Time-varying Learning and Content Analytics via Sparse Factor Analysis

Andrew S. Lan MR.LAN@SPARFA.COM
Christoph Studer STUDER@SPARFA.COM
Richard G. Baraniuk RICHB@SPARFA.COM
Dept. Electrical and Computer Engineering
Rice University
Houston, TX 77005, USA

Abstract

We propose SPARFA-Trace, a new machine learning-based framework for time-varying *learning* and *content analytics* for education applications. We develop a novel message passing-based, blind, approximate Kalman filter for sparse factor analysis (SPARFA), that jointly (i) traces learner concept knowledge over time, (ii) analyzes learner concept knowledge state transitions (induced by interacting with learning resources, such as textbook sections, lecture videos, etc, or the forgetting effect), and (iii) estimates the content organization and intrinsic difficulty of the assessment questions. These quantities are estimated solely from binary-valued (correct/incorrect) graded learner response data and a summary of the specific actions each learner performs (e.g., answering a question or studying a learning resource) at each time instance. Experimental results on two online course datasets demonstrate that SPARFA-Trace is capable of tracing each learner's concept knowledge evolution over time, as well as analyzing the quality and content organization of learning resources, the question–concept associations, and the question intrinsic difficulties. Moreover, we show that SPARFA-Trace achieves comparable or better performance in predicting unobserved learner responses than existing collaborative filtering and knowledge tracing approaches for personalized education.

Keywords: Kalman filter, knowledge tracing, learning analytics, personalized learning, sparse factor analysis (SPARFA), sparse probit regression

1. Introduction

The traditional "one-size-fits-all" education approach is one of the main bottlenecks for education in the 21st century. This approach largely limits learners' learning efficiency, as it is unable to provide personalized and timely feedback to learners, and remains linear in organization regardless of the different strengths, weaknesses, goals, and interests of different learners. Recent developments in *machine learning*-based personalized learning systems (PLSs) provide great potential to achieve personalized learning, by automatically mining data from learner interactions with educational content in order to provide a scalable, personalized education experience to a large number of learners (see Psotka et al. (1988); VanLehn et al. (2005); Knewton (2012) for examples).

In our vision, a PLS should consist of two key components:

- *Learning analytics* (LA), which estimates the learners' knowledge states and dynamically traces their change over time, as the learners either *learn* by interacting with learning materials (including *learning resources*, e.g., textbook sections, lecture videos, and assessments, namely *questions* in weekly quizzes, homework assignments, and exams), or *forget* by not doing remedial studies (see, e.g., Weiner and Reed (1969)).
- *Content analytics* (CA), which provides insight on the quality and content organization of all learning resources and the nature of the forgetting effect, as well as the difficulty and content organization of the available questions.

1.1 SPARFA: sparse factor analysis for learning and content analytics

The recently proposed sparse factor analysis (SPARFA) framework for personalized learning proposed statistical models and algorithms for machine learning-based LA and CA (Lan et al. (2012)). SPARFA assumes that the learners' responses to questions in the domain of a course/assessment are governed by their knowledge on a small number of latent "concepts." In particular, SPARFA relies on the following probability model for learners' graded responses to questions:

$$Y_{i,j} \sim Ber(\Phi(Z_{i,j})) \quad \text{and} \quad Z_{i,j} = \bar{\mathbf{w}}_i^T \mathbf{c}_j - \mu_i.$$

Here, $Y_{i,j}$ is the binary-valued graded response of learner j to question i, which is assumed to be a Bernoulli random variable, and $Z_{i,j}$ is a slack variable governing the probability of learner j answering question i correctly or incorrectly. $\Phi(\cdot)$ represents the inverse logit/probit link function. The variable $Z_{i,j}$ depends on three factors: (i) the question–concept association vector $\bar{\mathbf{w}}_i$ which characterizes how question i relates to each concept, (ii) the learner concept knowledge vector $\mathbf{c}_j$ of learner j, and (iii) the intrinsic difficulty parameter μ_i of question i.

The SPARFA framework jointly estimates (i) the question–concept association of each question, (ii) the concept knowledge of each learner, and (iii) the intrinsic difficulty of each question, solely from observed binary-valued (correct/incorrect) graded learner responses to questions, under assumptions (A1)–(A3).

This paper makes two major extensions to the SPARFA framework. First, the SPARFA framework assumes that the learners' concept knowledge states remain *constant* throughout the course/assessment. This assumption prohibits SPARFA from situations where learners' responses are made at different time instances, as is usual for homework sets assigned during a semester-long course. Second, the SPARFA framework only analyzes *questions*, which measure learner knowledge states, but does not analyze learner *knowledge state transitions*, which are induced by interacting with learning resources or by forgetting (Carrier and Pashler (1992); Larsen et al. (2009)). The analysis of learning resources is of utmost importance for a PLS, since this information enables the system to automatically recommend new resources to individual learners for remedial studies.

1.2 SPARFA-Trace: time-varying learning and content analytics

In this paper, we propose *SPARFA-Trace*, a blind approximate Kalman filtering approach to perform joint *time-varying* LA and CA. The main working principle of the approach is

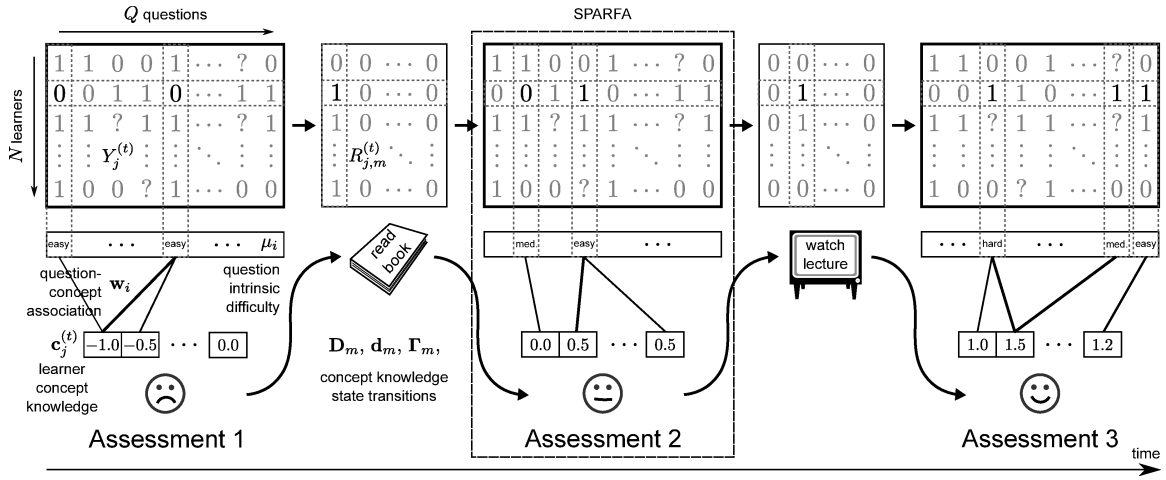


Figure 1: The SPARFA-Trace framework processes the binary-valued graded learner response matrix $\mathbf{Y}$ (binary-valued, with 1 denoting a correct response and 0 denoting an incorrect response) and the learner activity matrices $\{\mathbf{R}^{(t)}\}$ (binary-valued, with 1 denoting a learner studied a learning resource) to (i) trace learner concept knowledge states $\mathbf{c}_j^{(t)}$ over time and (ii) estimate learning resource content organization and quality parameters $\mathbf{D}_m$, $\mathbf{d}_m$ and $\mathbf{\Gamma}_m$, together with question–concept association and question intrinsic difficulty parameters $\bar{\mathbf{w}}_i$ and μ_i, respectively.

illustrated in Figure 1. Time-varying LA is performed by tracing each learner's concept knowledge (i.e., tracking the evolution of each learner's concept knowledge state vector $\mathbf{c}_j^{(t)}$ over different time instances t), based on (i) observed binary-valued (correct/incorrect) graded learner responses to questions matrix $\mathbf{Y}$ and (ii) available learner activity matrices $\mathbf{R}^{(t)}$, as shown in Figure 1. CA is performed by estimating all learner concept knowledge state transition parameters $\mathbf{D}_m$, $\mathbf{d}_m$ and $\mathbf{\Gamma}_m$, and question–concept associations and question intrinsic difficulties $\bar{\mathbf{w}}_i$ and μ_i, based on the estimated learner concept knowledge states at all time instances, as shown in Figure 1.

In order to perform all these tasks, we extend the SPARFA framework developed in Lan et al. (2012) using a new statistical model for learner knowledge state transitions that are induced by studying learning resources or the forgetting effect. Armed with this model, LA corresponds to the task of estimating latent concept knowledge states in a dynamical system, where binary-valued graded learner responses to questions are its observations. We develop a message passing-based *approximate* Kalman filtering approach to estimate the latent learner concept knowledge states at every time instance, as the underlying dynamical system is non-linear and non-Gaussian due to the binary-valued right/wrong observations, which inhibits the use of traditional Kalman filtering methods. We also propose novel convex optimization-based algorithms within the expectation-maximization (EM) framework for CA, i.e., algorithms that estimate learners' concept knowledge state transition parameters and question-dependent parameters. The estimation of these parameters is crucial, as the

approximate Kalman filtering approach requires all these parameters to be *given*, which is, in general, not the case in real educational scenarios.

To test and validate its effectiveness, we evaluate SPARFA-Trace on synthetic datasets and real-world educational datasets collected via OpenStax Tutor (OpenStaxTutor (2013)). SPARFA-Trace is accurate in tracing learner concept knowledge, and estimating learner concept knowledge state transition parameters and question-dependent parameters. Further, it outperforms existing approaches on predicting unobserved learner responses. We also demonstrate that SPARFA-Trace enables a PLS to (i) trace the learner concept knowledge state evolution over time in order to provide timely feedback to learners, and (ii) analyze the quality and content organization of all learning resources and assessment questions, in order to make effective and computerized recommendations to learners for remedial studies.

1.3 Related work

The closest related work to SPARFA-Trace is knowledge tracing (KT). KT is a popular technique for tracing learner knowledge evolution over time and for predicting future learner performance (see, e.g., Corbett and Anderson (1994); Baker et al. (2008); Pardos and Heffernan (2010b)). Powerful as it is, KT suffers from the following drawbacks: (i) KT uses binary learner knowledge state representations, characterizing learners as whether they have mastered a certain concept (or skill) or not. The limited explanatory power of binary concept knowledge state representations prohibit the design of more powerful and sophisticated LA and CA algorithms. (ii) KT assumes that each question is associated with exactly one concept. This restriction limits KT to very narrow educational domains (e.g., basic algebra), preventing it from generalizing to courses/assessments involving multiple concepts. (iii) KT uses a single "probability of learning" parameter to characterize the learner knowledge state transitions over time, and assumes that a concept cannot be forgotten once it is mastered. This modeling limitation forces all learner knowledge state transitions to be characterized by the same "probability of learning" parameter, prohibiting KT from performing CA, i.e., analyzing the quality and content organization of different learning resources that lead to different learner knowledge state transitions. See Section 6 for detailed discussions and comparisons with previous work in KT and other machine learning-based approaches to personalized learning.

1.4 Paper organization

In Section 2, we introduce the SPARFA-Trace statistical model for learner knowledge state transitions that are induced by interacting with learning resources or forgetting. In Section 3, we detail the approximate Kalman filtering approach for learner concept knowledge tracing. In Section 4, we detail convex optimization-based algorithms to estimate learners' concept knowledge state transition parameters and question-dependent parameters. In Section 5, we evaluate SPARFA-Trace on synthetic and real-world educational datasets. In Section 6, we provide a brief overview of related KT and machine learning-based techniques for personalized learning. We conclude in Section 7.

2. Statistical Model for Time-Varying Learning and Content Analytics

We start by extending the SPARFA framework (Lan et al. (2012)) to trace learner concept knowledge over time, and propose the corresponding statistical model in Section 2.1. In Section 2.2, we then characterize the transition of a learner's concept knowledge states between consecutive time instances as an affine model, which is parameterized by (i) the particular learning resource(s) the learner interacted with, and (ii) how these learning resource(s) affect learners' concept knowledge states.

2.1 Statistical model for time-varying graded learner responses to questions

The proposed statistical model characterizes the probability that a learner answers a question correctly at a particular time instance in terms of: (i) the learner's knowledge on every concept at this particular time instance, (ii) how the question relates to each concept, and (iii) the intrinsic difficulty of the question. To this end, let N denote the number of learners, K the number of latent concepts in the course/assessment, and T the total number of time instances throughout the course/assessment. We define the K-dimensional vectors $\mathbf{c}_j^{(t)} \in \mathbb{R}^K, t \in \{1,\dots,T\}, j \in \{1,\dots,N\}$, to represent the latent concept knowledge state of the j^{th} learner at time instance t. Let Q be the total number of questions. We further define the mapping $i(t,j) : \{1,\dots,T\} \times \{1,\dots,N\} \mapsto \{1,\dots,Q\}$, which maps learner and time instance indices to question indices; this information can be extracted from the learner activity log. We will use the shorthand notation $i_j^{(t)} = i(t,j)$ to denote the index of the question that the j^{th} learner answers $i_j^{(t)}$ at time instance t. Under this notation, we define the K-dimensional vector $\bar{\mathbf{w}}_{i_j^{(t)}} \in \mathbb{R}^K, i \in \{1,\dots,Q\}$ as the question–concept association vector of the question that the j^{th} learner answered at time instance t. Finally, we define the scalar $\mu_{i_j^{(t)}} \in \mathbb{R}$ to be the intrinsic difficulty of question $i_j^{(t)}$, with a large, positive values of $\mu_{i_j^{(t)}}$ representing a difficult question, while a small, negative values of $\mu_{i_j^{(t)}}$ represent an easy one.

Given these quantities, we characterize the binary-valued graded response, where 1 denotes a correct response and 0 an incorrect response, of learner j to question $i_j^{(t)}$ at time instance t as a Bernoulli random variable:

$$\begin{aligned} Y_j^{(t)} &\sim Ber(\Phi(Z_j^{(t)})), \qquad (t,j) \in \Omega_{\text{obs}}, \\ Z_j^{(t)} &= \bar{\mathbf{w}}_{i_j^{(t)}}^T \mathbf{c}_j^{(t)} - \mu_{i_j^{(t)}}, \quad \forall t, j. \end{aligned} \tag{1}$$

Here, the set $\Omega_{\text{obs}} \subseteq \{1,\dots,T\} \times \{1,\dots,N\}$ contains the indices associated with the observed graded learner response data, since some learner responses might not be observed in practice. $\Phi(z)$ denotes the *inverse probit link* function $\Phi_{\text{pro}}(z) = \int_{-\infty}^{z} \mathcal{N}(t)\,\mathrm{d}t$, where $\mathcal{N}(t) = \frac{1}{\sqrt{2\pi}} \exp(-t^2/2)$ is the probability density function (PDF) of the standard normal distribution. (Note that the inverse logit link function could also be used. However, the inverse probit link function simplifies the calculations detailed in Section 3.3.) The likelihood of an observation $Y_j^{(t)}$ can, alternatively, be written as

$$p(Y_j^{(t)} \,|\, \mathbf{c}_j^{(t)}) = \Phi\big((2Y_j^{(t)} - 1)(\bar{\mathbf{w}}_{i_j^{(t)}}^T \mathbf{c}_j^{(t)} - \mu_{i_j^{(t)}})\big),$$

an expression that we will often use in the remainder of the paper.

Following the original SPARFA framework, we impose the following model assumptions:

(A1) *The number of concepts is much smaller than the number of questions and the number of learners*: This assumption imposes a low-dimensional model on the learners' responses to questions.

(A2) *The vector* $\bar{\mathbf{w}}_i$ *is sparse*: This assumption bases on the observation that each question should only be associated with a few concepts out of all concepts in the domain of a course/assessment.

(A3) *The vector* $\bar{\mathbf{w}}_i$ *has non-negative entries*: This assumption enables one to interpret the entries in $\mathbf{c}_j$ to be the latent concept knowledge of each learner, with positive values represent high concept knowledge, while negative values represent low concept knowledge.

These assumptions are reasonable in the majority of real-world educational scenarios, alleviating the common identifiability issue (i.e., if $Z_{i,j} = \bar{\mathbf{w}}_i^T \mathbf{c}_j$, then for any orthonormal matrix $\mathbf{Q}$ with $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$, we have $Z_{i,j} = \bar{\mathbf{w}}_i^T \mathbf{Q}^T \mathbf{Q} \mathbf{c}_j = \widetilde{\bar{\mathbf{w}}}_i^T \widetilde{\mathbf{c}}_j$. Hence, estimating $\bar{\mathbf{w}}_i$ and $\mathbf{c}_j$ is, in general, non-unique up to a unitary unitary transformation. See Harman (1976) and Lan et al. (2012) for more details) of factor analysis and improving the interpretability of the variables $\bar{\mathbf{w}}_i$, $\mathbf{c}_j$, and μ_i.

2.2 Statistical model for learner knowledge state transition

The SPARFA model (1) assumes that each learner's concept knowledge remains *constant* throughout a course/assessment. Although this assumption is valid in the setting of a single test or exam, it provides limited explanatory power in analyzing the (possibly semester-long) process of a course, during which the learners' concept knowledge evolves through time. Concept knowledge state evolution can happen due to the following reasons: (i) A learner can interact with learning resources (read a section of an assigned textbook, watch a lecture video, conduct a lab experiment, or do a computer simulation), all likely to result in an increase of their concept knowledge. (ii) A learner can simply forget a learned concept, resulting in a decrease of their concept knowledge. For the sake of simplicity of exposition, we will treat the forgetting effect (Weiner and Reed (1969)) as a special learning resource that reduces learners' concept knowledge over time.

We propose a latent state transition model that models learner concept knowledge evolution between two consecutive time instances. To this end, we assume that there is a total number of M distinct learning resources. We define the mapping $m(t, j) : \{1, \ldots, T\} \times \{1, \ldots, N\} \mapsto \{1, \ldots, M\}$ from time and learner indices to learning resource indices; this information can be extracted from the learner activity log. We will use the shorthand notation $m_j^{(t-1)} = m(t-1, j)$ to denote the index of the learning resource that learner j studies between time instance $t-1$ and time instance t. Under these notations, the learner activity summary matrices $\mathbf{R}^{(t)}$ illustrated in Figure 1 are defined by $R^{(t)}_{j,m_j^{(t)}} = 1,\ \forall(t, j)$, meaning that learner j interacted with learning resource $m_j^{(t)}$ at time instance t, and 0 otherwise.

We are now ready to model the transition of learner j's latent concept knowledge state from time instance $t-1$ to t as:

$$\mathbf{c}_j^{(t)} = (\mathbf{I}_K + \mathbf{D}_{m_j^{(t-1)}})\mathbf{c}_j^{(t-1)} + \mathbf{d}_{m_j^{(t-1)}} + \boldsymbol{\epsilon}_j^{(t-1)}, \quad \boldsymbol{\epsilon}_j^{(t-1)} \sim \mathcal{N}(\mathbf{0}_K, \boldsymbol{\Gamma}_{m_j^{(t-1)}}), \tag{2}$$

where $\mathbf{I}_K$ is the $K \times K$ identity matrix; $\mathbf{D}_{m_j^{(t-1)}}$, $\mathbf{d}_{m_j^{(t-1)}}$ and $\boldsymbol{\Gamma}_{m_j^{(t-1)}}$ are latent learner concept knowledge state transition parameters, which define an affine model on the transition of the j^{th} learner's concept knowledge state by interacting with learning resource $m_j^{(t-1)}$ between time instances $t-1$ and t. $\mathbf{D}_{m_j^{(t-1)}}$ is a $K \times K$ matrix, $\mathbf{d}_{m_j^{(t-1)}}$ is a $K \times 1$ vector, and $\mathbf{0}_K$ is the K-dimensional zero vector. The covariance matrix $\boldsymbol{\Gamma}_{m_j^{(t-1)}}$ characterizes the uncertainty induced in the learner concept knowledge state transition by interacting with learning resource $m_j^{(t-1)}$. Note that (2) also has the following equivalent form:

$$p(\mathbf{c}_j^{(t)} \,|\, \mathbf{c}_j^{(t-1)}) = \mathcal{N}\left(\mathbf{c}_j^{(t)} \,|\, (\mathbf{I}_K + \mathbf{D}_{m_j^{(t-1)}})\mathbf{c}_j^{(t-1)} + \mathbf{d}_{m_j^{(t-1)}},\ \boldsymbol{\Gamma}_{m_j^{(t-1)}}\right), \tag{3}$$

where $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ represents a multivariate Gaussian distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$.

In order to reduce the number of parameters, to improve identifiability of the parameters $\mathbf{D}_{m_j^{(t-1)}}$, $\mathbf{d}_{m_j^{(t-1)}}$ and $\boldsymbol{\Gamma}_{m_j^{(t-1)}}$, and to account for real-world educational scenarios, we impose three additional assumptions on the learner knowledge state transition matrix $\mathbf{D}_{m_j^{(t-1)}}$:

(A4) $\mathbf{D}_{m_j^{(t-1)}}$ *is lower triangular*: This assumption means that, the k^{th} entry in the learner concept knowledge vector $\mathbf{c}_j^{(t)}$ is only influenced by the the $1^{\text{st}}, \ldots, (k-1)^{\text{th}}$ entry in $\mathbf{c}_j^{(t-1)}$. As a result, the upper entries in $\mathbf{c}_j^{(t-1)}$ represent pre-requisite concepts that are covered early in the course, while lower entries represent advanced concepts that are covered towards the end of the course. Using this assumption, it is possible to extract prerequisite relationships among concepts purely from learner response data.

(A5) $\mathbf{D}_{m_j^{(t-1)}}$ *has non-negative entries*: This assumption ensures that having low concept knowledge at time instance $t-1$ (negative entries in $\mathbf{c}_j^{(t-1)}$) does not result in high concept knowledge at time instance t (positive entries in $\mathbf{c}_j^{(t)}$).

(A6) $\mathbf{D}_{m_j^{(t-1)}}$ *is sparse*: This assumption amounts for the observation that learning resources typically only cover a small subset of concepts among all concepts covered in a course.

In contrast to the learner concept knowledge transition matrix $\mathbf{D}_{m_j^{(t-1)}}$, we do not impose sparsity or non-negativity properties on the intrinsic learner concept knowledge state transition vector $\mathbf{d}_{m_j^{(t-1)}}$ in (2); large, positive values in $\mathbf{d}_{m_j^{(t-1)}}$ represent learning resources with good quality that boost learners' concept knowledge, while small, negative values in $\mathbf{d}_{m_j^{(t-1)}}$ represent learning resources that reduce learners' concept knowledge. This setting enables our framework to model cases of poorly designed, misleading, or off-topic learning resources

that distract or confuse learners. Note that the forgetting effect can also be modeled as a learning resource with negative entries in $\mathbf{d}_{m_j^{(t-1)}}$.

To further reduce the number of parameters, the covariance matrix $\mathbf{\Gamma}_{m_j^{(t-1)}}$ is assumed to be diagonal, implying that the uncertainties of learning resources on learners' knowledge states are not correlated among different concepts. This assumption is mainly made for simplicity; the analysis of more evolved models is left for future work.

In the next section, we will describe how to estimate the learners' concept knowledge state vectors $\mathbf{c}_j^{(t)}, \forall t, j$, given observed graded learner responses $\mathbf{Y}_j^{(t)}, (t,j) \in \Omega_{\text{obs}}$, and all parameters $\mathbf{D}_{m_j^{(t-1)}}, \mathbf{d}_{m_j^{(t-1)}}, \mathbf{\Gamma}_{m_j^{(t-1)}}, \bar{\mathbf{w}}_{i_j^{(t)}}, \mu_{i_j^{(t)}}, \forall t, j$, i.e., how to perform time-varying learning analytics. Then, in Section 4, we will introduce methods to estimate these parameters and thus analyze the quality and content organization of all learning resources and questions, i.e., performing *content analytics*.

3. Time-Varying Learning Analytics

We now introduce a message passing-based approximate Kalman filtering approach for learner concept knowledge tracing. Since the observed data is binary-valued graded learner responses to questions, we cannot simply use common Kalman filter methods that assume Gaussian observation models. We start with a brief review of Kalman filtering and smoothing, and then introduce the necessary approximations in the Kalman filtering approach to estimate latent learner concept knowledge states at all time instances. For simplicity, we will drop the learner index j in this section, i.e., quantities $\mathbf{D}_{m_j^{(t-1)}}$ and $\mathbf{d}_{m_j^{(t-1)}}$ are replaced by $\mathbf{D}_{m^{(t-1)}}$ and $\mathbf{d}_{m^{(t-1)}}$. Moreover, we use the shorthand notation $\overline{\mathbf{D}}_{m^{(t-1)}}$ for the quantity $\mathbf{I}_K + \mathbf{D}_{m^{(t-1)}}$.

3.1 Kalman filtering

Kalman filtering (Kalman (1960); Haykin (2001)) solves a key inference problem in linear dynamical systems (LDS), where the system consist of a series of continuous latent state variables with Markovian state transition property and a Gaussian observation model. The following derivations briefly summarize the results in Minka (1999).

The Markov chain consist of a series of T latent state variables $\mathbf{c}^{(t)}, t = 1, \ldots, T$, and observations $\mathbf{y}^{(t)}, t = 1, \ldots, T$. Due to the Markovian property of the system, the joint probability of all latent states and all observations can be factorized as

$$p(\mathbf{c}^{(1)}, \ldots, \mathbf{c}^{(T)}, \mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(T)}) = p(\mathbf{c}^{(1)})\, p(\mathbf{y}^{(1)} \,|\, \mathbf{c}^{(1)}) \prod_{t=2}^{T} p(\mathbf{c}^{(t)} \,|\, \mathbf{c}^{(t-1)})\, p(\mathbf{y}^{(t)} \,|\, \mathbf{c}^{(t)}).$$

A visualization of the dynamical system as a factor graph (Kschischang et al. (2001); Loeliger (2004)) is shown in Figure 2.

The inference algorithm, which estimates the vectors $\mathbf{c}^{(t)}, \forall t$, based on the observations $\mathbf{y}^{(t)}, \forall t$, consist of two parts. First, a forward message passing phase (i.e., the Kalman filtering phase) is performed. Then, using estimates obtained during the Kalman filtering phase, a backward message passing phase (often referred to as Kalman smoothing or Rauch-Tung-Streibel (RTS) smoothing) is also performed.

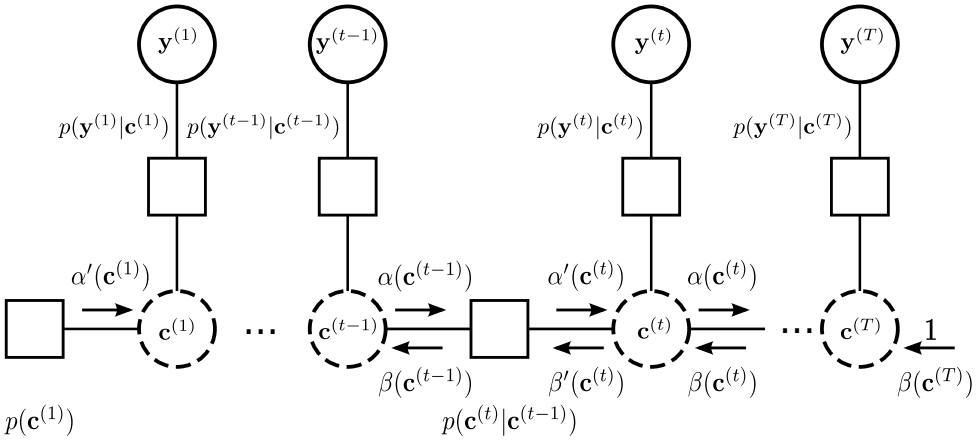


Figure 2: Factor graph message passing scheme for the inference of a set of T latent state variables with Markovian transition properties from (possibly noisy) observations.

In the forward message passing phase, the goal is to estimate latent state variables $\mathbf{c}^{(t)}$ based on the previous observations $\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(t)}$. In other words, the value of interest is $p(\mathbf{c}^{(t)} \,|\, \mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(t)})$, $\forall t$. This quantity can be obtained via a message passing scheme outlined in Figure 2. Specifically, by starting at $t = 1$, the incoming message to variable node $\mathbf{c}^{(1)}$ is given by $\alpha'(\mathbf{c}^{(1)}) = p(\mathbf{c}^{(1)})$. The outgoing message from variable node $\mathbf{c}^{(1)}$ to factor node $p(\mathbf{c}^{(2)} \,|\, \mathbf{c}^{(1)})$ is then given by

$$\alpha(\mathbf{c}^{(1)}) = \alpha'(\mathbf{c}^{(1)})\, p(\mathbf{y}^{(1)} \,|\, \mathbf{c}^{(1)}) = p(\mathbf{c}^{(1)})\, p(\mathbf{y}^{(1)} \,|\, \mathbf{c}^{(1)}) = b_1\, p(\mathbf{c}^{(1)} \,|\, \mathbf{y}^{(1)}),$$

according to Bayes rule, where $b_1 = p(\mathbf{y}^{(1)})$ is the *scaling factor*. Recursively following these rules, the outgoing message $\alpha(\mathbf{c}^{(t-1)})$ from variable node $\mathbf{c}^{(t-1)}$ to the factor node $p(\mathbf{c}^{(t)} \,|\, \mathbf{c}^{(t-1)})$ at time t is given by

$$\alpha(\mathbf{c}^{(t-1)}) = \left(\textstyle\prod_{\tau=1}^{t-1} b^{(\tau)}\right) p(\mathbf{c}^{(t-1)} \,|\, \mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(t-1)}).$$

The outgoing message $\alpha'(\mathbf{c}^{(t)})$ from factor node $p(\mathbf{c}^{(t)} \,|\, \mathbf{c}^{(t-1)})$ to variable node $\mathbf{c}^{(t)}$ is given by

$$\alpha'(\mathbf{c}^{(t)}) = \int \alpha(\mathbf{c}^{(t-1)}) p(\mathbf{c}^{(t)} \,|\, \mathbf{c}^{(t-1)}) \mathrm{d}\mathbf{c}^{(t-1)} = \left(\textstyle\prod_{\tau=1}^{t-1} b^{(\tau)}\right)\, p(\mathbf{c}^{(t)} \,|\, \mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(t-1)}).$$

The outgoing message $\alpha(\mathbf{c}^{(t)})$ from variable node $\mathbf{c}^{(t)}$ is given by

$$\alpha(\mathbf{c}^{(t)}) = \alpha'(\mathbf{c}^{(t)})\, p(\mathbf{y}^{(t)} \,|\, \mathbf{c}^{(t)}) = \left(\textstyle\prod_{\tau=1}^{t} b^{(\tau)}\right) p(\mathbf{c}^{(t)} \,|\, \mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(t)}),$$

where $b^{(t)} = p(\mathbf{y}^{(t)} \,|\, \mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(t-1)})$. We can see that a scaled version of $\alpha(\mathbf{c}^{(t)})$, $\widehat{\alpha}(\mathbf{c}^{(t)}) = \frac{\alpha(\mathbf{c}^{(t)})}{\prod_{\tau=1}^{t} b^{(\tau)}} = p(\mathbf{c}^{(t)} \,|\, \mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(t)})$, is exactly the value of interest.

The derivations above show that $\widehat{\alpha}(\mathbf{c}^{(t)})$ can be obtained in recursive fashion via

$$b^{(t)}\,\widehat{\alpha}(\mathbf{c}^{(t)}) = p(\mathbf{y}^{(t)}\,|\,\mathbf{c}^{(t)}) \int p(\mathbf{c}^{(t)}\,|\,\mathbf{c}^{(t-1)})\,\widehat{\alpha}(\mathbf{c}^{(t-1)})\mathrm{d}\mathbf{c}^{(t-1)}. \tag{4}$$

The key for obtaining a tractable and efficient estimator for $p(\mathbf{c}^{(t)}\,|\,\mathbf{y}^{(1)},\ldots,\mathbf{y}^{(t)})$ is that the transition probability $p(\mathbf{c}^{(t)}\,|\,\mathbf{c}^{(t-1)})$ and the observation likelihood $p(\mathbf{y}^{(t)}\,|\,\mathbf{c}^{(t)})$ satisfy certain properties such that the messages $\widehat{\alpha}(\mathbf{c}^{(t)})$ and $\widehat{\alpha}(\mathbf{c}^{(t-1)})$ take on the same functional form, just with different parameters. A LDS is a special case in which the transition probability and the observation likelihood are (multivariate) Gaussians of are of the following form:

$$\begin{aligned}
p(\mathbf{c}^{(t)}\,|\,\mathbf{c}^{(t-1)}) &= \mathcal{N}(\mathbf{c}^{(t)}\,|\,\overline{\mathbf{D}}_{m^{(t-1)}}\mathbf{c}^{(t-1)} + \mathbf{d}_{m^{(t-1)}}, \mathbf{\Gamma}_{m^{(t-1)}}),\\
p(\mathbf{y}^{(t)}\,|\,\mathbf{c}^{(t)}) &= \mathcal{N}(\mathbf{y}^{(t)}\,|\,\mathbf{W}_{i^{(t)}}\mathbf{c}^{(t)}, \mathbf{\Sigma}_{i^{(t)}}).
\end{aligned}$$

Here, $\mathbf{\Gamma}_{m^{(t-1)}}$ is the covariance matrix for state transition, $\mathbf{W}_{i^{(t)}}$ is the measurement matrix, and $\mathbf{\Sigma}_{i^{(t)}}$ is the covariance matrix for the multivariate observation of the system. In order for the functional form of the messages to stay the same over time, the messages are also Gaussian, i.e., $\widehat{\alpha}(\mathbf{c}^{(t)}) \sim \mathcal{N}(\mathbf{c}^{(t)}\,|\,\mathbf{m}^{(t)}, \mathbf{V}^{(t)})$. Under these conditions, the forward message passing recursion (4) takes on a compact form

$$b^{(t)}\,\widehat{\alpha}(\mathbf{c}^{(t)}) = \mathcal{N}\Big(\mathbf{c}^{(t)}\,|\,\mathbf{m}^{(t)}, \mathbf{V}^{(t)}\Big), \tag{5}$$

with the parameters $b^{(t)}$, $\mathbf{m}^{(t)}$ and $\mathbf{V}^{(t)}$ given by

$$\begin{aligned}
\mathbf{m}^{(t)} &= \overline{\mathbf{D}}_{m^{(t-1)}}\mathbf{m}^{(t-1)} + \mathbf{d}_{m^{(t-1)}} + \mathbf{K}^{(t)}\left(\mathbf{y}^{(t)} - \mathbf{W}_{i^{(t)}}\left(\overline{\mathbf{D}}_{m^{(t-1)}}\mathbf{m}^{(t-1)} + \mathbf{d}_{m^{(t-1)}}\right)\right),\\
\mathbf{V}^{(t)} &= \left(\mathbf{I} - \mathbf{K}^{(t)}\mathbf{W}_{i^{(t)}}\right)\mathbf{P}^{(t-1)},\\
b^{(t)} &= \mathcal{N}\left(\mathbf{y}^{(t)}\,|\,\mathbf{W}_{i^{(t)}}\left(\overline{\mathbf{D}}_{m^{(t-1)}}\mathbf{m}^{(t-1)} + \mathbf{d}_{m^{(t-1)}}\right), \mathbf{W}_{i^{(t)}}\mathbf{P}^{(t-1)}\mathbf{W}_{i^{(t)}}^T + \mathbf{\Sigma}_{i^{(t)}}\right),
\end{aligned}$$

in which the matrices $\mathbf{K}^{(t)}$ and $\mathbf{P}^{(t-1)}$ are given by

$$\begin{aligned}
\mathbf{K}^{(t)} &= \mathbf{P}^{(t-1)}\mathbf{W}_{i^{(t)}}^T\left(\mathbf{W}_{i^{(t)}}\mathbf{P}^{(t-1)}\mathbf{W}_{i^{(t)}}^T + \mathbf{\Sigma}_{i^{(t)}}\right)^{-1},\\
\mathbf{P}^{(t-1)} &= \overline{\mathbf{D}}_{m^{(t-1)}}\mathbf{V}^{(t-1)}\overline{\mathbf{D}}_{m^{(t-1)}}^T + \mathbf{\Gamma}_{m^{(t-1)}}.
\end{aligned}$$

The recursion starts with a prior $p(\mathbf{c}^{(1)}) = \mathcal{N}(\mathbf{c}^{(1)}\,|\,\mathbf{m}^{(0)}, \mathbf{V}^{(0)})$, and

$$\begin{aligned}
\mathbf{m}^{(1)} &= \mathbf{m}^{(0)} + \mathbf{K}^{(1)}\left(\mathbf{y}^{(1)} - \mathbf{W}_{i^{(1)}}\mathbf{m}^{(0)}\right),\\
\mathbf{V}^{(1)} &= \left(\mathbf{I}_K - \mathbf{K}^{(1)}\mathbf{W}_{i^{(1)}}\right)\mathbf{V}^{(0)},\\
\mathbf{K}^{(1)} &= \mathbf{V}^{(0)}\mathbf{W}_{i^{(1)}}^T\left(\mathbf{W}_{i^{(1)}}\mathbf{V}^{(0)}\mathbf{W}_{i^{(1)}}^T + \mathbf{\Sigma}_{i^{(1)}}\right)^{-1},\\
b^{(1)} &= \mathcal{N}\left(\mathbf{y}^{(1)}\,|\,\mathbf{W}_{i^{(1)}}\mathbf{m}^{(0)}, \mathbf{W}_{i^{(1)}}\mathbf{V}^{(0)}\mathbf{W}_{i^{(1)}}^T + \mathbf{\Sigma}_{i^{(1)}}\right).
\end{aligned}$$

The initial prior mean and variance for $\mathbf{c}^{(1)}$ are assumed to be $\mathbf{m}^{(0)} = \mathbf{0}_K$ and $\mathbf{V}^{(0)} = \sigma_0^2\mathbf{I}_K$ in what follows.

3.2 Kalman smoothing

As detailed above, Kalman filtering can be utilized to obtain $p(\mathbf{c}^{(t)} \,|\, \mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(t)})$, an estimate on the latent state at time instance t, given all observations $\mathbf{y}^{(\tau)}$ for $\tau < t$. This estimate is the value of interest for a variety of real-time tracking applications, since decisions have to be made based on all available observations up to a certain time instance. However, in our application, one could also use observations at $\tau \geq t$ to obtain a better estimate of the latent state at time instance t. In other words, the value of interest is now $p(\mathbf{c}^{(t)} \,|\, \mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(T)})$. In order to estimate this value, a set of backward recursions similar to the set of forward recursions (4) can be used.

Following the backward message passing scheme described in Loeliger (2004), the backwards message starts with a "one" message going into variable node $\mathbf{c}^{(T)}$: $\beta(\mathbf{c}^{(T)}) = \mathbf{1}$ (as shown in Figure 2). Then, the outgoing message from variable node $\mathbf{c}^{(T)}$ into factor node $p(\mathbf{c}^{(T)} \,|\, \mathbf{c}^{(T-1)})$ is

$$\beta'(\mathbf{c}^{(T)}) = p(\mathbf{y}^{(T)} \,|\, \mathbf{c}^{(T)}),$$

and the outgoing message from factor node $p(\mathbf{c}^{(T)} \,|\, \mathbf{c}^{(T-1)})$ into variable node $\mathbf{c}^{(T-1)}$ is

$$\beta(\mathbf{c}^{(T-1)}) = \int p(\mathbf{c}^{(T)} \,|\, \mathbf{c}^{(T-1)}) p(\mathbf{y}^{(T)} \,|\, \mathbf{c}^{(T)}) \mathrm{d}\mathbf{c}^{(T)} = p(\mathbf{y}^{(T)} \,|\, \mathbf{c}^{(T-1)}).$$

Following this convention, we obtain the following recursion:

$$\beta(\mathbf{c}^{(t-1)}) = \int p(\mathbf{c}^{(t)} \,|\, \mathbf{c}^{(t-1)}) p(\mathbf{y}^{(t)} \,|\, \mathbf{c}^{(t)}) \beta(\mathbf{c}^{(t)}) \mathrm{d}\mathbf{c}^{(t)} = p(\mathbf{y}^{(t)}, \ldots, \mathbf{y}^{(T)} \,|\, \mathbf{c}^{(t-1)}),$$

where we have implicitly used the Markovian properties of the latent state variables. Now, the marginal distribution of latent state variables $\mathbf{c}^{(t)}$ can be written as a product of the incoming messages into variable node $\mathbf{c}^{(t)}$ from both forward and backward recursions, i.e.,

$$p(\mathbf{c}^{(t)} \,|\, \mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(T)}) = \frac{p(\mathbf{c}^{(t)} \,|\, \mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(T)})\, p(\mathbf{y}^{(t+1)}, \ldots, \mathbf{y}^{(T)} \,|\, \mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(t)})}{p(\mathbf{y}^{(t+1)}, \ldots, \mathbf{y}^{(T)} \,|\, \mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(t)})} = \widehat{\alpha}(\mathbf{c}^{(t)}) \widehat{\beta}(\mathbf{c}^{(t)}),$$

where $\widehat{\beta}(\mathbf{c}^{(t)}) = \frac{\beta(\mathbf{c}^{(t)})}{\prod_{\tau=t+1}^{T} b^{(\tau)}}$ is a scaled version of $\beta(\mathbf{c}^{(t)})$. Now, the backward recursion is as follows:

$$b^{(t)}\, \widehat{\beta}(\mathbf{c}^{(t-1)}) = \int_{\mathbf{c}^{(t)}} p(\mathbf{c}^{(t)} \,|\, \mathbf{c}^{(t-1)}) p(\mathbf{y}^{(t)} \,|\, \mathbf{c}^{(t)}) \widehat{\beta}(\mathbf{c}^{(t)}) \mathrm{d}\mathbf{c}^{(t)}. \tag{6}$$

Although it is possible to obtain a backward recursion for $\widehat{\beta}(\mathbf{c}^{(t)})$, the common approach uses a recursion directly on $\widehat{\alpha}(\mathbf{c}^{(t)}) \widehat{\beta}(\mathbf{c}^{(t)})$ to obtain the value of interest $p(\mathbf{c}^{(t)} \,|\, \mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(T)})$. By multiplying both sides of the equation (6) by $\widehat{\alpha}(\mathbf{c}^{(t-1)})$, we obtain

$$\widehat{\alpha}(\mathbf{c}^{(t-1)}) \widehat{\beta}(\mathbf{c}^{(t-1)}) = \widehat{\alpha}(\mathbf{c}^{(t-1)}) \int_{\mathbf{c}^{(t)}} p(\mathbf{c}^{(t)} \,|\, \mathbf{c}^{(t-1)}) p(\mathbf{y}^{(t)} \,|\, \mathbf{c}^{(t)}) \frac{\widehat{\alpha}(\mathbf{c}^{(t)}) \widehat{\beta}(\mathbf{c}^{(t)})}{b^{(t)}\, \widehat{\alpha}(\mathbf{c}^{(t)})} \mathrm{d}\mathbf{c}^{(t)},$$

which can be computed recursively as a backward message passing process, given the estimates (5) following the completion of the forward message passing process detailed in Section 3.1. Specifically, for a LDS, the recursions take the form:

$$\widehat{\alpha}(\mathbf{c}^{(t-1)})\widehat{\beta}(\mathbf{c}^{(t-1)}) = \mathcal{N}(\mathbf{c}^{(t-1)} \,|\, \widehat{\mathbf{m}}^{(t-1)}, \widehat{\mathbf{V}}^{(t-1)}) \tag{7}$$

with the parameters $\widehat{\mathbf{m}}^{(t-1)}$ and $\widehat{\mathbf{V}}^{(t-1)}$ given by

$$\begin{aligned}
\widehat{\mathbf{m}}^{(t-1)} &= \mathbf{m}^{(t-1)} + \mathbf{J}^{(t-1)} \left(\widehat{\mathbf{m}}^{(t)} - \overline{\mathbf{D}}_{m^{(t-1)}} \mathbf{m}^{(t-1)} - \mathbf{d}_{m^{(t-1)}} \right), \\
\widehat{\mathbf{V}}^{(t-1)} &= \mathbf{V}^{(t-1)} + \mathbf{J}^{(t-1)} \left(\widehat{\mathbf{V}}^{(t)} - \mathbf{P}^{(t-1)} \right) (\mathbf{J}^{(t-1)})^T, \\
\mathbf{J}^{(t-1)} &= \mathbf{V}^{(t-1)} (\overline{\mathbf{D}}_{m^{(t-1)}})^T (\mathbf{P}^{(t-1)})^{-1}.
\end{aligned}$$

We start the recursion with $\widehat{\mathbf{m}}^{(T)} = \mathbf{m}^{(T)}$ and $\widehat{\mathbf{V}}^{(T)} = \mathbf{V}^{(T)}$, since $\beta(\mathbf{c}^{(T)}) = \mathbf{1}$.

In the above derivations, we have assumed that $\mathbf{y}^{(t)}$ is observed for all t. If $\mathbf{y}^{(t)}$ is unobserved, then the message passing scheme will simply have $\alpha(\mathbf{c}^{(t)}) = \alpha'(\mathbf{c}^{(t)})$ and $\beta'(\mathbf{c}^{(t)}) = \beta(\mathbf{c}^{(t)})$ instead, while the rest of the recursions remain unaffected.

3.3 Approximate Kalman filtering for learner concept knowledge tracing

The basic Kalman filtering and smoothing approaches (Equations (5) and (7)) are only suitable for applications with a Gaussian latent state transition model and a Gaussian observation model, while the forward and backward recursions (Equations (4) and (6)) holds for arbitrary state transition and observation models. When attempting to trace latent learner concept knowledge states under the SPARFA model, it is not possible to make Gaussian observations of these states. Concretely, we have only binary-valued graded learner responses as our observations in the present application. We will now detail approximations that have to be made to enable the estimation of latent learner concept knowledge states under our model.

As introduced in Section 2, the observation model at time t is given by (1) and the state transition model is given by (3). Therefore, the recursion formula for the forward message passing process (4) becomes

$$\begin{aligned}
b^{(t)}\, \widehat{\alpha}(\mathbf{c}^{(t)}) &= p(Y^{(t)} \,|\, \mathbf{c}^{(t)}) \int p(\mathbf{c}^{(t)} \,|\, \mathbf{c}^{(t-1)})\, \widehat{\alpha}(\mathbf{c}^{(t-1)}) \mathrm{d}\mathbf{c}^{(t)} \\
&= \Phi\left(\left(2Y^{(t)} - 1 \right) \left(\bar{\mathbf{w}}_{i^{(t)}}^T \mathbf{c}^{(t)} - \mu_{i^{(t)}} \right) \right) \int \mathcal{N}\left(\mathbf{c}^{(t)} | \overline{\mathbf{D}}_{m^{(t-1)}} \mathbf{c}^{(t-1)} + \mathbf{d}_{m^{(t-1)}}, \mathbf{\Gamma}_{m^{(t-1)}} \right) \\
&\quad \mathcal{N}\left(\mathbf{c}^{(t-1)} | \mathbf{m}^{(t-1)}, \mathbf{V}^{(t-1)} \right) \mathrm{d}\mathbf{c}^{(t)} \\
&= \Phi\left(\left(2Y^{(t)} - 1 \right) \left(\bar{\mathbf{w}}_{i^{(t)}}^T \mathbf{c}^{(t)} - \mu_{i^{(t)}} \right) \right) \mathcal{N}(\mathbf{c}^{(t)} | \overline{\mathbf{D}}_{m^{(t-1)}} \mathbf{m}^{(t-1)} + \mathbf{d}_{m^{(t-1)}}, \\
&\quad \overline{\mathbf{D}}_{m^{(t-1)}} \mathbf{V}^{(t-1)} \overline{\mathbf{D}}_{m^{(t-1)}}^T + \mathbf{\Gamma}_{m^{(t-1)}}) \\
&= \Phi\left(\left(2Y^{(t)} - 1 \right) \left(\bar{\mathbf{w}}_{i^{(t)}}^T \mathbf{c}^{(t)} - \mu_{i^{(t)}} \right) \right) \mathcal{N}\left(\mathbf{c}^{(t)} | \widetilde{\mathbf{m}}^{(t)}, \widetilde{\mathbf{V}}^{(t)} \right),
\end{aligned} \tag{8}$$

where we used a *tilde* to denote the mean and covariance of the messages $\alpha'(\mathbf{c}^{(t-1)})$.

Equation (8) shows that $\widehat{\alpha}(\mathbf{c}^{(t)})$ is no longer Gaussian even if $\widehat{\alpha}(\mathbf{c}^{(t-1)})$ is Gaussian, under the probit binary observation model. Thus, the closed-form updates in (5) and (7) can no longer be applied. Therefore, we have to perform an approximate message passing approach within the Kalman filtering framework to arrive at a tractable estimator of $\mathbf{c}^{(t)}$. In order to do so, a number of approaches has been proposed to approximate $\widehat{\alpha}(\mathbf{c}^{(t)})$ by a Gaussian distribution $\mathcal{N}\Big(\mathbf{c}^{(t)}\,|\,\overline{\mathbf{m}}^{(t)}, \overline{\mathbf{V}}^{(t)}\Big)$; here, the *bar* on the variables denote the means and covariances of the *approximated* Gaussian messages. These approaches include the extended Kalman filter (EKF) (Jazwinski (1970); Maybeck (1979); Einicke and White (1999)), which uses a linear approximation of the likelihood term around the point $\widetilde{\mathbf{m}}^{(t)}$, and thus reduce the non-Gaussian observation model to a Gaussian one; the unscented Kalman filter (UKF) (Julier and Uhlmann (1997); Wan and Van Der Merwe (2000)), which uses the unscented transform (UT) to create a set of sigma vectors from $p(\mathbf{c}^{(t-1)})$ and uses them to approximate the mean and covariance of $\widehat{\alpha}(\mathbf{c}^{(t)})$ after the non-Gaussian observation; and Laplace approximations (Wolfinger (1993); Rasmussen and Williams (2006)), which use an iterative algorithm to find the mode of $\widehat{\alpha}(\mathbf{c}^{(t)})$ and the Hessian at the mode in order to approximate the mean and covariance of the approximated Gaussian messages. We will employ an approximation approach introduced in the expectation propagation (EP) literature (Minka (2001)).

It is known that the specific values for $\overline{\mathbf{m}}^{(t)}$ and $\overline{\mathbf{V}}^{(t)}$ that minimize the Kullback-Leibler (KL) divergence between $\mathcal{N}\Big(\mathbf{c}^{(t)}|\overline{\mathbf{m}}^{(t)}, \overline{\mathbf{V}}^{(t)}\Big)$ and a target distribution $q(\mathbf{c})$ are the first and second moments of $q(\mathbf{c})$ Rasmussen and Williams (2006). Luckily, for the probit observation model $p(Y^{(t)}\,|\,\mathbf{c}^{(t)}) = \Phi\Big(\big(2Y^{(t)}-1\big)\Big(\bar{\mathbf{w}}_{i^{(t)}}^T\mathbf{c}^{(t)} - \mu_{i^{(t)}}\Big)\Big)$, $\overline{\mathbf{m}}^{(t)}$, $\overline{\mathbf{V}}^{(t)}$ and $b^{(t)}$ have closed-form expressions (see Section 8 for the details):

$$
\begin{aligned}
\overline{\mathbf{m}}^{(t)} &= \widetilde{\mathbf{m}}^{(t)} + \Big(2Y^{(t)}-1\Big)\frac{\widetilde{\mathbf{V}}^{(t)}\bar{\mathbf{w}}_{i^{(t)}}}{\sqrt{1+\bar{\mathbf{w}}_{i^{(t)}}^T\widetilde{\mathbf{V}}^{(t)}\bar{\mathbf{w}}_{i^{(t)}}}}\frac{\mathcal{N}(z)}{\Phi(z)},\\
\overline{\mathbf{V}}^{(t)} &= \widetilde{\mathbf{V}}^{(t)} - \frac{\widetilde{\mathbf{V}}^{(t)}\bar{\mathbf{w}}_{i^{(t)}}\bar{\mathbf{w}}_{i^{(t)}}^T\widetilde{\mathbf{V}}^{(t)}}{1+\bar{\mathbf{w}}_{i^{(t)}}^T\widetilde{\mathbf{V}}^{(t)}\bar{\mathbf{w}}_{i^{(t)}}}\left(z+\frac{\mathcal{N}(z)}{\Phi(z)}\right)\frac{\mathcal{N}(z)}{\Phi(z)},\\
b^{(t)} &= \Phi(z),
\end{aligned}
\tag{9}
$$

with

$$
z = \Big(2Y^{(t)}-1\Big)\frac{\bar{\mathbf{w}}_{i^{(t)}}^T\widetilde{\mathbf{m}}^{(t)} - \mu_{i^{(t)}}}{\sqrt{1+\bar{\mathbf{w}}_{i^{(t)}}^T\widetilde{\mathbf{V}}^{(t)}\bar{\mathbf{w}}_{i^{(t)}}}},
$$

and $\widetilde{\mathbf{m}}^{(t)}$ and $\widetilde{\mathbf{V}}^{(t)}$ are given by (8).

SPARFA studied two different inverse link functions for analyzing binary-valued graded learner responses: the inverse probit link function and the inverse logit link function. In this application, the inverse probit link function is preferred over the inverse logit link function, due to the existence of the closed-form first and second moments described above. The inverse logit link function is not preferred as such convenient close-form expressions do not exist. Therefore, we will focus on the inverse probit link function in the sequel.

Armed with the efficient approximation (9), the forward Kalman filtering message passing scheme described in Section 3.1 can be applied to the problem at hand; the backward Kalman smoothing message passing scheme described in Section 3.2 remains unchanged. Using these recursions, estimates of the desired quantities $p(\mathbf{c}^{(t)} \,|\, \mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(T)})$ can be computed efficiently, providing a way for learner concept knowledge tracing under the model (1).

4. Content Analytics

So far, we have described an approximate Kalman filtering and smoothing approach for learner concept knowledge tracing, i.e., to estimate $p(\mathbf{c}_j^{(t)} \mid \mathbf{y}_j^{(1)}, \ldots, \mathbf{y}_j^{(T)})$, $\forall t, j$. The proposed method is only able to retrieve these estimates given both the observed binary graded learner responses $Y_j^{(t)}$, $\forall t, j$, and all learner initial knowledge parameters $\mathbf{m}_j^{(0)}, \mathbf{V}_j^{(0)}$, $\forall j$, all learner concept knowledge state transition parameters $\mathbf{D}_m$, $\mathbf{d}_m$, and $\mathbf{\Gamma}_m$, $\forall m$, and all question parameters, $\bar{\mathbf{w}}_i$ and μ_i, $\forall i$.

However, in a typical PLS, these parameters are unknown, in general, and need to be estimated from the observed data. Hence, we now detail a set of convex optimization-based techniques to estimate the parameters $\mathbf{m}_j^{(0)}, \mathbf{V}_j^{(0)}$, $\forall j$, $\mathbf{D}_m$, $\mathbf{d}_m$, and $\mathbf{\Gamma}_m$, $\forall m$, and $\bar{\mathbf{w}}_i$, μ_i, $\forall i$, given the estimates of the latent learner concept knowledge states $\mathbf{c}_j^{(t)}$ obtained from the approximate Kalman filtering approach described in Section 3. Since the estimates of $\mathbf{c}_j^{(t)}$ are distributions rather than point estimates, SPARFA-Trace jointly traces learner concept knowledge and estimates learner, learning resource, and question-dependent parameters, using an expectation-maximization (EM) approach.

4.1 SPARFA-Trace: An EM algorithm for parameter estimation

EM has been widely used in the Kalman filtering framework to estimate the parameters of interest in the system (see Haykin (2001) and (Bishop and Nasrabadi, 2006, Chap. 13) for more details) due to numerous practical advantages (Roweis and Ghahramani (2001)). SPARFA-Trace performs parameter estimation in an iterative fashion in the EM framework. All parameters are initialized to random initial values, and then, each iteration of the algorithm consist of two phases: (i) the current parameter estimates are used to estimate the latent state distributions $p(\mathbf{c}_j^{(t)} \mid \mathbf{y}_j^{(1)}, \ldots, \mathbf{y}_j^{(T)})$, $\forall t, j$; (ii), these latent state estimates are then used to maximize the expected joint log-likelihood of all the observed and latent state variables, i.e.,

$$\underset{\mathbf{m}_j^{(0)}, \mathbf{V}_j^{(0)}, \forall j, \mathbf{D}_m, \mathbf{d}_m, \mathbf{\Gamma}_m, \forall m, \bar{\mathbf{w}}_i, \mu_i, \forall i}{\text{maximize}} \quad \sum_{j=1}^{N} \mathbb{E}_{\mathbf{c}_j^{(1)}} \left[\log \, p(\mathbf{c}_j^{(1)} \big| \, \mathbf{m}_j^{(0)}, \mathbf{V}_j^{(0)}) \right] + \sum_{t=2}^{T} \sum_{j=1}^{N} \tag{10}$$

$$\mathbb{E}_{\mathbf{c}_j^{(t-1)}, \mathbf{c}_j^{(t)}} \Big[\log p(\mathbf{c}_j^{(t)} \,|\, \mathbf{c}_j^{(t-1)}, \mathbf{D}_{m_j^{(t-1)}}, \mathbf{d}_{m_j^{(t-1)}}, \mathbf{\Gamma}_{m_j^{(t-1)}}) \Big] + \sum_{(t,j) \in \Omega_{\text{obs}}} \mathbb{E}_{\mathbf{c}_j^{(t)}} \big[\log p(Y_j^{(t)} \,|\, \mathbf{c}_j^{(t)}, \bar{\mathbf{w}}_{i_j^{(t)}}, \mu_{i_j^{(t)}}) \big],$$

in order to obtain new (and hopefully improved) parameter estimates. SPARFA-Trace alternates between these two phases until convergence, i.e., a maximum number of iterations is reached or the change in the estimated parameters between two consecutive iterations falls below a given threshold.

4.2 Estimating the learner initial knowledge parameters

We start with the estimation method for the learner initial knowledge parameters $\mathbf{m}_j^{(0)}, \mathbf{V}_j^{(0)}, \forall j$. To this end, we minimize the expected negative log-likelihood for the j^{th} learner:

$$\mathbb{E}_{\mathbf{c}_j^{(1)}}\big[-\log p(\mathbf{c}_j^{(1)} \,|\, \mathbf{m}_j^{(0)}, \mathbf{V}_j^{(0)})\big] = \frac{1}{2}\log|\mathbf{V}_j^{(0)}| + \mathbb{E}_{\mathbf{c}_j^{(1)}}\left[\frac{1}{2}(\mathbf{c}_j^{(1)} - \mathbf{m}_j^{(0)})^T(\mathbf{V}_j^{(0)})^{-1}(\mathbf{c}_j^{(1)} - \mathbf{m}_j^{(0)})\right],$$

where $|\mathbf{V}_j^{(0)}|$ denotes the determinant of the covariance matrix $\mathbf{V}_j^{(0)}$. Since we do not impose constraints on $\mathbf{m}_j^{(0)}$ and $\mathbf{V}_j^{(0)}$, these estimates can be obtained as

$$\mathbf{m}_j^{(0)} = \mathbb{E}_{\mathbf{c}_j^{(1)}}\left[\mathbf{c}_j^{(1)}\right] = \widehat{\mathbf{m}}_j^{(1)} \quad \text{and} \quad \mathbf{V}_j^{(0)} = \mathbb{E}_{\mathbf{c}_j^{(1)}}\left[(\mathbf{c}_j^{(1)} - \widehat{\mathbf{m}}_j^{(1)})(\mathbf{c}_j^{(1)} - \widehat{\mathbf{m}}_j^{(1)})^T\right] = \widehat{\mathbf{V}}_j^{(1)},$$

where the estimates $\widehat{\mathbf{m}}_j^{(1)}$ and $\widehat{\mathbf{V}}_j^{(1)}$ are obtained from the Kalman smoothing recursions (7) in Section 3.2.

4.3 Estimating the learner concept knowledge state transition parameters

Now we estimate the latent learner concept knowledge state transition (i.e., learning resource) parameters $\mathbf{D}_m$, $\mathbf{d}_m$, and $\mathbf{\Gamma}_m$, $\forall m$. To this end, define $\mathcal{M}^m$ as the set containing time and learner indices (t, j), indicating that learner j studies the m^{th} learning resource between time instances $t-1$ and t. With this definition, we aim to minimize the expected negative log-likelihood

$$\begin{aligned}
&\sum_{t,j:(t,j)\in\mathcal{M}^m} \mathbb{E}_{\mathbf{c}_j^{(t-1)},\mathbf{c}_j^{(t)}}[-\log p(\mathbf{c}_j^{(t)} \,|\, \mathbf{c}_j^{(t-1)}, \mathbf{D}_m, \mathbf{d}_m, \mathbf{\Gamma}_m)] \\
&= \sum_{t,j:(t,j)\in\mathcal{M}^m} \Bigg(\frac{1}{2}\log|\mathbf{\Gamma}_m| \\
&+ \mathbb{E}_{\mathbf{c}_j^{(t-1)},\mathbf{c}_j^{(t)}}\left[\frac{1}{2}(\mathbf{c}_j^{(t)} - \mathbf{c}_j^{(t-1)} - \mathbf{D}_m\mathbf{c}_j^{(t-1)} - \mathbf{d}_m)^T\mathbf{\Gamma}_m^{-1}(\mathbf{c}_j^{(t)} - \mathbf{c}_j^{(t-1)} - \mathbf{D}_m\mathbf{c}_j^{(t-1)} - \mathbf{d}_m)\right]\Bigg)
\end{aligned}$$

subject to the assumptions (A4)–(A6). We start by estimating $\mathbf{D}_m$ and $\mathbf{d}_m$ given $\mathbf{\Gamma}_m$, and then use these estimates to estimate $\mathbf{\Gamma}_m$. In order to induce sparsity on $\mathbf{D}_m$ to take (A6) into account, we impose an ℓ_1-norm penalty on $\mathbf{D}_m$ (Hastie et al. (2010)). Taking only the terms containing $\mathbf{D}_m$ and $\mathbf{d}_m$, we can formulate the following augmented optimization problem:

$$\begin{aligned}
(\mathrm{P_d}) \qquad \underset{\mathbf{D}_m\in\mathcal{L}^+,\mathbf{d}_m}{\text{minimize}} \sum_{t,j:(t,j)\in\mathcal{M}^m} &\mathbb{E}_{\mathbf{c}_j^{(t-1)},\mathbf{c}_j^{(t)}}\Big[(\widetilde{\mathbf{D}}_m\widetilde{\mathbf{c}}_j^{(t-1)})^T\mathbf{\Gamma}_m^{-1}(\widetilde{\mathbf{D}}_m\widetilde{\mathbf{c}}_j^{(t-1)}) - \\
&(\mathbf{c}_j^{(t)} - \mathbf{c}_j^{(t-1)})^T\mathbf{\Gamma}_m^{-1}(\mathbf{c}_j^{(t)} - \mathbf{c}_j^{(t-1)})\Big] + \gamma\|\mathbf{D}_m\|_1\,,
\end{aligned}$$

where $\mathcal{L}^+$ denotes the set of lower-triangular matrices with non-negative entries. For notational simplicity, we have written $[\mathbf{D}_m\ \mathbf{d}_m]$ as $\widetilde{\mathbf{D}}_m$. We also write the augmented latent

state vectors $[(\mathbf{c}_j^{(t-1)})^T\ 1]^T$ as $\widetilde{\mathbf{c}}_j^{(t-1)}$, when multiplied by $\widetilde{\mathbf{D}}_m$, correspondingly. Note that the ℓ_1-norm penalty only applies to the matrix $\mathbf{D}_m$ in the used notation.

The problem ($\mathrm{P_d}$) is convex in $\widetilde{\mathbf{D}}_m$, and hence, can be solved efficiently. In particular, we use the fast iterative shrinkage and thresholding algorithm (FISTA) framework (Beck and Teboulle (2009)). The FISTA algorithm starts with a random initialization of $\widetilde{\mathbf{D}}_m$ and iteratively updates $\widetilde{\mathbf{D}}_m$ until a maximum number of iterations $\ell_{\max}$ is reached or the change in the estimate of $\widetilde{\mathbf{D}}_m$ between two consecutive iterations falls below a certain threshold. In each iteration $\ell = 1, 2, \ldots, \ell_{\max}$, the algorithm performs two steps. First, a gradient step that aims to lower the cost function performs

$$\widehat{\mathbf{D}}_m^{\ell+1} \leftarrow \widetilde{\mathbf{D}}_m^{\ell} - \eta_\ell \nabla f(\widetilde{\mathbf{D}}_m), \tag{11}$$

where $f(\widetilde{\mathbf{D}}_m)$ corresponds to the differentiable part of the cost function (excluding the ℓ_1-norm penalty) in ($\mathrm{P_d}$). The quantity η_ℓ is a step size parameter for iteration ℓ. For simplicity, we will take $\eta_\ell = 1/L$ in all iterations, where L is the Lipschitz constant given by

$$L = \sigma_{\max}\left(\sum_{t,j:(t,j)\in\mathcal{M}^m} \mathbb{E}_{\mathbf{c}_j^{(t-1)},\mathbf{c}_j^{(t)}}[(\mathbf{c}_j^{(t)} - \mathbf{c}_j^{(t-1)})(\mathbf{c}_j^{(t-1)})^T] \right) \sigma_{\max}(|\mathcal{M}^m|\,\mathbf{\Gamma}_m^{-1}).$$

Here $\sigma_{\max}(\cdot)$ denotes the maximum singular value of a matrix, and $|\mathcal{M}^m|$ denotes the cardinality of the set $\mathcal{M}^m$. The gradient $\nabla f(\widetilde{\mathbf{D}}_m)$ in (11) is given by

$$\begin{aligned}
\nabla f(\widetilde{\mathbf{D}}_m) &= -\mathbf{\Gamma}_m^{-1} \sum_{t,j:(t,j)\in\mathcal{M}^m} \left(\mathbb{E}_{\mathbf{c}_j^{(t-1)},\mathbf{c}_j^{(t)}}[(\mathbf{c}_j^{(t)} - \mathbf{c}_j^{(t-1)})(\widetilde{\mathbf{c}}_j^{(t-1)})^T] - \mathbf{D}_m^{\ell} \mathbb{E}_{\mathbf{c}_j^{(t-1)}}[\widetilde{\mathbf{c}}_j^{(t-1)}(\widetilde{\mathbf{c}}_j^{(t-1)})^T] \right) \\
&= -\mathbf{\Gamma}_m^{-1} \sum_{t,j:(t,j)\in\mathcal{M}^m} \Bigg([\mathbf{J}_j^{(t-1)} \widehat{\mathbf{V}}_j^{(t)} + \widehat{\mathbf{m}}_j^{(t)} (\widehat{\mathbf{m}}_j^{(t-1)})^T - \widehat{\mathbf{V}}_j^{(t-1)} - \widehat{\mathbf{m}}_j^{(t-1)} (\widehat{\mathbf{m}}_j^{(t-1)})^T \\
&\quad \widehat{\mathbf{m}}_j^{(t)} - \widehat{\mathbf{m}}_j^{(t-1)}] - \mathbf{D}_m^{\ell} \begin{bmatrix} \widehat{\mathbf{V}}_j^{(t-1)} + \widehat{\mathbf{m}}_j^{(t-1)} (\widehat{\mathbf{m}}_j^{(t-1)})^T & \widehat{\mathbf{m}}_j^{(t-1)} \\ (\widehat{\mathbf{m}}_j^{(t-1)})^T & 1 \end{bmatrix} \Bigg).
\end{aligned}$$

The parameters $\mathbf{J}_j^{(t-1)}$, $\widehat{\mathbf{m}}_j^{(t-1)}$, $\widehat{\mathbf{m}}_j^{(t)}$, $\widehat{\mathbf{V}}_j^{(t-1)}$, and $\widehat{\mathbf{V}}_j^{(t)}$ are obtained from the backward recursions in (7). Next, the FISTA algorithm performs a projection step, which takes into account the sparsifying regularizer $\gamma\|\mathbf{D}_m\|_1$, and the assumptions (A4) and (A5):

$$\widetilde{\mathbf{D}}_m^{\ell+1} \leftarrow P_{\mathcal{L}^+}(\max\{\widehat{\mathbf{D}}_m^{\ell+1} - \gamma\eta_\ell, 0\}), \tag{12}$$

where $P_{\mathcal{L}^+}(\cdot)$ correspond to the projection onto the set of lower-triangular matrices by setting all entries in the upper triangular part of $\mathbf{D}_m^{\ell+1}$ to zero. The maximum operator operates element-wise on $\mathbf{D}_m^{\ell+1}$. The updates (11) and (12) are repeated until convergence, eventually providing a new estimate $\widetilde{\mathbf{D}}_m^{\text{new}}$ for $[\mathbf{D}_m\ \mathbf{d}_m]$.

Using these new estimates, the update for $\mathbf{\Gamma}_m$ can be computed in closed form:

$$\begin{aligned}
\mathbf{\Gamma}_m^{\text{new}} &= \frac{1}{|\mathcal{M}^m|} \sum_{t,j:(t,j)\in\mathcal{M}^m} \big(\mathbb{E}_{\mathbf{c}_j^{(t)}}[\mathbf{c}_j^{(t)}(\mathbf{c}_j^{(t)})^T] - \widetilde{\mathbf{D}}_m^{\text{new}}\mathbb{E}_{\mathbf{c}_j^{(t-1)},\mathbf{c}_j^{(t)}}[\widetilde{\mathbf{c}}_j^{(t-1)}(\mathbf{c}_j^{(t)})^T] \\
&\quad - \mathbb{E}_{\mathbf{c}_j^{(t-1)},\mathbf{c}_j^{(t)}}[\mathbf{c}_j^{(t)}(\widetilde{\mathbf{c}}_j^{(t-1)})^T](\widetilde{\mathbf{D}}_m^{\text{new}})^T + (\widetilde{\mathbf{D}}_m^{\text{new}})\mathbb{E}_{\mathbf{c}_j^{(t-1)}}[\widetilde{\mathbf{c}}_j^{(t-1)}(\widetilde{\mathbf{c}}_j^{(t-1)})^T](\widetilde{\mathbf{D}}_m^{\text{new}})^T\big) \\
&= \frac{1}{|\mathcal{M}^m|} \sum_{t,j:(t,j)\in\mathcal{M}^m} \Big(\widehat{\mathbf{V}}_j^{(t)} + \widehat{\mathbf{m}}_j^{(t)}(\widehat{\mathbf{m}}_j^{(t)})^T - \widetilde{\mathbf{D}}_m^{\text{new}} \begin{bmatrix} \mathbf{J}_{t-1,j}\widehat{\mathbf{V}}_j^{(t)} + \widehat{\mathbf{m}}_j^{(t)}(\widehat{\mathbf{m}}_j^{(t-1)})^T \\ (\widehat{\mathbf{m}}_j^{(t)})^T \end{bmatrix} \\
&\quad - \big[\mathbf{J}_j^{(t-1)}\widehat{\mathbf{V}}_j^{(t)} + \widehat{\mathbf{m}}_j^{(t)}(\widehat{\mathbf{m}}_j^{(t-1)})^T \;\; \widehat{\mathbf{m}}_j^{(t)}\big](\widetilde{\mathbf{D}}_m^{\text{new}})^T \\
&\quad + \widetilde{\mathbf{D}}_m^{\text{new}} \begin{bmatrix} \widehat{\mathbf{V}}_j^{(t-1)} + \widehat{\mathbf{m}}_j^{(t-1)}(\widehat{\mathbf{m}}_j^{(t-1)})^T & \widehat{\mathbf{m}}_j^{(t-1)} \\ (\widehat{\mathbf{m}}_j^{(t-1)})^T & 1 \end{bmatrix} (\widetilde{\mathbf{D}}_m^{\text{new}})^T\Big).
\end{aligned}$$

4.4 Estimating the question-dependent parameters

We next show how to estimate the question-dependent parameters $\bar{\mathbf{w}}_i$, μ_i, $\forall i$. To this end, we define $\mathcal{Q}^i$ as the collection set of time and learner indices (t, j) that learner j answered the i^{th} question at time instance t. We then minimize the expected negative log-likelihood of all the observed binary-valued graded learner responses (1) for the i^{th} question, subject to assumptions (A2) and (A3) on the question–concept association vector $\bar{\mathbf{w}}_i$. In order to impose sparsity on $\bar{\mathbf{w}}_i$, we add an ℓ_1-norm penalty to the cost function, which leads to the following optimization problem:

$$(\mathrm{P_w}) \qquad \underset{\bar{\mathbf{w}}_i : w_{i,k}\geq 0, \forall k}{\text{minimize}} \sum_{(t,j)\in\mathcal{Q}^i} \mathbb{E}_{\mathbf{c}_j^{(t)}}\Big[-\log\Phi((2Y_j^{(t)}-1)(\bar{\mathbf{w}}_i^T\mathbf{c}_j^{(t)} - \mu_i))\Big] + \lambda\|\bar{\mathbf{w}}_i\|_1 .$$

This problem corresponds to the (RR_1^+) problem of SPARFA detailed in Lan et al. (2012), where point estimates of $\mathbf{c}_j$ are given and the problem is convex in $\bar{\mathbf{w}}_i$. In particular, given the distribution $\mathbf{c}_j^{(t)} \sim \mathcal{N}(\mathbf{c}_j^{(t)} \mid \widehat{\mathbf{m}}_j^{(t)}, \widehat{\mathbf{V}}_j^{(t)})$, $(\mathrm{P_w})$ is still convex in $\bar{\mathbf{w}}_i$, thanks to the linearity of the expectation operator. However, the inverse probit link function prohibits us from obtaining a simple form of this expectation. In order to develop a tractable algorithm to approximately solve this problem, we utilize the unscented transform (UT) (Wan and Van Der Merwe (2000)) to approximate the cost function of $(\mathrm{P_w})$.

The UT is commonly used in Kalman filtering literature to approximate the statistics of a random variable undergoing non-linear transformations. Specifically, given a K-dimensional random variable $\mathbf{x}$ with known mean and covariance and a non-linear function $g(\cdot)$, the UT generates a set of $2K+1$ so-called *sigma vectors* $\{\mathcal{X}_n\}$ and a set of corresponding weights $\{u_n\}$ as detailed in (Wan and Van Der Merwe, 2000, Eq.15), in order to approximate the mean and covariance of the vector $\mathbf{y} = g(\mathbf{x})$. As shown in Wan and Van Der Merwe (2000), this approximation is accurate up to the third order for Gaussian distributed random variables $\mathbf{x}$.

Following the paradigms of the UT, we generate a set of sigma vectors $\{(\widetilde{\mathbf{c}}_j^{(t)})_n\}$ and a corresponding set of weights $\{u_n\}$, $n \in \{1, \ldots, 2K+1\}$, for each latent state vector $\mathbf{c}_j^{(t)}$, given the mean $\widehat{\mathbf{m}}_j^{(t)}$ and covariance $\widehat{\mathbf{V}}_j^{(t)}$. For computational simplicity, we will use the same

set of weights for all latent state vectors $\mathbf{c}_j^{(t)}$. The optimization problem ($\mathrm{P_w}$) can now be approximated by

$$\underset{\bar{\mathbf{w}}_i : w_{i,k} \geq 0, \forall k}{\text{minimize}} \sum_{(t,j)\in\mathcal{Q}^i} \sum_{n=1}^{2K+1} u_n \left(-\log \Phi((2Y_j^{(t)} - 1)(\bar{\mathbf{w}}_i^T (\widetilde{\mathbf{c}}_j^{(t)})_n - \mu_i)) \right) + \lambda \|\bar{\mathbf{w}}_i\|_1 ,$$

which, once again, can be solved efficiently by using the FISTA framework. The resulting iterative procedure performs two steps in each iteration ℓ: First, a gradient step that aims at lowing the cost function performs

$$\widehat{\bar{\mathbf{w}}}_i^{\ell+1} \leftarrow \bar{\mathbf{w}}_i^{\ell} - \eta_\ell \, \nabla f(\bar{\mathbf{w}}_i), \tag{13}$$

where $f(\bar{\mathbf{w}}_i)$ corresponds to the differentiable portion (excluding the ℓ_1-norm penalty part) of the cost function in ($\mathrm{P_w}$). The gradient $\nabla f(\bar{\mathbf{w}}_i)$ is given by $\nabla f(\bar{\mathbf{w}}_i) = -\widetilde{\mathbf{C}}_i \widetilde{\mathbf{r}}_i$, where $\widetilde{\mathbf{r}}_i$ is a $(2K+1)|\mathcal{Q}^i| \times 1$ vector $\mathbf{r}_i = [\mathbf{a}_i^1 \ldots, \mathbf{a}_i^{|\mathcal{Q}^i|}]^T$. The vector $\mathbf{a}_i^q$ is defined by $\mathbf{a}_i^q = [(g_i^q)_1, \ldots, (g_i^q)_{2K+1}]$, where

$$(g_i^q)_n = u_n 2(Y_{j_q}^{(t_q)} - 1) \frac{\mathcal{N}\left(2(Y_{j_q}^{(t_q)} - 1)\bar{\mathbf{w}}_i^T (\widetilde{\mathbf{c}}_{j_q}^{(t_q)})_n\right)}{\Phi\left(2(Y_{j_q}^{(t_q)} - 1)\bar{\mathbf{w}}_i^T (\widetilde{\mathbf{c}}_{j_q}^{(t_q)})_n\right)},$$

in which (t_q, j_q) represents the q^{th} time–learner index pair in $\mathcal{Q}^i$. The $K \times (2K+1)|\mathcal{Q}^i|$ matrix $\widetilde{\mathbf{C}}_i$ is defined as $\widetilde{\mathbf{C}}_i = \left[(\mathbf{G}_i)_1, \ldots, (\mathbf{G}_i)_{|\mathcal{Q}^i|}\right]$, where the $K \times (2K+1)$ matrix $(\mathbf{G}_i)_q$ is given by

$$(\mathbf{G}_i)_q = \left[(\widetilde{\mathbf{c}}_{j_q}^{(t_q)})_1, \ \ldots, \ (\widetilde{\mathbf{c}}_{j_q}^{(t_q)})_{2K+1} \right].$$

The quantity η_ℓ is a step size parameter for iteration ℓ. For simplicity, we will take $\eta_\ell = 1/L$ in all iterations, where L is the Lipschitz constant given by $L = \sigma_{\max}(\widetilde{\mathbf{C}}_i)\, \sigma_{\max}(\widetilde{\mathbf{C}}_i')$, where $\widetilde{\mathbf{C}}_i'$ is a $K \times (2K+1)|\mathcal{Q}^i|$ matrix defined as $\widetilde{\mathbf{C}}_i' = \left[(\mathbf{G}_i')_1, \ldots, (\mathbf{G}_i')_{|\mathcal{Q}^i|}\right]$, where the $K \times (2K+1)$ matrix $(\mathbf{G}_i')_q$ is given by

$$(\mathbf{G}_i')_q = \left[u_1(\widetilde{\mathbf{c}}_{j_q}^{(t_q)})_1, \ \ldots, \ u_{2K+1}(\widetilde{\mathbf{c}}_{j_q}^{(t_q)})_{2K+1} \right].$$

Next, the FISTA algorithm performs a projection step, which takes into account $\lambda \|\bar{\mathbf{w}}_i\|_1$ and the assumption (A3):

$$\bar{\mathbf{w}}_i^{\ell+1} \leftarrow \max\{\widehat{\bar{\mathbf{w}}}_i^{\ell+1} - \lambda \eta_\ell, 0\}. \tag{14}$$

The steps (13) and (14) are repeated until convergence, providing a new estimate $\bar{\mathbf{w}}_i^{\text{new}}$ of the question–concept association vector $\bar{\mathbf{w}}_i$. For simplicity of exposition, the question intrinsic difficulties μ_i are omitted in the derivations above, as they can be included as an additional entry in $\bar{\mathbf{w}}_i$ as $[\bar{\mathbf{w}}_i^T \ \mu_i]^T$; the corresponding latent learner concept knowledge state vectors $\mathbf{c}_j^{(t)}$ are augmented as $[(\mathbf{c}_j^{(t)})^T \ 1]^T$.

5. Experimental Results

We now demonstrate the efficacy of SPARFA-Trace on synthetic and real-world educational datasets. We start by performing experiments using synthetic data to demonstrate that SPARFA-Trace is able to accurately trace latent learner concept knowledge and accurately estimate learner concept knowledge state transition parameters and question-dependent parameters. We then compare SPARFA-Trace against two established methods on predicting unobserved binary-valued learner response data, namely knowledge tracing (KT) (Corbett and Anderson (1994); Pardos and Heffernan (2010a)) and SPARFA (Lan et al. (2012)). Finally, we show how SPARFA-Trace is able to visualize learners' concept knowledge state evolution over time, and the learning resource and question quality and their content organization. For all the synthetic and real data experiments shown next, the regularization parameters λ, γ and σ_0^2 are chosen via cross-validation (Hastie et al. (2010)), and all experiments are repeated for 25 independent Monte–Carlo trials for each instance of the model parameter we control.

5.1 Experiments with synthetic data

In the following experiments with synthetic data, we assess the performance of SPARFA-Trace in both (i) learner concept knowledge tracing, and (ii) estimating all learner concept knowledge state transition parameters and question-dependent parameters.

Dataset: We generate the learning resource-induced learner knowledge state transition parameters $\mathbf{D}_m$, $\mathbf{d}_m$, $\mathbf{\Gamma}_m$, $m \in \{1, \ldots, M\}$, $\bar{\mathbf{w}}_i, \mu_i, i \in \{1, \ldots, Q\}$, under the assumptions (A1)–(A6), and randomly generate learner prior parameters $\mathbf{m}_j^{(0)}$ and $\mathbf{V}_j^{(0)}, j \in \{1, \ldots, N\}$. Using these parameters, we randomly generate latent learner concept knowledge states $\mathbf{c}_j^{(t)}$ and observed binary-valued graded responses $Y_j^{(t)}$, $t \in \{1, \ldots, T\}$, according to (1) and (2). The number of time instances is $T = 100$, and one question is assigned to every learner at every time instance, so $Q = T = 100$. The dataset consist of 10 assignment sets, each consisting of 10 questions. The learners' concept knowledge states evolve between consecutive assignment sets, induced by their interaction with learning resources. Therefore, the number of learning resources is $M = 9$. There are a total of $K = 5$ concepts, as this choice is shown to be reasonable for real-world educational scenarios (see, e.g., Fronczyk et al. (2013, submitted) for a corresponding discussion).

Learner concept knowledge tracing: For the learner concept knowledge state estimation experiment, we fix the number of learners as $N = 50$ and vary the percentage of observed entries in the $Q \times N$ learner response matrix $\mathbf{Y}$ as $\{100\%, 75\%, 50\%, 25\%\}$ and calculate the normalized concept knowledge state estimation error

$$E_{\mathbf{c}} = \frac{1}{NT} \sum_{(t,j)} \frac{\|\mathbf{m}_j^{(t)} - \mathbf{c}_j^{(t)}\|_2^2}{\|\mathbf{c}_j^{(t)}\|_2^2}. \tag{15}$$

In this experiment, all learner-dependent and learner concept knowledge state transition and question parameters are assumed to be known. Thus, we only run the Kalman filtering and smoothing part of SPARFA-Trace.

Figure 3(a) shows the results from the learner concept knowledge state estimation experiment. We observe that the estimation of learner concept knowledge states becomes increasingly accurate as time proceeds. The performance of SPARFA-Trace decreases as the percentage of missing observations increases. Moreover, SPARFA-Trace can still obtain accurate estimates of $\mathbf{c}_j^{(t)}$ even when only a small portion of the response data is observed.

Estimating learner concept knowledge state transition and question parameters: To assess SPARFA-Trace on the estimation performance of learner concept knowledge state transition and question parameters, we perform a second experiment, which focus on the estimation of all learning resource and question-dependent parameters: $\mathbf{D}_m, \mathbf{d}_m, \mathbf{\Gamma}_m$, $\forall m$, $\bar{\mathbf{w}}_i, \mu_i$, $\forall i$. The learner concept knowledge states $\mathbf{c}_j^{(t)}$ are not given and are estimated simultaneously, while we treat the learner prior parameters $\mathbf{m}_j^{(0)}$ and $\mathbf{V}_j^{(0)}, j \in \{1, \ldots, N\}$ as given, to avoid the scaling unidentifiability issue in the model (one can arbitrarily scale the learner concept knowledge state vectors $\mathbf{c}_j^{(t)}$ and adjust the scale of the question–concept association vectors $\bar{\mathbf{w}}_i$ accordingly, and still arrive at the same likelihood for the observations. See, e.g., Lan et al. (2012) for a detailed discussion.) We fix the number of concepts as $K = 5$, vary the number of learners as $N \in \{50, 100, 200\}$, and examine the estimation error of SPARFA-Trace on all instructional and question-dependent parameters using a similar metric as in (15). The observed learner response matrix $\mathbf{Y}$ is assumed to be fully observed. We run SPARFA-Trace until convergence, to provide estimates of all unknown parameters.

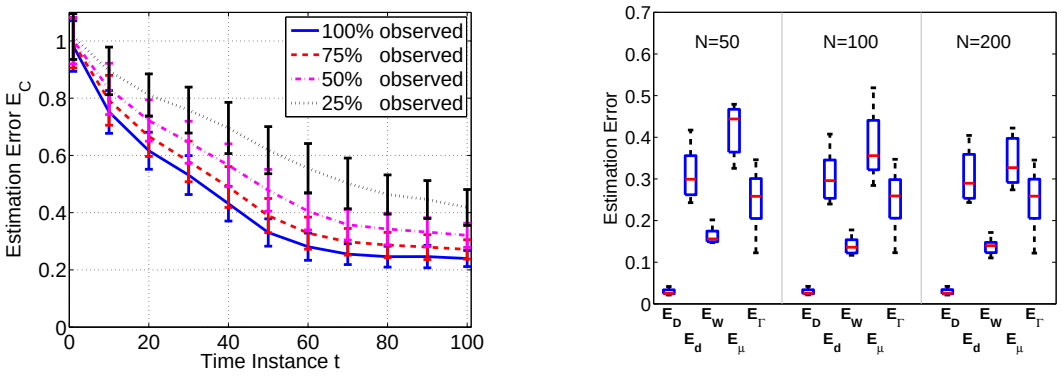
Figure 3(b) shows the box-and-whisker plots of the estimation error on all five types of parameters for different numbers of learner N. We can see that the parameter estimation performance of SPARFA-Trace improves as the number of learners increase. More importantly, SPARFA-trace provides accurate estimates of these parameters even when the problem size is relatively small (e.g., the number of learners $N = 50$).

In summary of these synthetic experiments, we can conclude that SPARFA-Trace is capable of accurately estimating both latent learner concept knowledge states and the learner concept knowledge state transition and question parameters.

5.2 Predicting responses for new learners

We now compare SPARFA-Trace against the KT method described in Pardos and Heffernan (2010a) on predicting responses for new learners that do not have previous recorded response history.

Dataset: The dataset we use for this experiment is from an undergraduate computer engineering course collected from OpenStax Tutor (OST) (OpenStaxTutor (2013)). We will refer to this dataset as "Dataset 1" in the following experiments. This dataset consist of the binary-valued graded response from 92 learners answering 203 questions, with 99.5% of the responses observed. Since the KT implementation of Pardos and Heffernan (2010a) is unable to handle missing data, we removed learners that do not answer every question from the dataset, resulting in a pruned dataset of 73 learners. The course is organized into three independent sections: The first section is on digital logic, the second on data structures, and the third on basic programming concepts. The full course consist of 11 assessments, including 8 homework assignments and an exam at the end of each section; we assume that



(a) (b)

Figure 3: Accuracy of latent concept knowledge state and learning resource and question-dependent parameters estimation for synthetic data. (a) Learner concept knowledge state estimation error versus time instance t for different percentages of observed responses; (b) Learning resource parameter estimation error for various number of learners N. Note the general trend that all considered performance measures improve as the amount of observed data increases.

the learners' concept knowledge state transitions can only happen between two consecutive assignments/exams, due to their interaction with all the lectures/readings/exercises.

Experimental setup: Since KT is only capable of handling educational datasets that involve a single concept, we partition Dataset 1 into three parts, with each part corresponding to one of the three independent sections. We run KT independently on the three parts, and aggregate the prediction results. (We also ran KT on the entire Dataset 1 without separating it into 3 independent sections. The obtained results are inferior to those obtained by running KT on 3 independent sections.) The four parameters of KT (learner prior, learning probability, guessing probability, slipping probability) are initialized with the best initial value we find over 5 different initializations. For SPARFA-Trace, we use $K = 3$, with each concept corresponding to one section of the dataset. In order to alleviate the identifiability issue in our model, we initialize the algorithm with $\bar{\mathbf{w}}_{i,k} = 1$ where question i is in section k and $\bar{\mathbf{w}}_{i,k} = 0$ otherwise. We also initialize the matrices $\mathbf{D}_m$ with identity matrices $\mathbf{I}_{3\times 3}$, the vectors $\mathbf{d}_m$ with zero vectors, and covariance matrices $\mathbf{\Gamma}_m$ with identity matrices.

For cross-validation, we randomly partition Dataset 1 into 5 folds, with each fold consisting of 1/5 of the learners answering all questions. Four folds of the data are used as the training set and the other fold is used as the test set. We train both KT and SPARFA-Trace on the training set and obtain estimates on all learner, learning resource and question-dependent parameters, and test their prediction performances on the test set. For previously unobserved new learners in the test set, both algorithms make the first prediction of $Y_j^{(1)}$ at $t = 1$ using question-dependent parameters estimated from the training set. As time goes

Table 1: Comparisons of SPARFA-Trace against knowledge tracing (KT) on predicting responses for new learners using using Dataset 1. SPARFA-Trace outperforms KT on all considered metrics.

	KT	**SPARFA-Trace**
Prediction accuracy	$86.42 \pm 0.16\%$	$\mathbf{87.49 \pm 0.12}\%$
Prediction likelihood	0.7718 ± 0.0011	$\mathbf{0.8128 \pm 0.0044}$
Area under the ROC curve	0.5989 ± 0.0056	$\mathbf{0.8157 \pm 0.0028}$

on, more and more observed responses $Y_j^{(t)}$ are available to both algorithms, and they use these responses to make future predictions.

We compare both algorithms on three metrics: prediction accuracy, prediction likelihood, and area under the receiver operation characteristic (ROC) curve. The prediction accuracy corresponds to the percentage of correctly predicted responses; the prediction likelihood corresponds to the average the predicted likelihood of the unobserved responses, i.e., $\frac{1}{|\Omega^c_{\text{obs}}|} \sum_{t,j:(t,j)\in\Omega^c_{\text{obs}}} p(Y_j^{(t)}|\bar{\mathbf{w}}_{i_j^{(t)}}, \mathbf{c}_j^{(t)})$ where Ω^c_{obs} is the set of learner responses in the test set; the area under the ROC curve is a commonly-used performance metric for binary classifiers (see Pardos and Heffernan (2010b) for details). The area under the ROC curve always is always between 0 and 1, with a larger value representing higher classification accuracy.

Since SPARFA-Trace does not provide point estimates of $\mathbf{c}_j^{(t)}$ but rather their distributions, we compute the predicted likelihood of unobserved responses by:

$$\mathbb{E}_{\mathbf{c}_j^{(t)}}\left[p(Y_j^{(t)}|\bar{\mathbf{w}}_{i_j^{(t)}}, \mathbf{c}_j^{(t)})\right] = \Phi\left(\left(2Y_j^{(t)}-1\right)\frac{\bar{\mathbf{w}}_{i_j^{(t)}}^T \widehat{\mathbf{m}}_j^{(t)} - \mu_{i_j^{(t)}}}{\sqrt{1+\bar{\mathbf{w}}_{i_j^{(t)}}^T \widehat{\mathbf{V}}_j^{(t)} \bar{\mathbf{w}}_{i_j^{(t)}}}}\right).$$

Results: The means and standard deviations of all three metrics covering multiple cross-validation trials are shown in Table 1. We can see that SPARFA-Trace outperforms KT on all performance metrics for Dataset 1. We also emphasize that SPARFA-Trace is capable of achieving superior prediction performance while simultaneously estimating the quality and content organization parameters of all learning resources and questions.

5.3 Predicting unobserved learner responses

It has been shown (Gong et al. (2010); Pardos and Heffernan (2010b)) that collaborative filtering methods often outperform KT in predicting unobserved learner responses, even though they ignore any temporal evolution aspects of the dataset. Hence, we compare SPARFA-Trace against the original SPARFA framework (Lan et al. (2012)), which shows state-of-the-art collaborative filtering performance on predicting unobserved learner responses.

Datasets: We will use two datasets in this experiment. The first dataset is the full Dataset 1 with 92 learners answering 203 questions, explained in Section 5.2. The second dataset we use is from a signals and systems undergraduate course on OST, consisting

Table 2: Comparisons of SPARFA-Trace against SPARFA-M on predicting unobserved learner responses for Dataset 1.

	SPARFA-M	**SPARFA-Trace**
Prediction accuracy	$87.10 \pm 0.04\%$	$\mathbf{87.31 \pm 0.05}\%$
Prediction likelihood	0.7274 ± 0.0005	$\mathbf{0.7295 \pm 0.0007}$

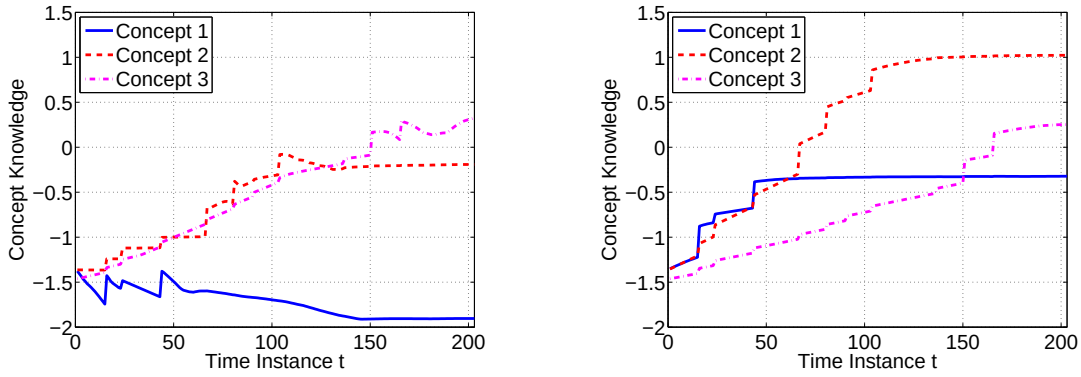
Table 3: Comparisons of SPARFA-Trace against SPARFA-M on predicting unobserved learner responses for Dataset 2.

	SPARFA-M	**SPARFA-Trace**
Prediction accuracy	$86.64 \pm 0.14\%$	$\mathbf{86.29 \pm 0.25}\%$
Prediction likelihood	0.7037 ± 0.0024	$\mathbf{0.7066 \pm 0.0028}$

of 41 learners answering 143 questions, with 97.1% of the responses observed. We will refer to this dataset as "Dataset 2" in the following experiments. All the questions were manually labeled with a number of $K = 4$ concepts, with the concepts being listed in Figure 6(b). The full course consist of 14 assessments, including 12 assignments and 2 exams; we will treat all the lectures/readings/exercises the learners interact with between two consecutive assignments/exams as an learning resource.

Experimental setup: We randomly partition the 143×43 (or 203×92) matrix $\mathbf{Y}$ of observed graded learner responses into 5 folds for cross-validation. Four folds of the data are used as the training set and the other fold is used as the test set. We train both the probit variant of SPARFA-M and SPARFA-Trace on the training set to estimate the learner concept knowledge states and the learner, learning resource and question-dependent parameters, and then use these estimates to predict unobserved held-out responses in the test set.

Results: The means and standard deviations of the prediction accuracy and prediction likelihood metrics covering multiple cross-validation trials are shown in Tables 2 and 3. We see that SPARFA-Trace achieves comparable prediction performance to SPARFA-M on both datasets, although the datasets are treated as if they do not have time-varying effects. We emphasize that, in addition to providing competitive prediction performance, SPARFA-Trace is capable of (i) tracing learner concept knowledge evolution over time and (ii) analyzing learning resource and question qualities and their content organization. This extracted information is very important, as it allow a PLS to provide timely feedback to learners about their strengths and weaknesses, and to automatically recommend learning resources to learners for remedial studies based on their qualities and contents.



(a) (b)

Figure 4: Estimated latent learner concept knowledge states for all time instances, for Dataset 1. (a) Learner 1's latent concept knowledge state evolution; (b) Average learner latent concept knowledge states evolution.

5.4 Visualizing time-varying learning and content analytics

In this section, we showcase another advantage of SPARFA-Trace over existing KT and collaborative filtering methods, i.e., the visualization of both learner knowledge state evolution over time and the estimated learning resource and question quality and content organization.

Visualizing learner concept knowledge state evolution: Figure 4(a) shows the estimated latent learner concept knowledge states at all time instances for Learner 1 in Dataset 1. We can see that their knowledge on Concepts 2 and 3 gradually improve over time, while their knowledge on Concept 1 does not. Therefore, recommending Learner 1 remedial material on Concept 1 seems necessary, which is verified by the fact that Learner 1 often responds incorrectly on questions covering Concept 1 towards the end of the course.

Figure 4(b) shows the average learner concept knowledge states over the entire class at all time instances for Dataset 1. Since Concept 1 is the basic concept that is covered in the early stages of the course, we can see that its mean knowledge among all learners increases in early stages of the course and then remain constant afterwards. In contrast, Concept 3 is the most advanced concept covered near the end of the course, and the improvement in which is not obvious until very late stages of the course. Hence, SPARFA-Trace enables a PLS to provide timely feedback to individual learners on the their concept knowledge at all times, which reveals the learning progress of the learners. SPARFA-Trace also informs instructors on the trend of concept knowledge state evolution of the entire class, in order to help them make timely adjustments to their course plans.

Visualizing learning resource quality and content: Figure 5(a) and Figure 5(b) show the quality and content organization of learning resources 3 and 9 for Dataset 2. These figures visualize the leaners' concept knowledge state transitions induced by interacting with learning resources 3 and 9. Circular nodes represent concepts; the leftmost set of dashed nodes represent the concept knowledge state vector $\mathbf{c}^{(t-1)}$, which are the learners'

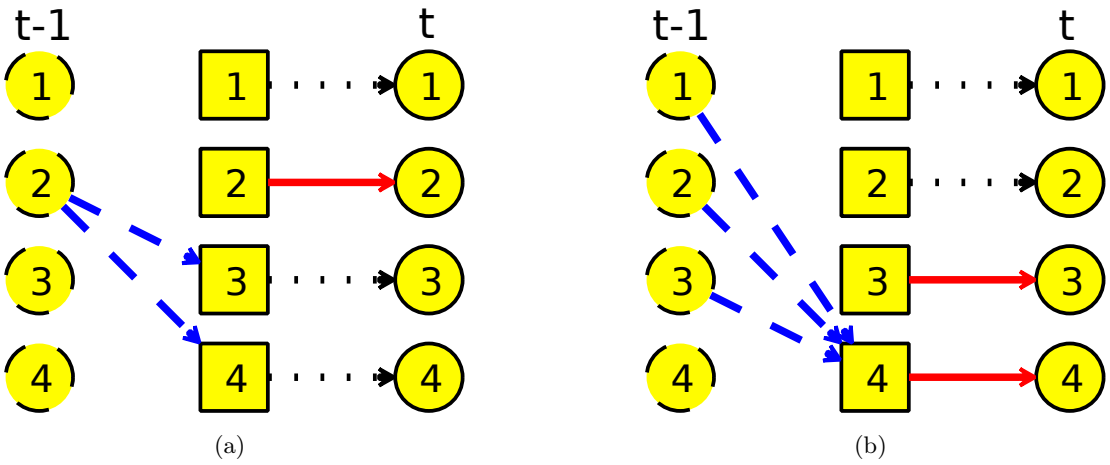
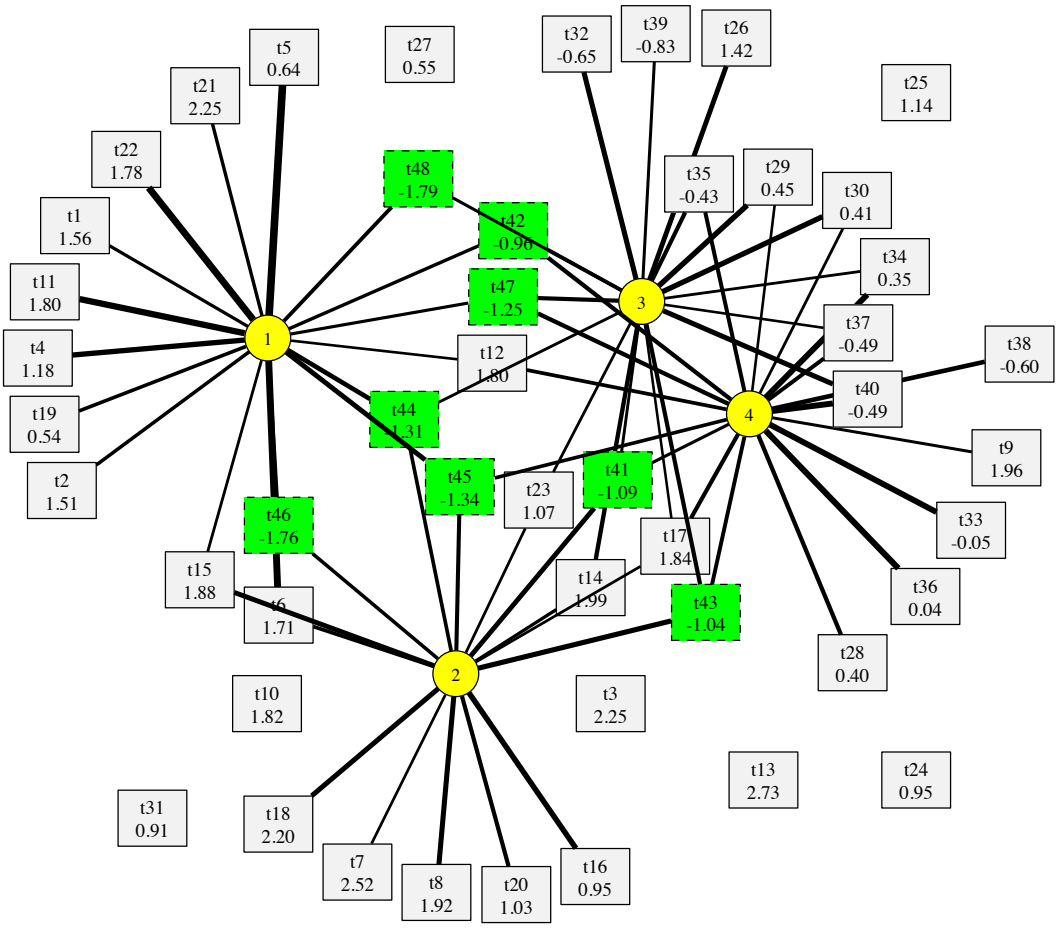


Figure 5: Visualized learner knowledge state transition effect of two distinct learning resources for Dataset 2. (a) Learner knowledge state transition effect for Learning resource 3; (b) Learner knowledge state transition effect for Learning resource 9.

concept knowledge states before interacting with these learning resources, and the rightmost set of solid nodes represent the concept knowledge state vector $\mathbf{c}^{(t)}$, which are the learners' concept knowledge states after interacting with these learning resources. Arrows represent the the learner concept knowledge state transition matrix $\mathbf{D}_m$, the intrinsic quality vector of the learning resource $\mathbf{d}_m$, and their transformation effects on learners' concept knowledge states. Black, dotted arrows represent unchanged learner concept knowledge states; these arrows correspond to zero entries in $\mathbf{D}_m$ and $\mathbf{d}_m$. Red, solid arrows represent the intrinsic knowledge gain of some concepts, characterized by large, positive entries in $\mathbf{d}_m$. Blue, dashed arrows represent the change in knowledge of advanced concepts due to their pre-requisite concepts, characterized by non-zero entries in $\mathbf{D}_m$: High knowledge level on pre-requisite concepts can result in improved understanding and an increase on knowledge of advanced concepts, while low knowledge level on these pre-requisite concepts can result in confusion and a decrease on knowledge of advanced concepts.

As shown in Figure 5(a), Learning resource 3 is used in early stage of the course, and we can see that this learning resource gives the learners' a positive knowledge gain of Concept 2, while also helping on more advanced Concepts 3 and 4. As shown in Figure 5(b), Learning resource 9 is used in later stage of the course, and we can see that it uses the learners' knowledge on all previous concepts to improve their knowledge on Concept 4, while also providing a positive knowledge gain on Concepts 3 and 4.

By analyzing the content organization of learning resources and their effects on learner concept knowledge state transitions, SPARFA-Trace enables a PLS to automatically recommend corresponding learning resources to learners based on their strengths and weaknesses. The estimated learning resource quality information also helps course instructors to distinguish between effective learning resources, and poorly-designed, off-topic, or misleading learning resources, thus helping them to manage these learning resources more easily.



(a)

Concept 1	**Concept 2**
Laplace transform and filters	Sampling and reconstruction
Concept 3	**Concept 4**
Fourier series and Fourier transform	Signals and systems basics

(b)

Figure 6: (a) Question–concept association graph and concept labels for Dataset 2. (a) Question–concept association graph. Note that for the visualization to be compact, we show only 1/3 of all questions in the dataset; (b) Label of each concept.

Visualizing question quality and content: Figure 6 shows the question–concept association graph obtained from Dataset 2. Yellow, circle nodes represent concept nodes, while green, box nodes represent question nodes. Each question box is labeled with the time instance at which it is assigned and its estimated intrinsic difficulty. From the graph we can see time-evolving effects, as questions assigned in the early stages of the course cover basic concepts (Concepts 1 and 2), while questions assigned in later stages cover more advanced concepts (Concepts 3 and 4). Some questions are associated with multiple concepts, and they mostly correspond to the final exam questions (boxes with dashed boundaries) where the entire course is covered.

Thus, by estimating the intrinsic difficulty and content organization of each question, SPARFA-Trace allows a PLS to generate feedback to instructors on the underlying knowledge structure of questions, which enables them to identify ill-posed or off-topic questions (such as questions that are not associated to any concepts in Figure 6(a)).

6. Related Work on Knowledge Tracing for Personalized Learning

Various machine learning algorithms have been designed for personalized learning. Specifically, matrix and tensor factorization approaches have been applied to analyze graded learner responses in order to extract learner ability parameters and/or question–concept relationships. Examples include item response theory (IRT) (Lord (1980); Rasch (1993); Hooker et al. (2009); Jordan and Spiess (2012)), and other factor analysis models (Barnes (2005); Cen et al. (2006); Lan et al. (2012)). While these methods have shown to provide good prediction performance on unobserved learner responses, they do not take into account the temporal dynamics involved in the process of a course. Therefore, these approaches are only suitable to a static testing scenario, such as the graduate record examinations (GRE), standardized tests, placement exams, etc (see van der Linden (1998) for details).

A number of approaches have also been developed to analyze temporal learner response data (see, e.g., Corbett and Anderson (1994); Pardos and Heffernan (2010b) for details). In particular, knowledge tracing (KT) estimates learner concept knowledge over time, given question–concept mappings and graded binary learner response data. Since such methods all require pre-defined question–concept mappings which are, in general, not available in practice, these methods are labor-intensive to instructors and domain experts, and are not scalable to large-scale applications such as massive online open courses (MOOCs) (see Martin (2012); Knox et al. (2012) for an overview).

Recent approaches to KT without requiring question–concept mappings, described in González-Brenes and Mostow (2012) and González-Brenes and Mostow (2013) jointly estimate both question–concept (item–skill) mappings and learner concept mastery evolution over time purely from response data. Their method, however, suffers from the following deficiencies: First, González-Brenes and Mostow (2012) models the learners' latent concept knowledge as a small number of *discrete* values and the entire dynamic process for learning is modeled as a hidden Markov model (HMM). Such discrete concept knowledge states do not provide desirable interpretability when the number of discrete learner concept knowledge values is low (the authors used 3 distinct knowledge levels in their paper). In contrary, the proposed SPARFA-Trace framework models learner latent concept knowledge states as continuous random variables, providing finer knowledge repre-

sentations. Second, González-Brenes and Mostow (2012) does not handle questions that involve multiple concepts. In contrary, the proposed SPARFA-Trace framework directly takes into account questions involving multiple concepts in the probabilistic model. Third, González-Brenes and Mostow (2012) introduced a Gibbs sampler approach to infer all parameters; such an approach is known to be computationally intensive and, hence, not scalable to large datasets, such as data obtained in a MOOC. In contrary, the proposed SPARFA-Trace framework uses a computationally efficient EM approach, which is capable of scaling to personalized learning applications at MOOC scale.

7. Conclusions

We have proposed SPARFA-Trace, a novel, message passing-based approximate Kalman filtering approach for time-varying learning and content analytics. The proposed method jointly traces latent learner concept knowledge and simultaneously estimates the quality and content organization of the corresponding learning resources (such as textbook sections or lecture videos), and the questions in assessment sets. In order to estimate latent learner concept knowledge states at each time instance from observed binary-valued graded learner responses, we have introduced an approximate Kalman filtering framework, given all learner concept knowledge state transition parameters of learning resources and the question-dependent parameters. In order to estimate these parameters, we have introduced novel convex optimization-based algorithms that estimate all the learner concept knowledge state transition parameters of learning resources and question–concept associations and their intrinsic difficulties. The proposed approach applied to real-world educational datasets has shown its capability of accurately predicting unobserved learner responses, while obtaining interpretable estimates of all learner concept knowledge state transition parameters and question–concept associations.

A PLS can benefit from the information extracted by the SPARFA-Trace framework in a number of ways. Being able to trace learners' concept knowledge enables a PLS to make timely feedback to learners on their strengths and weaknesses. Meanwhile, this information will also enable adaptivity in designing personalized learning pathways in real time, as instructors can recommend different actions for different learners to take, based on their individual concept knowledge states. Furthermore, the estimated content-dependent parameters provide rich information on the knowledge structure and quality of learning resources. This capacity is crucial for a PLS to automatically suggest learning resources to learners for remedial studies. Together with the question parameters estimated, a PLS would be able to operate in a *hands-off* manner, requiring only minimal human input and intervention; this paves the way of applying SPARFA-Trace to MOOC-scale education scenarios, where the massive amount of data prevents any manual intervention.

We finally note that a number of improvements/extensions to SPARFA-Trace could be made. For example, more accurate message-passing schemes like expectation propagation (Qi (2004)) could be applied to improve the performance and accuracy of SPARFA-Trace. More sophisticated non-affine learner concept knowledge state transition models can also be applied, in contrast to the affine model proposed in Section 2.2. In order to provide better interpretation to the estimated learner concept knowledge state transition and question parameters, tagging and question text information can be coupled with SPARFA-Trace (see

Lan et al. (2013a,b) for corresponding extensions to SPARFA that mine question tags and question text information). It is worth mentioning that SPARFA-Trace has potential to be applied to a wide range of other datasets, including (but not necessarily limited to) the analysis of temporal evolution in legislative voting data (Wang et al. (2013)), and the study of temporal effects in general collaborative filtering settings (Silva and Carin (2012)). The extension of SPARFA-Trace to such applications is part of an on-going work.

8. Appendix

We derive the closed-form moment matching expressions for the approximate Kalman filtering approach detailed in Section 3.3. The following derivation can be seen as a multi-variate counterpart of the approach in (Rasmussen and Williams, 2006, Sec. 3.9).

We start by associating the K-dimensional latent variable vector $\mathbf{c}$ with a Gaussian prior $p(\mathbf{c}) = \mathcal{N}(\mathbf{c} \mid \mathbf{m}, \mathbf{V})$, where $\mathbf{m}$ and $\mathbf{V}$ are the prior's mean and covariance matrix, respectively. The observation likelihood takes the form $p(y \,|\, \mathbf{c}) = \Phi\left((2y-1)\left(\bar{\mathbf{w}}^T\mathbf{c} - \mu\right)\right)$. For simplicity of exposition, we will write $\widetilde{\bar{\mathbf{w}}} = (2y-1)\,\bar{\mathbf{w}}$ and $\widetilde{\mu} = (2y-1)\,\mu$ in the following derivations. According to Bayes rule, the posterior distribution of $\mathbf{c}$ given the observation y can be written as:

$$p(\mathbf{c}\,|\,y) = \frac{p(\mathbf{c})p(y\,|\,\mathbf{c})}{p(y)} = \frac{p(\mathbf{c})p(y\,|\,\mathbf{c})}{\int p(y\,|\,\mathbf{c})p(\mathbf{c})\mathrm{d}c}.$$

In order to approximate this posterior distribution of $\mathbf{c}$, we start by evaluating its denominator $p(y)$.

$$\begin{aligned}
p(y) &= \int p(y\,|\,\mathbf{c})p(\mathbf{c})\mathrm{d}\mathbf{c} \\
&= \int \Phi\left(\widetilde{\bar{\mathbf{w}}}^T\mathbf{c} - \widetilde{\mu}\right)\mathcal{N}\left(\mathbf{c}\,|\,\mathbf{m},\mathbf{V}\right)\mathrm{d}\mathbf{c} \\
&= \int\int_{-\infty}^{\infty} \mathcal{N}\left(t\,|\,0,1\right)\mathrm{d}t\,\mathcal{N}\left(\mathbf{c}\,|\,\mathbf{m},\mathbf{V}\right)\mathrm{d}\mathbf{c} \\
&= \frac{1}{\sqrt{2\pi}}\frac{1}{\sqrt{(2\pi)^K|\mathbf{V}|}}\int\int_{-\infty}^{\widetilde{\bar{\mathbf{w}}}^T\mathbf{c}-\widetilde{\mu}} e^{-t^2/2}\mathrm{d}t\, e^{-\frac{(\mathbf{c}-\mathbf{m})^T\mathbf{V}^{-1}(\mathbf{c}-\mathbf{m})}{2}}\mathrm{d}\mathbf{c}.
\end{aligned}$$

Now, substituting the variable $\mathbf{c}$ with $\mathbf{c}+\mathbf{m}$ and then, t with $t - \widetilde{\bar{\mathbf{w}}}^T\mathbf{c}$, we have

$$\begin{aligned}
p(y) &= \frac{1}{\sqrt{2\pi}}\frac{1}{\sqrt{(2\pi)^K|\mathbf{V}|}}\int\int_{-\infty}^{\widetilde{\bar{\mathbf{w}}}^T\mathbf{m}-\widetilde{\mu}} e^{-\frac{(t-\widetilde{\bar{\mathbf{w}}}^T\mathbf{c})^2}{2}}\mathrm{d}t\, e^{-\frac{\mathbf{c}^T\mathbf{V}^{-1}\mathbf{c}}{2}}\mathrm{d}\mathbf{c} \\
&= \frac{1}{\sqrt{(2\pi)^{K+1}|\mathbf{V}|}}\int_{-\infty}^{\widetilde{\bar{\mathbf{w}}}^T\mathbf{m}-\widetilde{\mu}}\int e^{-\frac{(t-\widetilde{\bar{\mathbf{w}}}^T\mathbf{c})^2+\mathbf{c}^T\mathbf{V}^{-1}\mathbf{c}}{2}}\mathrm{d}\mathbf{c}\mathrm{d}t \\
&= \int_{-\infty}^{\widetilde{\bar{\mathbf{w}}}^T\mathbf{m}-\widetilde{\mu}}\int \mathcal{N}\left(\begin{bmatrix} t \\ \mathbf{c}\end{bmatrix}|\mathbf{0},\begin{bmatrix} 1 & -\widetilde{\bar{\mathbf{w}}}^T \\ -\widetilde{\bar{\mathbf{w}}} & \widetilde{\bar{\mathbf{w}}}\widetilde{\bar{\mathbf{w}}}^T + \mathbf{V}^{-1}\end{bmatrix}^{-1}\right)\mathrm{d}\mathbf{c}\mathrm{d}t \\
&= \int_{-\infty}^{\widetilde{\bar{\mathbf{w}}}^T\mathbf{m}-\widetilde{\mu}}\mathcal{N}\left(t\,|\,0,1+\widetilde{\bar{\mathbf{w}}}^T\mathbf{V}\widetilde{\bar{\mathbf{w}}}\right)\mathrm{d}t = \Phi\left(\frac{\widetilde{\bar{\mathbf{w}}}^T\mathbf{m}-\widetilde{\mu}}{\sqrt{1+\widetilde{\bar{\mathbf{w}}}^T\mathbf{V}\widetilde{\bar{\mathbf{w}}}}}\right). \qquad (16)
\end{aligned}$$

In the last two steps of this derivation, we have used the Woodbury matrix identity (Horn and Johnson (1991)) and marginal Gaussian properties (Rasmussen and Williams (2006)). Since the posterior distribution is not Gaussian and prohibits the message passing procedure described in Section 3, our goal is to approximate it with a Gaussian distribution $q(\mathbf{c}) = \mathcal{N}(\mathbf{c} \mid \widehat{\mathbf{m}}, \widehat{\mathbf{V}})$ so that the message passing procedure is tractable. As shown in Rasmussen and Williams (2006), the specific values for $\widehat{\mathbf{m}}$ and $\widehat{\mathbf{V}}$ that minimizes the Kullback-Leibler (KL) divergence between $q(\mathbf{c})$ and $p(\mathbf{c} \mid y)$ are the first and second moments of the posterior $p(\mathbf{c}\,|\,y)$.

Next, we evaluate the first and second moments of the posterior distribution

$$p(\mathbf{c}\,|\,y) = p(y)^{-1}\Phi\left(\widetilde{\widetilde{\mathbf{w}}}^T\mathbf{c} - \widetilde{\mu}\right)\mathcal{N}\left(\mathbf{c}\,|\,\mathbf{m}, \mathbf{V}\right).$$

where $p(y)$ is given by (16). From (16) we can write

$$\Phi\left(\frac{\widetilde{\widetilde{\mathbf{w}}}^T\mathbf{m} - \widetilde{\mu}}{\sqrt{1 + \widetilde{\widetilde{\mathbf{w}}}^T\mathbf{V}\widetilde{\widetilde{\mathbf{w}}}}}\right) = \int \Phi\left(\widetilde{\widetilde{\mathbf{w}}}^T\mathbf{c} - \widetilde{\mu}\right)\mathcal{N}\left(\mathbf{c}\,|\,\mathbf{m}, \mathbf{V}\right)\mathrm{d}\mathbf{c}. \tag{17}$$

Taking the derivative with respect to $\mathbf{m}$ of both sides of (17) yields

$$\mathcal{N}\left(\frac{\widetilde{\widetilde{\mathbf{w}}}^T\mathbf{m} - \widetilde{\mu}}{\sqrt{1 + \widetilde{\widetilde{\mathbf{w}}}^T\mathbf{V}\widetilde{\widetilde{\mathbf{w}}}}}\right)\frac{\widetilde{\widetilde{\mathbf{w}}}}{\sqrt{1 + \widetilde{\widetilde{\mathbf{w}}}^T\mathbf{V}\widetilde{\widetilde{\mathbf{w}}}}} = \int \mathbf{V}^{-1}\left(\mathbf{c} - \mathbf{m}\right)\Phi\left(\widetilde{\widetilde{\mathbf{w}}}^T\mathbf{c} - \widetilde{\mu}\right)\mathcal{N}\left(\mathbf{c}\,|\,\mathbf{m}, \mathbf{V}\right)\mathrm{d}\mathbf{c}.$$

Let $z = \frac{\widetilde{\widetilde{\mathbf{w}}}^T\mathbf{m} - \widetilde{\mu}}{\sqrt{1 + \widetilde{\widetilde{\mathbf{w}}}^T\mathbf{V}\widetilde{\widetilde{\mathbf{w}}}}}$, we have

$$\mathcal{N}(z)\frac{\widetilde{\widetilde{\mathbf{w}}}}{\sqrt{1 + \widetilde{\widetilde{\mathbf{w}}}^T\mathbf{V}\widetilde{\widetilde{\mathbf{w}}}}} = \mathbf{V}^{-1}\int \mathbf{c}\;\Phi\left(\widetilde{\widetilde{\mathbf{w}}}^T\mathbf{c} - \widetilde{\mu}\right)\mathcal{N}\left(\mathbf{c}\,|\,\mathbf{m}, \mathbf{V}\right)\mathrm{d}\mathbf{c} - \mathbf{V}^{-1}\mathbf{m}\Phi(z).$$

Thus, the mean of the posterior distribution of $\mathbf{c}$ is given by:

$$\begin{aligned}
\mathbb{E}_{p(\mathbf{c}|y)}[\mathbf{c}] &= \int \mathbf{c}\,p(\mathbf{c}\,|\,y)\mathrm{d}\mathbf{c} \\
&= \int \mathbf{c}\frac{\Phi\left(\widetilde{\widetilde{\mathbf{w}}}^T\mathbf{c} - \widetilde{\mu}\right)\mathcal{N}\left(\mathbf{c}\,|\,\mathbf{m}, \mathbf{V}\right)}{p(y)}\mathrm{d}\mathbf{c} \\
&= \mathbf{m} + \frac{\mathbf{V}\widetilde{\widetilde{\mathbf{w}}}}{\sqrt{1 + \widetilde{\widetilde{\mathbf{w}}}^T\mathbf{V}\widetilde{\widetilde{\mathbf{w}}}}}\frac{\mathcal{N}(z)}{\Phi(z)}.
\end{aligned} \tag{18}$$

Similarly, taking the derivative with respect to $\mathbf{m}$ twice of both sides of (17) yields

$$\begin{aligned}
-z\mathcal{N}(z)\frac{\widetilde{\widetilde{\mathbf{w}}}\widetilde{\widetilde{\mathbf{w}}}^T}{1 + \widetilde{\widetilde{\mathbf{w}}}^T\mathbf{V}\widetilde{\widetilde{\mathbf{w}}}} &= -\mathbf{V}^{-1}\int \Phi\left(\widetilde{\widetilde{\mathbf{w}}}^T\mathbf{c} - \widetilde{\mu}\right)\mathcal{N}\left(\mathbf{c}\,|\,\mathbf{m}, \mathbf{V}\right)\mathrm{d}\mathbf{c} \\
&\quad + \mathbf{V}^{-1}\left(\int \left(\mathbf{c} - \mathbf{m}\right)\left(\mathbf{c} - \mathbf{m}\right)^T\Phi\left(\widetilde{\widetilde{\mathbf{w}}}^T\mathbf{c} - \widetilde{\mu}\right)\mathcal{N}\left(\mathbf{c} \mid \mathbf{m}, \mathbf{V}\right)\mathrm{d}\mathbf{c}\right)\mathbf{V}^{-1} \\
&= -\mathbf{V}^{-1}\Phi(z) + \mathbf{V}^{-1}\mathbb{E}_{p(\mathbf{c}|y)}[\mathbf{c}\mathbf{c}^T]\mathbf{V}^{-1}\Phi(z) \\
&\quad - \mathbf{V}^{-1}\left(\mathbb{E}_{p(\mathbf{c}|y)}[\mathbf{c}]\mathbf{m}^T + \mathbf{m}\mathbb{E}_{p(\mathbf{c}|y)}[\mathbf{c}]^T\right)\mathbf{V}^{-1}\Phi(z) \\
&\quad + \mathbf{V}^{-1}\mathbf{m}\mathbf{m}^T\mathbf{V}^{-1}\Phi(z),
\end{aligned}$$

where we implicitly used the fact that the covariance matrix $\mathbf{V}$ is symmetric. Therefore, we have

$$\mathbb{E}_{p(\mathbf{c}|y)}[\mathbf{c}\mathbf{c}^T] = \mathbf{V} + \mathbf{m}\mathbf{m}^T + \left(\mathbb{E}_{p(\mathbf{c}|y)}[\mathbf{c}]\mathbf{m}^T + \mathbf{m}\mathbb{E}_{p(\mathbf{c}|y)}[\mathbf{c}]^T\right) - z\frac{\mathcal{N}(z)}{\Phi(z)}\frac{\mathbf{V}\widetilde{\bar{\mathbf{w}}}\widetilde{\bar{\mathbf{w}}}^T\mathbf{V}}{1+\widetilde{\bar{\mathbf{w}}}^T\mathbf{V}\widetilde{\bar{\mathbf{w}}}}.$$

Thus, the covariance of the posterior distribution is given by

$$\begin{aligned}
&\mathbb{E}_{p(\mathbf{c}|y)}[\left(\mathbf{c} - \mathbb{E}_{p(\mathbf{c}|y)}[\mathbf{c}]\right)\left(\mathbf{c} - \mathbb{E}_{p(\mathbf{c}|y)}[\mathbf{c}]\right)^T] \\
&= \mathbb{E}_{p(\mathbf{c}|y)}[\mathbf{c}\mathbf{c}^T] - \mathbb{E}_{p(\mathbf{c}|y)}[\mathbf{c}]\mathbb{E}_{p(\mathbf{c}|y)}[\mathbf{c}]^T \\
&= \mathbf{V} + \mathbf{m}\mathbf{m}^T + \left(\mathbb{E}_{p(\mathbf{c}|y)}[\mathbf{c}]\mathbf{m}^T + \mathbf{m}\mathbb{E}_{p(\mathbf{c}|y)}[\mathbf{c}]^T\right) - z\frac{\mathcal{N}(z)}{\Phi(z)}\frac{\mathbf{V}\widetilde{\bar{\mathbf{w}}}\widetilde{\bar{\mathbf{w}}}^T\mathbf{V}}{1+\widetilde{\bar{\mathbf{w}}}^T\mathbf{V}\widetilde{\bar{\mathbf{w}}}} - \mathbb{E}_{p(\mathbf{c}|y)}[\mathbf{c}]\mathbb{E}_{p(\mathbf{c}|y)}[\mathbf{c}]^T \\
&= \mathbf{V} - \frac{\mathcal{N}(z)}{\Phi(z)}\left(z + \frac{\mathcal{N}(z)}{\Phi(z)}\right)\frac{\mathbf{V}\widetilde{\bar{\mathbf{w}}}\widetilde{\bar{\mathbf{w}}}^T\mathbf{V}}{1+\widetilde{\bar{\mathbf{w}}}^T\mathbf{V}\widetilde{\bar{\mathbf{w}}}}, \qquad (19)
\end{aligned}$$

where in the last step we have used (18) to simplify the expression.

Thus, given the prior distribution $p(\mathbf{c}) = \mathcal{N}(\mathbf{c} \mid \mathbf{m}, \mathbf{V})$ and the observation likelihood $p(y \mid \mathbf{c}) = \Phi\left((2y-1)\left(\bar{\mathbf{w}}^T\mathbf{c} - \mu\right)\right)$, we can approximate the posterior distribution $p(\mathbf{c} \mid y) \approx q(\mathbf{c}) = \mathcal{N}(\mathbf{c} \mid \widehat{\mathbf{m}}, \widehat{\mathbf{V}})$, with $\widehat{\mathbf{m}}$ and $\widehat{\mathbf{V}}$ as in (18) and (19), respectively.

Acknowledgments

Thanks to Joe Cavallaro, Kim Davenport and JP Slavinsky for providing the OpenStax Tutor (OST) data, and Andrew Waters and Ryan Ning for helpful discussions. This work was supported by the National Science Foundation under Cyberlearning grant IIS-1124535, the Air Force Office of Scientific Research under grant FA9550-09-1-0432, and the Google Faculty Research Award program. Visit our website `www.sparfa.com`, where you can learn more about the SPARFA project and purchase SPARFA t-shirts and other merchandise.

References

R. Baker, A. T. Corbett, and V. Aleven. More accurate student modeling through contextual estimation of slip and guess probabilities in Bayesian knowledge tracing. In *Proc. Intelligent Tutoring Systems*, pages 406–415. Springer, June 2008.

T. Barnes. The Q-matrix method: Mining student response data for knowledge. In *Proc. AAAI Workshop Educational Data Mining*, pages 1–8, July 2005.

A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Science*, 2(1):183–202, Mar. 2009.

C. M. Bishop and N. M. Nasrabadi. *Pattern recognition and machine learning*. Springer New York, 2006.

M. Carrier and H. Pashler. The influence of retrieval on retention. *Memory & Cognition*, 20(6):633–642, Nov. 1992.

H. Cen, K. R. Koedinger, and B. Junker. Learning factors analysis–a general method for cognitive model evaluation and improvement. In M. Ikeda, K. D. Ashley, and T. W. Chan, editors, *Intelligent Tutoring Systems*, volume 4053 of *Lecture Notes in Computer Science*, pages 164–175. Springer, June 2006.

A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4(4):253–278, Dec. 1994.

G. A. Einicke and L. B. White. Robust extended Kalman filtering. *IEEE Trans. on Signal Processing*, 47(9):2596–2599, Sep. 1999.

K. Fronczyk, A. E. Waters, M. Guindani, R. G. Baraniuk, and M. Vannucci. A Bayesian infinite factor model for learning and content analytics. *Computational Statistics and Data Analysis*, June 2013, submitted.

Y. Gong, J. E. Beck, and N. T. Heffernan. Comparing knowledge tracing and performance factor analysis by using multiple model fitting procedures. In *Intelligent Tutoring Systems*, pages 35–44. Springer, June 2010.

J. P. González-Brenes and J. Mostow. Dynamic cognitive tracing: Towards unified discovery of student and cognitive models. In *Proc. 5th Intl. Conf. on Educational Data Mining*, pages 49–56, June 2012.

J. P. González-Brenes and J. Mostow. What and when do students learn? Fully data-driven joint estimation of cognitive and student models. In *Proc. 6th Intl. Conf. on Educational Data Mining*, pages 236–239, July 2013.

H. H. Harman. *Modern Factor Analysis*. University of Chicago Press, 1976.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2010.

S. S. Haykin. *Kalman filtering and neural networks*. Wiley Online Library, 2001.

G. Hooker, M. Finkelman, and A. Schwartzman. Paradoxical results in multidimensional item response theory. *Psychometrika*, 74(3):419–442, Sep. 2009.

R. A. Horn and C. R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, 1991.

A. H. Jazwinski. *Stochastic Processes and Filtering Theory*. Academic Press, New York, 1970.

P. Jordan and M. Spiess. Generalizations of paradoxical results in multidimensional item response theory. *Psychometrika*, 77(1):127–152, Jan. 2012.

S. J. Julier and J. K. Uhlmann. New extension of the Kalman filter to nonlinear systems. In *AeroSense'97: The 11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls*, pages 182–193, Apr. 1997.

R. E. Kalman. A new approach to linear filtering and prediction problems. *ASME Journal of basic Engineering*, 82(1):35–45, 1960.

Knewton. Knewton adaptive learning: Building the world's most powerful recommendation engine for education. *online: http://www.knewton.com/adaptive-learning-white-paper/*, June 2012.

J. Knox, S. Bayne, H. MacLeod, J. Ross, and C. Sinclair. MOOC pedagogy: the challenges of developing for coursera. *Online Newsletter of the Association for Learning Technologies*, Aug. 2012.

F. R. Kschischang, B. J. Frey, and H-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Trans. on Information Theory*, 47(2):498–519, Feb. 2001.

A. S. Lan, A. E. Waters, C. Studer, and R. G. Baraniuk. Sparse factor analysis for learning and content analytics. *arXiv preprint:1303.5685v2*, Oct. 2012.

A. S. Lan, C. Studer, A. E. Waters, and R. G. Baraniuk. Tag-aware ordinal sparse factor analysis for learning and content analytics. In *Proc. 6th Intl. Conf. on Educational Data Mining*, pages 90–97, July 2013a.

A. S. Lan, C. Studer, A. E. Waters, and R. G. Baraniuk. Joint topic modeling and factor analysis of textual information and graded response data. In *Proc. 6th Intl. Conf. on Educational Data Mining*, pages 324–325, July 2013b.

D. P. Larsen, A. C. Butler, and H. L. Roediger III. Repeated testing improves long-term retention relative to repeated study: a randomised controlled trial. *Medical education*, 43 (12):1174–1181, Dec. 2009.

H-A. Loeliger. An introduction to factor graphs. *IEEE Signal Processing Magazine*, 21(1): 28–41, Jan. 2004.

F. M. Lord. *Applications of Item Response Theory to Practical Testing Problems*. Erlbaum Associates, 1980.

F. G. Martin. Will massive open online courses change how we teach? *Communications of the ACM*, 55(8):26–28, Aug. 2012.

P. S. Maybeck. *Stochastic Models, Estimation and Control, Vol.1*. Academic Press, New York, 1979.

T. P. Minka. From hidden Markov models to linear dynamical systems. Technical report, 1999. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.51.1207&rep=rep1&type=pdf.

T. P Minka. Expectation propagation for approximate Bayesian inference. In *Proceedings of the 17th conference on Uncertainty in Artificial Intelligence*, pages 362–369, Aug. 2001.

OpenStaxTutor. Openstax tutor, Sep. 2013. URL `https://openstaxtutor.org/`.

Z. A. Pardos and N. T. Heffernan. Modeling individualization in a Bayesian networks implementation of knowledge tracing. In *Proc. 18th Intl. Conf. on User Modeling, Adaptation, and Personalization*, pages 255–266. Springer, June 2010a.

Z. A. Pardos and N. T. Heffernan. Navigating the parameter space of Bayesian knowledge tracing models: Visualizations of the convergence of the expectation maximization algorithm. In *Proc. 3rd Intl. Conf. on Educational Data Mining*, pages 161–170, June 2010b.

J. Psotka, L. D. Massey, and S. A. Mutter. *Intelligent Tutoring Systems: Lessons Learned*. Lawrence Erlbaum Associates, 1988.

Y. Qi. *Extending expectation propagation for graphical models*. PhD thesis, Massachusetts Institute of Technology, 2004.

G. Rasch. *Probabilistic Models for Some Intelligence and Attainment Tests*. MESA Press, 1993.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Process for Machine Learning*. MIT Press, 2006.

S. Roweis and Z. Ghahramani. Learning nonlinear dynamical systems using the expectation–maximization algorithm. *Kalman filtering and neural networks*, 6:175–220, 2001.

J. Silva and L. Carin. Active learning for online bayesian matrix factorization. In *Proc. 18th ACM SIGKDD Intl. Conf. on Knowledge discovery and data mining*, pages 325–333, Aug. 2012.

W. J. van der Linden. Bayesian item selection criteria for adaptive testing. *Psychometrika*, 63(2):201–216, June 1998.

K. VanLehn, C. Lynch, K. Schulze, J. A. Shapiro, R. Shelby, L. Taylor, D. Treacy, A. Weinstein, and M. Wintersgill. The Andes physics tutoring system: Lessons learned. *Intl. Journal of Artificial Intelligence in Education*, 15(3):147–204, 2005.

E. A. Wan and R. Van Der Merwe. The unscented Kalman filter for nonlinear estimation. In *Adaptive Systems for Signal Processing, Communications, and Control Symposium*, pages 153–158, Oct. 2000.

E. Wang, E. Salazar, D. Dunson, and L. Carin. Spatio-temporal modeling of legislation and votes. *Bayesian Analysis*, 8(1):233–268, Mar. 2013.

B. Weiner and H. Reed. Effects of the instructional sets to remember and to forget on short-term retention: Studies of rehearsal control and retrieval inhibition (repression). *Journal of Experimental Psychology*, 79(2):226, Feb. 1969.

R. Wolfinger. Laplace's approximation for nonlinear mixed models. *Biometrika*, 80(4): 791–795, Dec. 1993.