, allows us to reasonably exclude this group from further scrutiny. By contrast, the answer patterns for the other groups reveal strong evidence of collaboration due to very similar wording and grammar. This is especially true for the pair \{1, 88\}; Learner 1 consistently provides a shortened version of the responses provided by Learner 88, including those answered incorrectly.\(^1\)

### 5.6 Conclusions

We have developed new methods for pairwise collaboration-type identification in large educational datasets, where the objective is to both identify which learners work together and classify the type of collaboration employed. Our framework combines

\(^{1}\)We analyzed the same computer engineering dataset using a different collaboration detection framework in [77]. We omitted two learners in this work due to their failure to submit all homework assignments. Thus, learner indices in these two papers differ. As an example, Learners 1 and 88 in this work thus correspond to Learners 1 and 90 in [77].
sophisticated approaches to learning analytics (LA) with new models for real-world collaboration and employs powerful algorithms that fuse the two to search for active collaborations among all pairs of learners. We have validated our methodology on both synthetic and real-world educational data and have shown that they significantly outperform the state-of-the-art methods available in the open literature. Additionally, we detected several cases of non-permissible collaboration (which is considered cheating) on both homework assignments and examinations in two undergraduate-level courses.

The collaboration rankings that our method provides can greatly facilitate collaboration identification as it provides a good (and small) set of candidates that need to be evaluated in greater depth (with respect to collaborative behavior) by an instructor. This advantage reduces the instructor’s workload and promotes fairness in educational settings.

One interesting avenue for future work involves modeling more complicated social groups among learners. In particular, extending the capability of collaboration detection methods beyond pairwise collaboration is useful in real-world educational scenarios in which learners often work in larger groups with complicated social dynamics. Another avenue for future work consists of using collaboration detection methods to “denoise” or, more colloquially, “decollaborate” LA. Such an application is crucial in the deployment of intelligent tutoring systems [81], as it could use its beliefs about collaboration to estimate the true learner ability (i.e., without collaboration).
Figure 5.5: Collaboration-type identification result for the Bayesian model selection method for the first set of homework assignments in the undergraduate signal processing class dataset. The data consists of 38 learners answering 50 homework questions plus 14 midterm exam questions. Grey ellipses designate the assigned homework groups. Dashed green lines denote parasitic collaborations, while solid blue lines denote symbiotic collaborations detected by CMS. Dotted red lines denote the connections found using Wesolowsky’s method, which, in general, finds fewer ground truth connections than the CMS method.
Figure 5.6: Collaboration-type identification result for a take-home exam in an undergraduate electrical engineering course consisting of 38 learners answering 24 questions. The connected nodes correspond to learners for which the collaboration hypothesis. Manual inspection of the open-form responses provided by Learners 14 and 38 further strengthens the collaboration hypothesis.
Figure 5.7: Collaboration identification result for a take-home exam in an undergraduate electrical engineering course consisting of 97 learners answering 38 questions. The connected nodes correspond to learners identified by CMS to be collaborating, with dashed green lines denoting one-side copying and solid blue lines denoting symbiotic collaboration. Manual inspection of the open-form responses provided by Learners 1 and 88 (highlighted by a gray oval) reveals obvious collaboration.
Appendix A

Proof of Theorem 1

Proof. To prove Theorem 1, we first define some notation. Let \( \mathcal{N}(x|m, s) = \frac{1}{\sqrt{2\pi s}} e^{-(x-m)^2/2s} \) define the normal PDF with mean \( m \) and variance \( s \). Furthermore, let \( \text{Exp}(m|\lambda) = \lambda e^{-\lambda m} \), \( m \geq 0 \) define the PDF of the exponential distribution with rate parameter \( \lambda \).

We are ultimately concerned with identifying whether the factor \( W_{i,k} \) is active given our current beliefs about all other parameters in the model. Given the probit model, this is equivalent to determining whether or not an exponential random variable is present in Gaussian noise. Let \( x|m, s \sim \mathcal{N}(0|m, s) \) with \( m \sim r \text{Exp}(m|\lambda) + (1-r) \delta_0 \) and \( \delta_0 \) the Dirac delta function located at 0. The posterior distribution \( p(m = 0|x) \) can be derived via Bayes’ rule as follows:

\[
p(m = 0|x) = \frac{\mathcal{N}(x|m = 0, s)(1-r)}{\mathcal{N}(x|m = 0, s)(1-r) + r \int \mathcal{N}(x|m, s) \text{Exp}(m|\lambda) dm},
\]

\[
= \frac{\mathcal{N}(x|0,s)}{\mathcal{N}(x|0,s) \text{Exp}(m|\lambda) dm \int \mathcal{N}(x|m, s) \text{Exp}(m|\lambda) dm \int \mathcal{N}(x|m, s) \text{Exp}(m|\lambda) dm \int \text{Exp}(0|\lambda) \mathcal{N}(x|0,s) \text{Exp}(m|\lambda) dm \int \mathcal{N}(x|m, s) \text{Exp}(m|\lambda) dm (1-r) + r \text{Exp}(0|\lambda)}.
\]

Here, it is important to recognize that \( \int \mathcal{N}(x|m, s) \text{Exp}(m|\lambda) dm \) denotes the posterior under
the continuous portion of the prior (i.e. \(m \neq 0\)). Since the exponential prior we have chosen is not conjugate to the normal likelihood, we must compute this distribution in closed form. To this end, let \(\mathcal{N}(x|m,s,\lambda) \propto \mathcal{N}(x|m,s)\text{Exp}(m|\lambda) = C_0 e^{-(x-m)^2/2s-\lambda m}\) denote a rectified normal distribution with normalization constant \(C_0\). Completing the square and carrying out the integration, we find 

\[
C_0 = e^{m^2/2s}\frac{1}{\sqrt{2\pi s}}
\]

which leads to

\[
\mathcal{N}(x|m,s,\lambda) = e^{m^2/2s/2}\frac{1}{\sqrt{2\pi s}} e^{-(x-m)^2/2s-\lambda m}.
\]

We can now rewrite (A.1) as

\[
p(m = 0|x) = \frac{\mathcal{N}(0|m,\hat{s},\lambda)\text{Exp}(0|\lambda)}{\mathcal{N}(0|m,\hat{s},\lambda)\text{Exp}(0|\lambda) + (1-r)}
\]

or, alternatively, as

\[
\hat{r} = p(m \neq 0|x) = 1 - p(m = 0|x) = \frac{\text{Exp}(0|\lambda)}{\mathcal{N}(0|m,\hat{s},\lambda)\text{Exp}(0|\lambda) + \frac{1-r}{r}}.
\] (A.2)

All that remains now is to determine \(\hat{m}\) and \(\hat{s}\) in (A.2) for our full factor analysis scenario. Recall that our probabilistic model corresponds to \(Z = WC + M\). Further recall our definition of the observation set \(\Omega_{obs} = \{(i,j) : Y_{i,j} \text{ is observed}\}\). We can
now calculate the posterior on each coefficient $W_{i,k} \neq 0$ as follows:

\[
p(W_{i,k} | Z, C, \mu) \propto p(W_{i,k}) p(Z | W_{-(i,k)}, C, \mu)
\]

\[
\propto e^{-\lambda W_{i,k}} e^{-\frac{1}{2\sigma^2} \sum_{(j, (i,j)) \in \Omega_{obs}} \left( (Z_{i,j} - \mu_i) - \sum_{k' \neq k} W_{i,k'} C_{k', j} \right)^2}
\]

\[
= e^{-\lambda W_{i,k}} e^{-\frac{1}{2\sigma^2} \sum_{(j, (i,j)) \in \Omega_{obs}} \left( (Z_{i,j} - \mu_i) - \sum_{k' \neq k} W_{i,k'} C_{k', j} - W_{i,k} C_{k,j} \right)^2}
\]

\[
\propto e^{-\lambda W_{i,k}} e^{-\frac{1}{2\sigma^2} \sum_{(j, (i,j)) \in \Omega_{obs}} \left( W_{i,k}^2 C_{k,j}^2 - 2((Z_{i,j} - \mu_i) - \sum_{k' \neq k} W_{i,k'} C_{k', j}) W_{i,k} C_{k,j} \right)}
\]

\[
\propto e^{-\lambda W_{i,k}} e^{-\frac{1}{2\sigma^2} \sum_{(j, (i,j)) \in \Omega_{obs}} C_{k,j} \left( W_{i,k} - \frac{\sum_{(j, (i,j)) \in \Omega_{obs}} \left( (Z_{i,j} - \mu_i) - \sum_{k' \neq k} W_{i,k'} C_{k', j} \right) C_{k,j}}{\sum_{(j, (i,j)) \in \Omega_{obs}} C_{k,j}^2} \right)^2},
\]

(A.3)

where the last step is obtained by completing the square in $W_{i,k}$.

The final result in (A.3) implies that $W_{i,k} \sim \mathcal{N}^r(\hat{m}, \hat{s}, \lambda)$, where

\[
\hat{m} = \frac{\sum_{(j, (i,j)) \in \Omega_{obs}} \left( (Z_{i,j} - \mu_i) - \sum_{k' \neq k} W_{i,k'} C_{k', j} \right) C_{k,j}}{\sum_{(j, (i,j)) \in \Omega_{obs}} C_{k,j}^2}
\]

and $\hat{s} = \frac{\sigma^2}{\sum_{(j, (i,j)) \in \Omega_{obs}} C_{k,j}^2}$. Combining the results of (A.2) and (A.3), recognizing that $\sigma^2 = 1$ in the standard probit model, and adopting the notation $\hat{R}_{i,k}$, $\hat{M}_{i,k}$ and $\hat{S}_{i,k}$ for the values of $\hat{r}$, $\hat{m}$ and $\hat{s}$ corresponding to each $\hat{W}_{i,k}$, furnishes the final sampling result. \qed
Appendix B

IMPLEMENTATION DETAILS FOR NON-PARAMETRIC SPARFA

We provide the details for the MCMC algorithm for the Bayesian infinite factor model. Given the observations, $Y$, we obtain inference for the parameters of interest through MCMC techniques, using a combination of Gibbs sampling and Metropolis-Hastings updates.

1. **Update for $Y$:** We need to include possible missing values in $Y$. Let $\tilde{Y}_{i,j}$ represent a missing answer for learner $j$ at question $i$ with a corresponding latent variable $\tilde{Z}_{i,j}$. Then, the likelihood can be split into observed and unobserved data,

   $p(Y | \ldots) = \prod_{i,j \in \Omega_{obs}} \text{Bern}(Y_{i,j}; \Phi(Z_{i,j}; 0, \psi_i^{-1})) \prod_{i,j \notin \Omega_{obs}} \text{Bern}(\tilde{Y}_{i,j}; \Phi(\tilde{Z}_{i,j}; 0, \psi_j^{-1})).$

The $\tilde{Z}_{i,j}$ are readily integrated out and, therefore, we impute the $\tilde{Y}_{i,j}$ via a Bernoulli distribution with probability $\Phi(\mathbf{w}_i^T \mathbf{c}_j + \mu_i, \psi_j^{-1})$. Conditional on the imputed values, the rest of the updates are carried out assuming we have a fully
observed $Y$.

2. **Update for $Z$:** The latent variables, $Z_{i,j}$, are updated from a truncated normal distribution with mean $\bar{w}_i^T c_j + \mu_i$ and variance $\psi_j^{-1}$. This truncated normal distribution is truncated below by $\xi_{Y_{i,j}}$ and above by $\xi_{Y_{i,j}}$.

3. **Update for $\mu$:** The full conditional for $\mu_i$ follows a normal distribution with mean $s^* \left( m_0 * v_\mu + \sum_{j=1}^{N} \psi_j (Y_{i,j} - \bar{w}_i^T c_j) \right)$ and variance $s^* = (\sum_j \psi_j + v_\mu)^{-1}$.

4. **Joint update for $(H, W)$:** The $(i, k)$th element of the binary, IBP matrix, $H_{i,k}$, has a prior ratio of

$$\frac{\Pr(H_{i,k} = 1 | \ldots)}{\Pr(H_{i,k} = 0 | \ldots)} = \frac{m_{-i,k}}{Q - m_{-i,k}}$$

where $m_{-i,k}$ counts the number of questions, excluding $i$, for which concept $k$ is active. The likelihood given $H_{i,k} = 1$ requires integrating over the truncated normal prior on $W_{i,k}$. Consequently, with $\tau_k$ is the precision of factor $k$ and $e_i = z_i - \mu_i 1$, the ratio of likelihoods is given by

$$\frac{P(Z | H_{i,k} = 1, \ldots)}{P(Z | H_{i,k} = 0, \ldots)} = (\tau_k \sigma^*)^{1/2} \exp \left\{ \frac{1}{2\sigma^* \mu^2} \right\} \left( 1 - \Phi(0; \mu^*, \sigma^*) \right),$$

where $\sigma^* = (\sum_i \psi_j C_{k,j}^2 + \tau_k)^{-1}$, $\mu^* = \sigma^* \sum_i \psi_j C_{k,j} E_{i,j}$, and $\tilde{W}$ is the $W$ matrix with the $(i, k)$th cell set to 0.

Multiplying the ratios of prior and likelihood gives the ratio of posterior probabilities to be used for sampling $H_{i,k}$. Then, if $H_{i,k} = 1$, we sample $W_{i,k}$ from a truncated normal with mean $\mu^*$ and variance $\sigma^*$. 
In order to add new concepts, we must sample the number of concepts active only for question \(i\) (call this \(k_i\)). We can integrate over the new elements of the mixing matrix, \(W_{i,k_i}\), or the new rows of the latent feature matrix, \(C_{k_i,j}\), but not both.

Given that \(c_{k_i}\) will generally be of higher dimension than \(w_{k_i}\), we choose to integrate over the \(c_{k_i}\). In this case, the new elements of \(W\) are added to the proposal distribution, \(J(k_i)\) is as follows

\[
J(k_i) = \{(1 - p)\text{Pois} \left( k_i; \frac{ca}{Q-1} \right) + p1_{k_i=1}\}N^+(w_{k_i}; 0, \tau_{k_i}^{-1}).
\]

Therefore, the proposal is accepted with probability

\[
r = \min\{1, a_p\}.
\]

where

\[
a_p = \text{Pois}(k_i; \frac{ca}{Q-1})/\text{Pois}(k_i; \frac{ca}{Q-1})\) and \(a_l = p(Z | k_i, W_{k_i}, \ldots)/p(Z | \ldots).\)

The expression for \(a_l\) is given by

\[
\prod_i |\Sigma_i^*|^{-1/2} \exp \left\{ \frac{1}{2} \sum_{i=1}^{N} m_i^{*} \Sigma_i^* m_i^* \right\}
\]

where \(\Sigma_i^* = [W_{k_i}W_{k_i}^T \psi_j + I_{k_i}]\) and \(m_i^* = \Sigma_i^{-1} \hat{w}_{k_i} \hat{E}_{i,j}\) with

\[
\hat{E}_{i,j} = (Z_{i,j} - w_i^T c_j - \mu_i) \psi_j.
\]

5. **Update C**: The latent factors, \(c_j\), are updated for each \(j \in [1, \ldots, N]\) through a \(K\)-variate normal distribution with mean

\[
L^* W^T (z_j - \mu) \psi_j\) and variance \(L^* = [W^T W \psi_j + I_K]^{-1}.
\]

6. **Update for \(\xi\)**: Cutoff positions are sampled via a Metropolis-Hastings step.

Concretely, we sample \(\xi_d\) for all \(d = 2, \ldots, D - 1\) using the following proposal distribution

\[
\xi_d^0 \sim N^+(\xi_d, \sigma_{\text{MH}}^{-2}, \xi_{d-1}, \xi_{d+1}), \)

where the cutoff values \((\xi_{d-1}, \xi_{d+1})\) enforce the ordering constraint on the cutoff positions.
The accept/reject ratio is given as follows:

\[
R = \left( \prod_{j=1}^{N} \prod_{i=1}^{Q} \frac{\Phi(\sqrt{\psi_j}(\xi_{Y_{i,j}} - Y_{i,j})) - \Phi(\sqrt{\psi_j}(\xi_{Y_{i,j-1}} - Y_{i,j}))}{\Phi(\sqrt{\psi_j}(\xi_{Y_{i,j}} - Y_{i,j})) - \Phi(\sqrt{\psi_j}(\xi_{Y_{i,j-1}} - Y_{i,j}))} \right) \times \left( \prod_{d=2}^{D-1} \frac{\Phi((\xi_{d+1} - \xi_{i,d})/\sigma_{\text{MH}}) - \Phi((\xi_{d+1} - \xi_{d})/\sigma_{\text{MH}})}{\Phi((\xi_{d+1} - \xi_{i,d})/\sigma_{\text{MH}}) - \Phi((\xi_{d+1} - \xi_{d})/\sigma_{\text{MH}})} \right),
\]

where the first term corresponds the likelihood ratio while the second accounts for the non-symmetric transition probability of the proposal distribution. To make the final acceptance decision, we generate \( U \sim \text{Unif}(0, 1) \) and accept if \( U \leq R \).

7. **Update for \( \{\psi_j\}, \{\tau_k\}, \text{and} \alpha \):** The full conditionals for the \( \psi_j \) follow a gamma distribution with shape parameter \( a_\psi + Q/2 \) and rate parameter \( b_\psi + \sum_j (Z_{i,j} - \bar{w}_i^T c_j - \mu_i)^2 \).

The concept precisions, \( \tau_k \), are given the same Gamma prior, and therefore have Gamma full conditionals with shape and rate parameters \( a_\tau + \frac{m_k}{2} \) and \( b_\tau + \frac{1}{2} \sum_j W_{i,k}^2 \), where \( m_k \) is the number of questions for which concept \( k \) is active.

The full conditional for the IBP parameter, \( \alpha \), given the conjugate Gamma prior, follows a Gamma distribution with shape parameter \( K_\alpha + a_\alpha \) and rate parameter \( b_\alpha + H_Q \), where \( H_Q = \sum_{i=1}^{Q} \frac{1}{i} \) is the \( Q^{th} \) harmonic number.
Here we derive the sampling steps for the Rasch MCMC sampler. Recall that the generative model for the data $Y$ under the Rasch approach is given by

$$Y_{i,j} \sim Ber(\Phi(Z_{i,j})) \quad \text{with} \quad Z_{i,j} = c_j + \mu_i, \forall i, j.$$

It can be shown (see, e.g., [82]) that this model is equivalent to

$$Y_{i,j} \sim sign(\Phi(Z'_{i,j})) \quad \text{with} \quad Z'_{i,j} = c_j + \mu_i + e_{i,j}, \forall i, j.$$

where $sign(\cdot)$ is the signum function and $e_{i,j} \sim \mathcal{N}(0, 1)$. This latter representation is more convenient for the purposes of MCMC.

By specifying the following prior distributions

$$\pi(c_j) \sim \mathcal{N}(0, \sigma_c^2) \quad \text{and} \quad \pi(\mu_i) \sim \mathcal{N}(0, \sigma_\mu^2),$$

we can perform Gibbs’ sampling on each of the variables $c_j, \mu_i$ by augmenting with
the latent variable \( Z^i_{i,j} \). The sampling steps at each MCMC iteration are given by

1. For all \( i = 1, \ldots, Q \) and \( j = 1, \ldots, N \) sample \( Z^i_{i,j} \sim \mathcal{N}(c_i + \mu_i, 1) \), truncating above 0 if \( Y_{i,j} = 1 \), and truncating below 0 if \( Y_{i,j} = 0 \).

2. For all \( i = 1, \ldots, Q \) sample \( \mu_i \sim \mathcal{N}(\hat{\sigma}_\mu^2 \sum_{j=1}^N (Z^i_{i,j} - c_j), \hat{\sigma}_\mu^2) \), where \( \hat{\sigma}_\mu^2 = \left( \frac{1}{\sigma_\mu^2} + N \right)^{-1} \).

3. For all \( j = 1, \ldots, N \) sample \( c_j \sim \mathcal{N}(\hat{\sigma}_c^2 \sum_{i=1}^N (Z^i_{i,j} - \mu_i), \hat{\sigma}_c^2) \), where \( \hat{\sigma}_c^2 = \left( \frac{1}{\sigma_c^2} + Q \right)^{-1} \).

By repeating this sampling scheme over several iterations, we assemble a set of samples from the posterior distribution of the Rasch parameters \( c_j, \forall j \) and \( \mu_i, \forall i \). In addition, the values of \( p_{i,j} = \Phi(c_j + \mu_i) \) are samples of the probability of learner \( j \) answering item \( i \) correctly, which are then used by the collaboration-type identification algorithms of Sec. 5.4.
Appendix D

NUMERICAL EVALUATION OF (5.1)

Here, we detail the efficient numerical evaluation of the SHT collaboration hypothesis (5.1). We do this specifically for the case of a two-parameter collaboration model such as $\mathcal{M}_2$ or $\mathcal{M}_3$. Reduction to a single parameter model such as $\mathcal{M}_4$ or to the extension to a model with additional parameters is straightforward.

First, it is important to notice that the product term in (5.1) is a polynomial in the variables $\varepsilon_1$ and $\varepsilon_2$ of the form

$$\prod_{i=1}^{Q} P(Y_{i,k}, Y_{i,\ell} \mid p_{i,k}, p_{i,\ell}, \varepsilon_1, \varepsilon_2, \mathcal{M}_j) =
\begin{align*}
g_{0,0} \varepsilon_1^0 \varepsilon_2^0 + g_{0,1} \varepsilon_1^0 \varepsilon_2^{-1} + \ldots + g_{0,Q} \varepsilon_1^0 \varepsilon_2^Q + g_{1,0} \varepsilon_1^1 \varepsilon_2^0 + \ldots + g_{Q,Q} \varepsilon_1^Q \varepsilon_2^Q. 
\end{align*}
$$

(D.1)

The coefficients $g_{a,b}$ of the polynomial expansion in (D.1) can be evaluated efficiently using a 2-dimensional convolution. In particular, consider the matrix expansion

$$G = \bigotimes_{i=1}^{Q} G_i(Y_{i,k}, Y_{i,\ell} \mid p_{i,k}, p_{i,\ell}, \varepsilon_1, \varepsilon_2, \mathcal{M}_j),
$$

(D.2)

where $\bigotimes$ is the (2-dimensional) convolution operator. The term $G_i(\cdot) \in \mathbb{R}^{2 \times 2}$ is a
matrix polynomial in the variables $\varepsilon_1$ and $\varepsilon_2$ of the form

$$G_i(\cdot) = \begin{bmatrix} \tilde{G}_{i,0} & \tilde{G}_{i,1} \\ \tilde{G}_{i,0} & \tilde{G}_{i,1} \end{bmatrix},$$

where $\tilde{G}_{a,b}$ is the coefficient associated with $\varepsilon_1^a \varepsilon_2^b$ corresponding to the $i^{th}$ question $P(Y_{i,k}, Y_{i,\ell} | p_{i,k}, p_{i,\ell}, \varepsilon_1, \varepsilon_2, M_j)$. For example, $G_i(0, 0 | p_{i,k}, p_{i,\ell}, M_2)$ is given by

$$G_i(0, 0 | p_{i,k}, p_{i,\ell}, M_2) = \begin{bmatrix} \bar{p}_{i,k}\bar{p}_{i,\ell} & -\bar{p}_{i,k}\bar{p}_{i,\ell} + \bar{p}_{i,k} \\ 0 & -\bar{p}_{i,k} + \bar{p}_{i,\ell} \end{bmatrix}.$$

The result of (D.2) is a matrix $G \in \mathbb{R}^{(Q+1) \times (Q+1)}$ where $G_{a,b} = g_{a,b}$. Since

$$\int_0^1 \int_0^1 g_{a,b} \varepsilon_1^a \varepsilon_2^b d\varepsilon_1 d\varepsilon_2 = \frac{g_{a,b}}{(a+1)(b+1)},$$

we can evaluate (5.1) by computing

$$P(H_i^2) = \sum_{ij} H_{ij} \quad \text{with} \quad H = G \circ F,$$

(D.3)

where the entries of the matrix $F$ correspond to $F_{a,b} = \frac{1}{(a+1)(b+1)}$, and $\circ$ denotes the Hadamard (element-wise) matrix product. Simply put, $P(H_i^2)$ is given by the sum of all elements in the matrix $H = G \circ F$.

It is important to note that finite precision artifacts in the computation of (D.2) and (D.3) become non-negligible as $Q$ becomes large, i.e., if $Q$ exceeds around 35 items with double-precision floating-point arithmetic. In order to ensure numerical stability while evaluating of (D.2) and (D.3) for large $Q$, we deploy specialized high-precision computation software packages. Specifically, for our experiments, we used
Advanpix’s Multiprecision Computing Toolbox for MATLAB.¹
REFERENCES


[26] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Springer, 2010. 1, 2, 2.4.2, 2.5.1.4

[27] M. West, “Bayesian factor regression models in the “large p, small n” paradigm,” *Bayesian Statistics*, vol. 7, no. 2003, pp. 723–732, Sep. 2003. 2.3.1, 2.3.3.3, 3.1


overcomplete dictionaries for sparse representation,” IEEE Transaction on Signal
Processing, vol. 54, no. 11, pp. 4311–4322, Dec. 2006. 2.5, 2.5.1, 2.5.1.2, 2.6

[37] ———, “K-SVD and its non-negative variant for dictionary design,” in Proceedings
of the SPIE Conference on on Wavelets, vol. 5914, Jul. 2005, pp. 327–339. 2.5.1.2

sparse solutions to underdetermined systems of equations,” IEEE Transactions on Info Theory, vol. 54, no. 11, pp. 4813–4820, Nov. 2008. 2.5.1.2

[Online]. Available: http://dsp.rice.edu/courses/elec301 2.5.2.1

Available: https://www.mturk.com/mturk/welcome 2.5.2.3

[41] Y. Bergner, S. Droschler, G. Kortemeyer, S. Rayyan, D. Seaton, and
D. Pritchard, “Model-based collaborative filtering analysis of student response
data: Machine-learning item response theory,” in Proceedings of the 5th Interna-
tional Conference on Educational Data Mining, June 2012, pp. 95–102. 2.5.2.3,
5.4.1.3

19–60, 2010. 2.6

compressed signals,” in IEEE International Conference on on Acoustics, Speech
and Signal Processing, Kyoto, Japan, Mar. 2012, pp. 3341–3344. 2.6

of the Conference on Information Science and Systems (CISS), Princeton, NJ,
Mar. 2008. 2.6

compressive sensing via binary stable embeddings of sparse vectors,” Apr. 2011,
submitted. 2.6

[46] Y. Plan and R. Vershynin, “Robust 1-bit compressed sensing and sparse logistic

buffet process,” Technical Report GCNU TR 2005-001, 2005. 3.1, 3.2.2, 4.1


[50] K. Fronczyk, A. E. Waters, M. Guindani, R. G. Baraniuk, and M. Vannucci, “A bayesian infinite factor model for learning and content analytics,” submitted to Computational Statistics and Data Analysis., 2013. 3.1, 4.3.1


[56] V. Johnson and J. Albert, Ordinal Data Modeling. Springer, 1999. 3.3.2


[63] P. Arabie, L. Hubert, and G. De Soete, *Clustering and classification*. World Scientific, 1996. 4.1


[77] A. E. Waters, C. Studer, and R. G. Baraniuk, “Bayesian pairwise collaboration detection in educational datasets,” in *Proc. IEEE Global Conf. on Sig. and Info. Proc. (GlobalSIP)*, Austin, TX, Dec. 2013. 5.4, 5.5.1, 1

[78] P. D. Hoff, *A First Course in Bayesian Statistical Methods*. Springer Verlag, 2009. 5.4, 5.4.2, 5.4.2.2


