Time-varying Learning and Content Analytics via Sparse Factor Analysis

Andrew S. Lan
Rice University
Houston, TX 77005
mr.lan@sparfa.com

Christoph Studer
Cornell University
Ithaca, NY 14853
studer@sparfa.com

Richard G. Baraniuk
Rice University
Houston, TX 77005
richb@sparfa.com

ABSTRACT

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

Keywords

Expectation maximization, Kalman filter, learning analytics, personalized learning, sparse factor analysis

1. INTRODUCTION

The traditional "one-size-fits-all" approach to education is a major bottleneck to improving learning outcomes worldwide. Fortunately, significant progress has been made over the past few decades on new technologies that provide timely feedback to learners as they follow personalized learning pathways through nonlinearly interconnected learning contents. Increasingly, these technologies are based on *machine learning algorithms* that automatically mine data from a potentially large number of learners interacting with learning contents (see [14, 15] for examples).

In our view, a *personalized learning system* (PLS) consists of two key components: (i) *learning analytics* (LA), which estimate each learner's knowledge state and dynamically trace its changes over time, as they either *learn* by interacting with various *learning resources* (e.g., textbook sections, lecture videos, labs) and *questions* (e.g., in quizzes, homework assignments, exams, and other assessments), or *forget* (see [30]), and (ii) *content analytics* (CA), which provide insight on the quality, difficulty, and organization of the learning resources and questions.

1.1 SPARse Factor Analysis (SPARFA)

The recently developed sparse factor analysis (SPARFA) framework [18] proposes a set of statistical model and algorithms for machine learning-based LA and CA. SPARFA models $Y_{i,j}$, the binary-valued graded response of learner j to question i, as a Bernoulli random variable (with 1 representing a correct response and 0 an incorrect one):

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

Here, $\Phi(\cdot)$ is the inverse logit/probit link function, and the slack variable $Z_{i,j}$ depends on three factors: (i) the *question–concept association vector* $\mathbf{w}_i$ which characterizes how question i relates to each abstract concept, (ii) the *learner concept knowledge vector* $\mathbf{c}_j$ of learner j, and (iii) the *intrinsic difficulty* μ_i of question i. Given a dataset of graded learner response data $\mathbf{Y}$, SPARFA jointly estimates $\mathbf{c}_j, \forall j$ to effect LA and $\mathbf{w}_i$ and μ_i, $\forall i$ to effect CA.

While powerful, the SPARFA framework has two important limitations. First, it assumes that the learners' concept knowledge states remain *constant* over time; this reduces it's efficacy when applied to scenarios, where learners learn (and forget) concepts over time (weeks, months, years, decades) [4]. Second, SPARFA models only the learners' interactions with questions, which measure concept knowledge states, and not other kinds of learning opportunities, such as reading a textbook, viewing a lecture video, or conducting a laboratory or Gedankenexperiment; this complicates its application in automatically recommending new resources to individual learners for remedial or enrichment studies.

1.2 SPARFA-Trace: Time-varying LA and CA

In this paper, we extend the SPARFA framework to address these limitations. We develop *SPARFA-Trace*, an online estimation algorithm that jointly performs *time-varying*

KDD'14, August 24–27, 2014, New York, NY, USA.

http://dx.doi.org/10.1145/2623330.2623631.

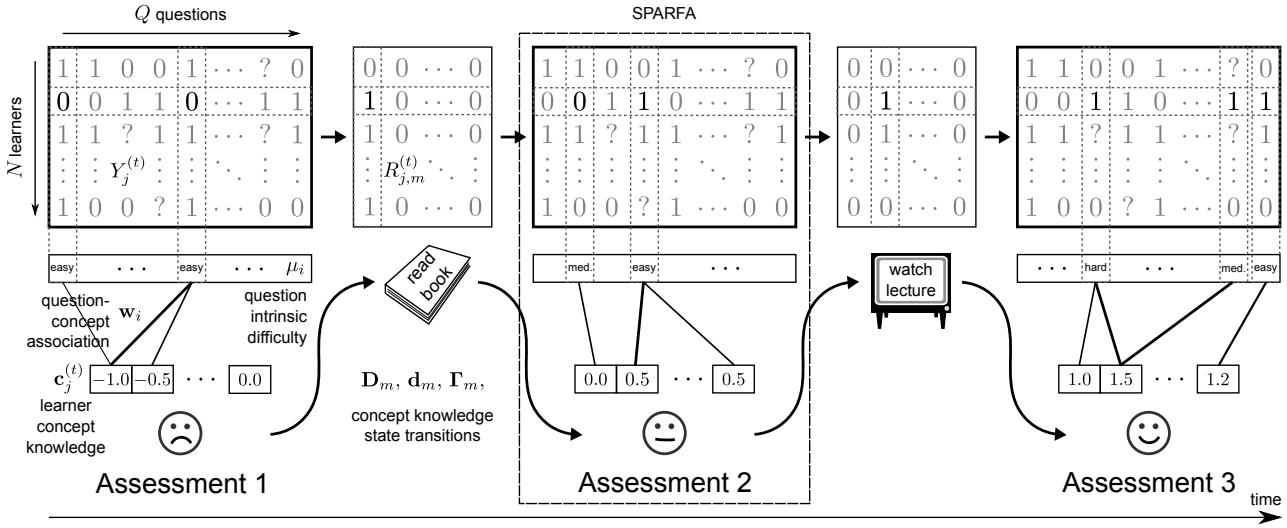


Figure 1: The SPARFA-Trace framework processes the graded learner response matrix $\mathbf{Y}$ (binary-valued, with '1' denoting a correct response, '0' an incorrect one, and '?' indicates an unobserved one) and the learner activity matrices $\{\mathbf{R}^{(t)}\}$ (binary-valued, with '1' denoting that a learner studied a particular learning resource, and '0' otherwise). Upon analyzing this data, SPARFA-Trace jointly traces the learner concept knowledge states $\mathbf{c}_j^{(t)}$ (a happy face represents high concept knowledge, and a sad face represents low concept knowledge) over time, and estimates the learning resource content organization and quality parameters $\mathbf{D}_m$, $\mathbf{d}_m$ and $\mathbf{\Gamma}_m$, together with question–concept association parameters $\mathbf{w}_i$ and question difficulty parameters μ_i.

LA and CA. The core machinery is based on blind approximate Kalman filtering. The working principles of SPARFA-Trace are illustrated in Figure 1. Time-varying LA is performed by tracing the evolution of each learner's concept knowledge state vector $\mathbf{c}_j^{(t)}$ over time t, based on observed binary-valued (correct/incorrect) graded learner responses to questions matrix $\mathbf{Y}$ and on the learner activity matrices $\mathbf{R}^{(t)}$. CA is performed by estimating the learner concept knowledge state transition parameters $\mathbf{D}_m$, $\mathbf{d}_m$, and $\mathbf{\Gamma}_m$, the question–concept associations and the question intrinsic difficulties $\mathbf{w}_i$ and μ_i, based on the estimated learner concept knowledge states at all time instances.

Tracing the learners' concept knowledge states over time is non-trivial due to the fact that the observations are noisy, binary-valued graded learner responses to questions. To perform this on-line estimation process, we develop a novel message passing-based algorithm in Section 3 that employs an approximate Kalman filter [12]. Furthermore, the underlying state-transition and observation parameters are, in general, unknown in real educational scenarios. Therefore, in Section 4, we introduce a set of novel convex optimization-based algorithms to estimate these parameters directly (and solely) from learner response data.

To test and validate the effectiveness of SPARFA-Trace, we conduct a series of validation experiments in Section 5 using real-world educational datasets collected with OpenStax Tutor [3, 21]. We show that SPARFA-Trace can effectively trace learner concept knowledge, estimate learner concept knowledge state transition parameters, and estimate the question-dependent parameters. Furthermore, we show that it achieves comparable or better performance than existing approaches on predicting unobserved learner responses.

1.3 Related work in knowledge tracing

The closest related work to SPARFA-Trace is *knowledge tracing* (KT), a popular technique for tracing learner knowledge evolution over time and for predicting future learner performance (see, e.g., [5, 22]). Powerful as it is, KT suffers from three drawbacks. First, KT uses binary learner knowledge state representations, characterizing learners as to whether they have mastered a certain concept or not, which provides limited explanatory power. Second, KT assumes that each question is associated with exactly one concept. This restriction limits KT to very narrow educational domains and prevents it from generalizing to typical courses/assessments involving multiple concepts. Third, KT uses a single "probability of learning" parameter to characterize learner knowledge state transitions. This limits KT's ability to analyze the quality and organization of different learning resources that would lead to different learner knowledge state transitions.

2. SPARFA-TRACE STATISTICAL MODEL

We start by extending SPARFA [18] to model learner concept knowledge evolution over time. Then, in Section 2.2, we characterize the transitions of a learner's concept knowledge states between consecutive time instances as an affine model.

2.1 Model for graded learner responses

The SPARFA-Trace 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 vector $\mathbf{c}_j^{(t)} \in \mathbb{R}^K, t \in \{1, \ldots, T\}, j \in \{1, \ldots, N\}$, 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 $i(t,j) : \{1,\ldots,T\} \times \{1,\ldots,N\} \mapsto \{1,\ldots,Q\}$, 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 at time instance t as $i_j^{(t)}$. Under this notation, we define the K-dimensional vector $\mathbf{w}_{i_j^{(t)}} \in \mathbb{R}^K, i \in \{1,\ldots,Q\}$, as the question–concept association vector of this question. Finally, we define the scalar $\mu_{i_j^{(t)}} \in \mathbb{R}$ to be the intrinsic difficulty of question $i_j^{(t)}$, with positive values of $\mu_{i_j^{(t)}}$ representing difficult questions, and negative $\mu_{i_j^{(t)}}$ representing easy ones.

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

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

Here, the set $\Omega_{\text{obs}} \subseteq \{1,\ldots,T\} \times \{1,\ldots,N\}$ contains the indices associated with the observed graded learner response data, since some 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 standard normal distribution. (The inverse logit link function could also be used; the inverse probit link function is preferred because it simplifies the calculations in Section 3.2.) The likelihood of an observation $Y_j^{(t)}$ can, alternatively, be written as

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

a shorthand expression we will use in what follows.

Following the original SPARFA framework [18], we impose the following model assumptions:

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

(A2) *The vector $\mathbf{w}_i$ is sparse*: This assumption is based 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 $\mathbf{w}_i$ 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 representing high concept knowledge, and negative values representing low concept knowledge.

These assumptions are reasonable in most real-world educational scenarios and alleviate the common *identifiability* issue inherent to factor analysis (if $Z_{i,j} = \mathbf{w}_i^T \mathbf{c}_j$, then we have $Z_{i,j} = \mathbf{w}_i^T \mathbf{Q}^T \mathbf{Q} \mathbf{c}_j = \widetilde{\mathbf{w}}_i^T \widetilde{\mathbf{c}}_j$ for any orthonormal matrix $\mathbf{Q}$. Hence, the estimation of $\mathbf{w}_i$ and $\mathbf{c}_j$ is non-unique up to a unitary transformation). The assumptions also improve the *interpretability* of the variables $\mathbf{w}_i$, $\mathbf{c}_j$, and μ_i.

2.2 Model for state transitions

In this section, we propose a latent state transition model that characterizes the learner concept knowledge evolution between two consecutive time instances. We assume here that the concept knowledge state evolves for two primary reasons: (i) A learner may interact with learning resources (e.g., read a section of an assigned textbook, watch a lecture video, conduct a lab experiment, run a computer simulation, etc.), all of which are likely to result in an increase of their concept knowledge. (ii) A learner may 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 [30] as a special learning resource that reduces learners' concept knowledge over time.

We assume that there are a total 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. Armed with this notation, 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 $m_j^{(t)}$ between time instances t and $t+1$, 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

$$\begin{aligned} &p(\mathbf{c}_j^{(t)} \,|\, \mathbf{c}_j^{(t-1)}) = \\ &\mathcal{N}\Big(\mathbf{c}_j^{(t)} \,|\, (\mathbf{I}_K + \mathbf{D}_{m_j^{(t-1)}})\mathbf{c}_j^{(t-1)} + \mathbf{d}_{m_j^{(t-1)}},\ \mathbf{\Gamma}_{m_j^{(t-1)}}\Big), \end{aligned} \tag{2}$$

where $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ represents a multivariate Gaussian distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. $\mathbf{I}_K$ is the $K \times K$ identity matrix; $\mathbf{D}_{m_j^{(t-1)}}$, $\mathbf{d}_{m_j^{(t-1)}}$, and $\mathbf{\Gamma}_{m_j^{(t-1)}}$ 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, and $\mathbf{d}_{m_j^{(t-1)}}$ is a $K \times 1$ vector. The covariance matrix $\mathbf{\Gamma}_{m_j^{(t-1)}}$ characterizes the uncertainty induced in the learner concept knowledge state transition by interacting with learning resource $m_j^{(t-1)}$.

In order to reduce the number of parameters and to improve identifiability of the parameters $\mathbf{D}_{m_j^{(t-1)}}$, $\mathbf{d}_{m_j^{(t-1)}}$ and $\mathbf{\Gamma}_{m_j^{(t-1)}}$, 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 covered early in the course, while lower entries represent advanced concepts covered towards the end of the course.

(A5) $\mathbf{D}_{m_j^{(t-1)}}$ *has non-negative entries*: This assumption ensures, for example, that having low concept knowl-

edge 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); this 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 be modeled as a learning resource with negative entries in $\mathbf{d}_{m_j^{(t-1)}}$. To further reduce the number of parameters, we assume that the covariance matrix $\mathbf{\Gamma}_{m_j^{(t-1)}}$ is diagonal.

3. TIME-VARYING LA

Time-varying LA requires an on-line algorithm [13] that traces the evolution of learner concept knowledge over time. Designing such an algorithm is complicated by the fact that the binary-valued graded learner responses correspond to a non-linear and non-Gaussian observation model. The particle filter [6] is an on-line state estimation algorithm in non-linear and non-Gaussian systems, which uses a set of Monte-Carlo particles to approximate the latent state distribution. However, its excessive computational complexity prevents it from being applied to personalized learning at large scale (especially if immediate feedback is required). On the contrary, the Kalman filter [12] is an efficient on-line state estimation algorithm for linear dynamical systems (LDSs) with Gaussian observations, but it cannot be directly applied to time-varying LA because of the non-linear, non-Gaussian observation model (1).

In order to recast the time-varying LA problem as an approximate Kalman filter, we next introduce a set of approximations that build upon ideas in expectation propagation [20, 23]. We begin in Section 3.1 by reviewing the key elements of the Kalman filtering and smoothing approach, and then detail our approximate Kalman filter in Section 3.2. For notational simplicity, we will omit the learner index j in this section, i.e., the 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 and smoothing

The Kalman filter [9, 12] solves the problem of state estimation in LDSs, where the system consist of a series of continuous latent state variables that are separated by linear state transitions; the state observations are corrupted by Gaussian noise. We briefly summarize the main findings from [19]. Let the LDS consists of a series of T latent state variables $\mathbf{c}^{(t)}, t = 1, \ldots, T$, and observations $\mathbf{y}^{(t)}, t = 1, \ldots, T$. The factor graph [17, 28] associated to this LDS is visualized in Figure 2. The latent states (denoted by dashed circles) form a Markov chain, meaning that the next state only depends on the current state but not on previous ones. The Kalman filter estimation

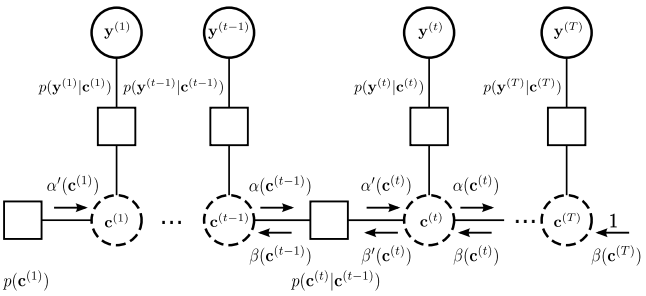


Figure 2: Factor graph message passing algorithm for the estimation of a set of T latent state variables with Markovian transition properties from (possibly noisy) observations.

procedure of the variables $\mathbf{c}^{(t)}, \forall t$ based on the observations $\mathbf{y}^{(t)}, \forall t$ (denoted by solid circles) can be formulated as a message-passing algorithm that consists of two phases. First, a forward message passing phase (i.e., the Kalman filtering phase) is performed. Then, using the 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 performed.

In the forward message passing phase (see Figure 2), 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 left-right message passing algorithm outlined in Figure 2.

In Figure 2, the outgoing message $\alpha(\mathbf{c}^{(t)})$ from variable node $\mathbf{c}^{(t)}$ is given by [19]

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

where $b^{(t)} = p(\mathbf{y}^{(t)} \mid \mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(t-1)})$ is a scaling factor. We can see that a scaled version of $\alpha(\mathbf{c}^{(t)})$, $\widehat{\alpha}(\mathbf{c}^{(t)}) = \frac{\alpha(\mathbf{c}^{(t)})}{\prod_{\tau=1}^{t} b^{(\tau)}} = p(\mathbf{c}^{(t)} \mid \mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(t)})$, is exactly the value of interest, which can be obtained in recursive fashion via

$$b^{(t)}\, \widehat{\alpha}(\mathbf{c}^{(t)}) = p(\mathbf{y}^{(t)} \,|\, \mathbf{c}^{(t)}) \int p(\mathbf{c}^{(t)} \,|\, \mathbf{c}^{(t-1)})\, \widehat{\alpha}(\mathbf{c}^{(t-1)}) \mathrm{d}\mathbf{c}^{(t-1)}. \tag{3}$$

The key to obtaining a tractable and efficient estimator for $p(\mathbf{c}^{(t)} \mid \mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(t)})$ is that the transition probability $p(\mathbf{c}^{(t)} \mid \mathbf{c}^{(t-1)})$ and the observation likelihood $p(\mathbf{y}^{(t)} \mid \mathbf{c}^{(t)})$ 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 satisfies this requirement, in which the transition probability and the observation likelihood are (multivariate) Gaussians of the following form:

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

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. The functional form of the messages is 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 (3) is given by

$$b^{(t)}\,\widehat{\alpha}(\mathbf{c}^{(t)}) = \mathcal{N}\Big(\mathbf{c}^{(t)}\,|\,\mathbf{m}^{(t)}, \mathbf{V}^{(t)}\Big)\,, \tag{4}$$

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

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

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

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

The recursion starts with $p(\mathbf{c}^{(1)}) = \mathcal{N}(\mathbf{c}^{(1)} \mid \mathbf{m}^{(0)}, \mathbf{V}^{(0)})$, where we assume $\mathbf{c}^{(1)}$ to be $\mathbf{m}^{(0)} = \mathbf{0}_K$ and $\mathbf{V}^{(0)} = \sigma_0^2\mathbf{I}_K$.

Kalman smoothing uses future observations $\mathbf{y}^{(\tau)}, \tau > t$ to obtain a better estimate of the latent state at time instance t. In other words, the value of interest is now $p(\mathbf{c}^{(t)} \mid \mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(T)})$. In order to estimate this value, a set of backward recursions similar to the set of forward recursions (3) can be used:

$$\begin{aligned}
&\widehat{\alpha}(\mathbf{c}^{(t-1)})\widehat{\beta}(\mathbf{c}^{(t-1)}) = \widehat{\alpha}(\mathbf{c}^{(t-1)}) \\
&\int_{\mathbf{c}^{(t)}} p(\mathbf{c}^{(t)}\,|\,\mathbf{c}^{(t-1)})p(\mathbf{y}^{(t)}\,|\,\mathbf{c}^{(t)})\frac{\widehat{\alpha}(\mathbf{c}^{(t)})\widehat{\beta}(\mathbf{c}^{(t)})}{b^{(t)}\,\widehat{\alpha}(\mathbf{c}^{(t)})}\mathrm{d}\mathbf{c}^{(t)},
\end{aligned} \tag{5}$$

where $\widehat{\beta}(\mathbf{c}^{(t)}) = \frac{p(\mathbf{y}^{(t+1)},\ldots,\mathbf{y}^{(T)}|\mathbf{c}^{(t)})}{\prod_{\tau=t+1}^{T} b^{(\tau)}}$. The quantity of interest $p(\mathbf{c}^{(t)}\,|\,\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(T)}) = \widehat{\alpha}(\mathbf{c}^{(t)})\widehat{\beta}(\mathbf{c}^{(t)})$ can be computed recursively as a backward message passing process, given the estimates (4) following the completion of the forward message passing process detailed above.

Specifically, in an LDS, the recursions take the form:

$$\widehat{\alpha}(\mathbf{c}^{(t-1)})\widehat{\beta}(\mathbf{c}^{(t-1)}) = \mathcal{N}(\mathbf{c}^{(t-1)}\,|\,\widehat{\mathbf{m}}^{(t-1)}, \widehat{\mathbf{V}}^{(t-1)}), \tag{6}$$

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

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

with $\widehat{\mathbf{m}}^{(T)} = \mathbf{m}^{(T)}$ and $\widehat{\mathbf{V}}^{(T)} = \mathbf{V}^{(T)}$.

3.2 Approximate Kalman filtering for LA

The basic Kalman filtering and smoothing, i.e., (4) and (6) are only suitable for applications with a Gaussian latent state transition model and a Gaussian observation model, while the forward and backward recursions (3) and (5) hold 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. We will now detail approximations that enable the estimation of latent learner concept knowledge states for 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 (2). Therefore, the recursion formula for the forward message passing process (3) becomes

$$\begin{aligned}
b^{(t)}\widehat{\alpha}(\mathbf{c}^{(t)}) &= p(Y^{(t)}\,|\,\mathbf{c}^{(t)})\int p(\mathbf{c}^{(t)}\,|\,\mathbf{c}^{(t-1)})\,\widehat{\alpha}(\mathbf{c}^{(t-1)})\mathrm{d}\mathbf{c}^{(t)} \\
&= \Phi\Big(\Big(2Y^{(t)}-1\Big)\Big(\mathbf{w}_{i^{(t)}}^T\mathbf{c}^{(t)} - \mu_{i^{(t)}}\Big)\Big)\mathcal{N}\Big(\mathbf{c}^{(t)}|\,\widetilde{\mathbf{m}}^{(t)}, \widetilde{\mathbf{V}}^{(t)}\Big)\,,
\end{aligned} \tag{7}$$

where $\widetilde{\mathbf{m}}^{(t)} = \overline{\mathbf{D}}_{m^{(t-1)}}\mathbf{m}^{(t-1)} + \mathbf{d}_{m^{(t-1)}}$ and $\widetilde{\mathbf{V}}^{(t)} = \overline{\mathbf{D}}_{m^{(t-1)}}\mathbf{V}^{(t-1)}\overline{\mathbf{D}}_{m^{(t-1)}}^T + \mathbf{\Gamma}_{m^{(t-1)}}$.

Equation 7 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 (4) and (6) 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 $\mathbf{c}^{(t)}$.

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) [7, 10], 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) [11, 27], 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 approximation [23, 31], which use an iterative algorithm to find the mode of $\widehat{\alpha}(\mathbf{c}^{(t)})$ and the Hessian at the mode to approximate the mean and covariance of the approximated Gaussian messages. We will employ an approximation approach first introduced in the expectation propagation (EP) literature [20].

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 $q(\mathbf{c})$ [23]. Fortuneately, for the probit observation model $p(Y^{(t)}\,|\,\mathbf{c}^{(t)}) = \Phi\Big(\Big(2Y^{(t)}-1\Big)\Big(\mathbf{w}_{i^{(t)}}^T\mathbf{c}^{(t)} - \mu_{i^{(t)}}\Big)\Big)$, $\overline{\mathbf{m}}^{(t)}$, $\overline{\mathbf{V}}^{(t)}$ and $b^{(t)}$ have closed-form expressions (details omitted for simplicity):

$$\begin{aligned}
\overline{\mathbf{m}}^{(t)} &= \widetilde{\mathbf{m}}^{(t)} + \Big(2Y^{(t)}-1\Big)\frac{\widetilde{\mathbf{V}}^{(t)}\mathbf{w}_{i^{(t)}}}{\sqrt{1 + \mathbf{w}_{i^{(t)}}^T\widetilde{\mathbf{V}}^{(t)}\mathbf{w}_{i^{(t)}}}}\frac{\mathcal{N}(z)}{\Phi(z)}, \\
\overline{\mathbf{V}}^{(t)} &= \widetilde{\mathbf{V}}^{(t)} - \frac{\widetilde{\mathbf{V}}^{(t)}\mathbf{w}_{i^{(t)}}\mathbf{w}_{i^{(t)}}^T\widetilde{\mathbf{V}}^{(t)}}{1 + \mathbf{w}_{i^{(t)}}^T\widetilde{\mathbf{V}}^{(t)}\mathbf{w}_{i^{(t)}}}\Big(z + \frac{\mathcal{N}(z)}{\Phi(z)}\Big)\frac{\mathcal{N}(z)}{\Phi(z)}, \\
b^{(t)} &= \Phi(z),
\end{aligned} \tag{8}$$

with

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

and $\widetilde{\mathbf{m}}^{(t)}$, $\widetilde{\mathbf{V}}^{(t)}$ as given by (7).

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. Therefore, we will focus on the inverse probit link function in the sequel.

Armed with the efficient approximation (8), the forward Kalman filtering and backward Kalman smoothing message passing scheme described in Section 3.1 can be applied to the problem at hand. Using these recursions, estimates of the desired quantities $p(\mathbf{c}^{(t)} \mid \mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(T)})$ can be computed efficiently, providing a way for learner concept knowledge tracing under the model (1).

4. TIME-VARYING CA

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 method proposed in Section 3 is only able to provide these estimates if 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, $\mathbf{w}_i$ and μ_i, $\forall i$, are given a priori.

However, in a typical PLS, these parameters are unknown and need to be estimated from the observed data. We now detail a set of convex optimization-based techniques to estimate the parameters $\mathbf{D}_m$, $\mathbf{d}_m$, and $\mathbf{\Gamma}_m$, $\forall m$, and $\mathbf{w}_i$, μ_i, $\forall i$,[1] 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. The techniques we detail in this chapter allows SPARFA-Trace to jointly trace learner concept knowledge and estimate learner, learning resource, and question-dependent parameters, using an expectation-maximization (EM) approach.

4.1 SPARFA-Trace as an EM algorithm

EM has been widely used in the Kalman filtering framework to estimate the parameters of interest in the system (see [2, Chap. 13] and [9] for more details) due to numerous practical advantages [24]. SPARFA-Trace performs parameter estimation in an iterative fashion in the EM framework. All parameters are initialized by random 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 $p(\mathbf{c}_j^{(t)} \mid \mathbf{y}_j^{(1)}, \ldots, \mathbf{y}_j^{(T)})$, $\forall t, j$, and (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.,

$$\underset{\mathbf{D}_m, \mathbf{d}_m, \mathbf{\Gamma}_m, \forall m, \mathbf{w}_i, \mu_i, \forall i}{\text{maximize}} \sum_{t=2}^{T} \sum_{j=1}^{N} \mathbb{E}_{\mathbf{c}_j^{(t-1)}, \mathbf{c}_j^{(t)}} \Big[\log p(\mathbf{c}_j^{(t)} \mid \mathbf{c}_j^{(t-1)}, \mathbf{D}_{m_j^{(t-1)}}, \mathbf{d}_{m_j^{(t-1)}}, \mathbf{\Gamma}_{m_j^{(t-1)}}) \Big] + \sum_{(t,j) \in \Omega_{\text{obs}}} \mathbb{E}_{\mathbf{c}_j^{(t)}} \big[\log p(Y_j^{(t)} \mid \mathbf{c}_j^{(t)}, \mathbf{w}_{i_j^{(t)}}, \mu_{i_j^{(t)}}) \big], \quad (9)$$

in order to obtain new 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 state transition parameters

[1]The estimation of the learner initial knowledge parameters $\mathbf{m}_j^{(0)}, \mathbf{V}_j^{(0)}$, $\forall j$ is trivial and can be found in [2].

We start by estimating 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 maximize the expected log-likelihood (9) with respect to $\mathbf{D}_m$ and $\mathbf{d}_m$, subject to the assumptions (A4)–(A6). We start by estimating $\mathbf{D}_m$ and $\mathbf{d}_m$ given $\mathbf{\Gamma}_m$. In order to induce sparsity on $\mathbf{D}_m$ to take (A6) into account, we impose an ℓ_1-norm penalty on $\mathbf{D}_m$, which is defined as the sum of the absolute values of all entries of $\mathbf{D}_m$ [8]. Taking only the terms containing $\mathbf{D}_m$ and $\mathbf{d}_m$, we can formulate the following augmented optimization problem:

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

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 this notation.

The problem (P_d) is convex in $\widetilde{\mathbf{D}}_m$, and hence, can be solved efficiently. In particular, we use the iterative fast iterative shrinkage and thresholding algorithm (FISTA) framework [1]. In each iteration $\ell = 1, 2, \ldots, L_{\max}$, the algorithm performs two steps. First, a gradient step that aims to lower the objective function performs

$$\widehat{\mathbf{D}}_m^{\ell+1} \leftarrow \widetilde{\mathbf{D}}_m^{\ell} - \eta_\ell \nabla f(\widetilde{\mathbf{D}}_m), \quad (10)$$

where $f(\widetilde{\mathbf{D}}_m)$ corresponds to the differentiable part of the objective function (excluding the ℓ_1-norm penalty) in (P_d). The quantity η_ℓ is a step size parameter for iteration ℓ. Details on how to choose η_ℓ can be found in [1]. The gradient $\nabla f(\widetilde{\mathbf{D}}_m)$ in (10) is given by

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

where 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 (6). 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):

$$\widetilde{\mathbf{D}}_m^{\ell+1} \leftarrow P_{\mathcal{L}^+}(\max\{\widehat{\mathbf{D}}_m^{\ell+1} - \gamma \eta_\ell, 0\}), \quad (11)$$

where $P_{\mathcal{L}^+}(\cdot)$ corresponds 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 acts element-wise on $\mathbf{D}_m^{\ell+1}$. The updates (10) and (11) 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, which will be omitted for simplicity.

4.3 Estimating the question parameters

We next show how to estimate the question-dependent parameters $\mathbf{w}_i,\ \mu_i,\ \forall i$. 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 (A2) and (A3). In order to impose sparsity on $\mathbf{w}_i$, we add an ℓ_1-norm penalty to the cost function, which leads to the following optimization problem:

$$(\mathrm{P_w}) \qquad \underset{\mathbf{w}_i : w_{i,k} \geq 0, \forall k}{\text{minimize}} \sum_{(t,j) \in \mathcal{Q}^i} \mathbb{E}_{\mathbf{c}_j^{(t)}}$$

$$\left[-\log \Phi((2Y_j^{(t)} - 1)(\mathbf{w}_i^T \mathbf{c}_j^{(t)} - \mu_i)) \right] + \lambda \|\mathbf{w}_i\|_1 .$$

The problem $(\mathrm{P_w})$ is convex in $\mathbf{w}_i$, thanks to the fact that the negative log-likelihood of the observation likelihood is convex and the linearity of the expectation operator (see [18] for details). 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) [27] to approximate the cost function of $(\mathrm{P_w})$.

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 $\widehat{\mathbf{V}}_j^{(t)}$. The cost function in the optimization problem $(\mathrm{P_w})$ can now be approximated by a weighted average of the cost function evaluated at $\{(\widetilde{\mathbf{c}}_j^{(t)})_n\}$. Once again, $(\mathrm{P_w})$ can be solved efficiently by using the FISTA framework [1]. The gradient step is given by

$$\widehat{\mathbf{w}}_i^{\ell+1} \leftarrow \mathbf{w}_i^{\ell} - \eta_\ell \, \nabla f(\mathbf{w}_i). \tag{12}$$

The gradient $\nabla f(\mathbf{w}_i)$ is given by $\nabla f(\mathbf{w}_i) = -\widetilde{\mathbf{C}}_i \widetilde{\mathbf{r}}_i$, 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

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

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 projection step is given by:

$$\mathbf{w}_i^{\ell+1} \leftarrow \max\{\widehat{\mathbf{w}}_i^{\ell+1} - \lambda \eta_\ell, 0\}. \tag{13}$$

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 $\mathbf{w}_i$ as $[\mathbf{w}_i^T \ \mu_i]^T$; the corresponding latent learner concept knowledge state vectors $\mathbf{c}_j^{(t)}$ are augmented accordingly as $[(\mathbf{c}_j^{(t)})^T \ -1]^T$.

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

Performance metric	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	$\mathbf{0.8157 \pm 0.0028}$

5. EXPERIMENTAL RESULTS

We now demonstrate the efficacy of SPARFA-Trace using real-world educational datasets. We begin by comparing SPARFA-Trace against two established methods on predicting unobserved binary-valued learner response data, namely knowledge tracing (KT) [5, 22] and SPARFA [18]. Then, 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. The regularization parameters λ and γ are chosen via cross-validation [8], and all experiments are repeated for 25 independent Monte–Carlo trials.

5.1 Predicting responses for new learners

We now compare SPARFA-Trace against the KT method described in [22] for predicting responses for new learners that do not have previous recorded response history.

The dataset we use for this experiment is from an undergraduate computer engineering course collected using OpenStax Tutor (OST) [21]. We will refer to this dataset as "Dataset 1" in the following experiments. This dataset consists of the binary-valued graded response from 92 learners answering 203 questions, with 99.5% of the responses observed. The course is organized as 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 the learners' concept knowledge state transitions can only happen between two consecutive assignments/exams, due to their interaction with all the lectures/readings/exercises.

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 initialize the four parameters of KT (learner prior, learning probability, guessing probability, slipping probability) 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.

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 future predictions of $Y_j^{(t)}$ based on these estimates and observations $Y_j^{(1)}, \ldots, Y_j^{(t-1)}$, for $t = 1, \ldots, T$.

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

	SPARFA-M		**SPARFA-Trace**	
	Dataset 1	Dataset 2	Dataset 1	Dataset 2
Accuracy	$87.10 \pm 0.04\%$	$\mathbf{86.64 \pm 0.14\%}$	$\mathbf{87.31 \pm 0.05\%}$	$86.29 \pm 0.25\%$
Likelihood	0.727 ± 0.001	0.704 ± 0.002	$\mathbf{0.730 \pm 0.001}$	$\mathbf{0.707 \pm 0.003}$

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., $\frac{1}{|\Omega^c_{\text{obs}}|} \sum_{t,j:(t,j)\in\Omega^c_{\text{obs}}} p(Y_j^{(t)} | \mathbf{w}_{i_j^{(t)}}, \mathbf{c}_j^{(t)}, \mu_{i_j^{(t)}})$ where Ω^c_{obs} is the set of unobserved learner responses in the test set; the area under the ROC curve is a commonly-used performance metric for binary classifiers (see [22] for details).

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 slightly 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.

5.2 Predicting unobserved learner responses

We now compare SPARFA-Trace against the original SPARFA framework [18], which offers state-of-the-art collaborative filtering performance on predicting unobserved binary-valued graded learner responses.

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 5.1. The second dataset we use is from a signals and systems undergraduate course on OST, consisting 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 5(b). The full course consist of 14 assessments, including 12 assignments and 2 exams.

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 obtain estimates of all model parameters and then use these estimates to predict unobserved held-out responses in the test set.

The means and standard deviations of the prediction accuracy and prediction likelihood metrics covering multiple cross-validation trials are shown in Tables 1 and 2. We see that SPARFA-Trace achieves comparable or better performance than the static SPARFA-M on both datasets.

5.3 Visualizing time-varying LA and CA

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 parameters.

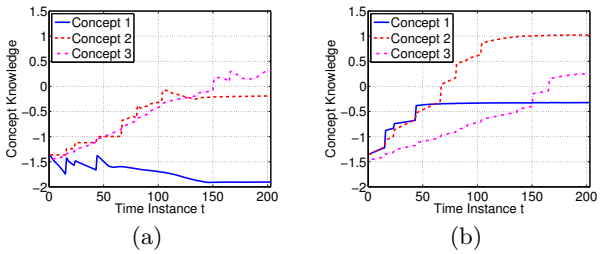


Figure 3: Estimated latent learner concept knowledge states for all time instances, for Dataset 1. (a) Learner 1's latent concept knowledge state evolution; (b) Average learner latent concept knowledge states evolution.

Figure 3(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. Hence, SPARFA-Trace can enable 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. Figure 3(b) shows the average learner concept knowledge states over the entire class at all time instances for Dataset 1. Using this information, SPARFA-Trace can also inform instructors on the trend of the concept knowledge state evolution of the entire class, in order to help them make timely adjustments to their course plans.

Figure 4(a) and Figure 4(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' 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. Dotted arrows represent unchanged learner concept knowledge states; these arrows correspond to zero entries in $\mathbf{D}_m$ and $\mathbf{d}_m$. Solid arrows represent the intrinsic knowledge gain of some concepts, characterized by large, positive entries in $\mathbf{d}_m$. 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 4(a), Learning Resource 3 is used in early stage of the course, and we can see that this learn-

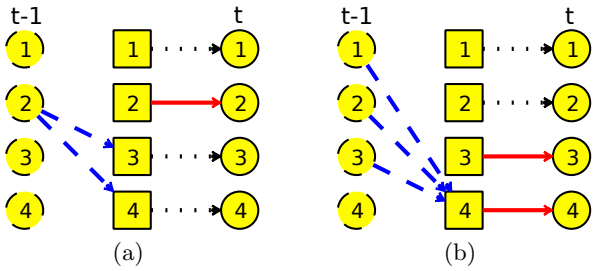


Figure 4: Visualized learner knowledge state transition effect of two distinct learning resources for Dataset 2. (a) Learner knowledge state transition effect for Learning Resource 3; (b) Learner knowledge state transition effect for Learning resource 9.

ing resource gives the learners a positive knowledge gain of Concept 2, while also helping on the more advanced Concepts 3 and 4. As shown in Figure 4(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.

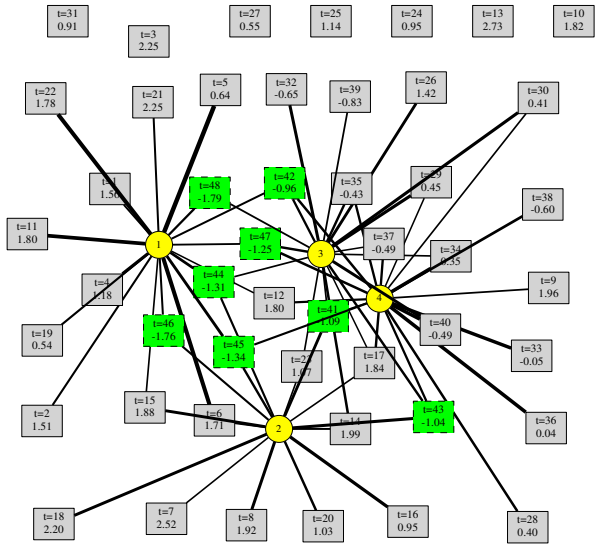
By analyzing the content organization of learning resources and their effects on learner concept knowledge state transitions, SPARFA-Trace enables a PLS to automatically recommend corresponding learning resources to learners based on their strengths and weaknesses. The estimated learning resource quality 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.

Figure 5 shows the question–concept association graph obtained from Dataset 2. Circle nodes represent concept nodes, while square, 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 5(a)).

6. CONCLUSIONS

We have proposed SPARFA-Trace, a novel blind approximate Kalman filtering approach for time-varying learning and content analytics. The proposed method jointly traces latent learner concept knowledge evolution over time 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. Being able to trace learners' concept knowledge evolution over time will enable a PLS to make timely feedback



(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 5: (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.

to learners on their strengths and weaknesses. Furthermore, the estimated content-dependent parameters provide rich information on the knowledge structure and quality of learning resources. Together with the question parameters estimated, a PLS would be able to operate in an *autonomous* 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 precludes manual intervention.

We note 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 [29], and the study of temporal effects in general collaborative filtering settings [16, 25, 26, 32].

Acknowledgments

Thanks to Kim Davenport and JP Slavinsky for providing the OpenStax Tutor (OST) data, and Andrew Waters and Ryan Ning for helpful discussions. Visit the website `www.sparfa.com`, where you can learn more about the SPARFA project and purchase t-shirts and other merchandise.

References

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

[2] C. M. Bishop and N. M. Nasrabadi. *Pattern Recognition and Machine Learning*. Springer New York, 2006.

[3] A. C. Butler, E. J. Marsh, J. P. Slavinsky, and R. G. Baraniuk. Integrating cognitive science and technology improve learning in a STEM classroom. *Educational Psychology Review*, 26(1), Feb. 2014.

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

[5] 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.

[6] A. Doucet, N. De Freitas, K. Murphy, and S. Russell. Rao-Blackwellised particle filtering for dynamic Bayesian networks. In *Proc.16th Conf. on Uncertainty in Artificial Intelligence*, pages 176–183, June 2000.

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

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

[9] S. S. Haykin. *Kalman Filtering and Neural Networks*. Wiley Online Library, 2001.

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

[11] S. J. Julier and J. K. Uhlmann. New extension of the Kalman filter to nonlinear systems. In *Proc. 11th Intl. Symposium on Aerospace/Defense Sensing, Simulation and Controls*, pages 182–193, Apr. 1997.

[12] R. E. Kalman. A new approach to linear filtering and prediction problems. *ASME J. of Basic Engineering*, 82(1):35–45, 1960.

[13] S. P. Kasiviswanathan, H. Wang, A. Banerjee, and P. Melville. Online ℓ_1-dictionary learning with application to novel document detection. *Advances in Neural Information Processing Systems*, pages 2267–2275, Dec. 2012.

[14] G. Kennedy, I. Ioannou, Y. Zhou, J. Bailey, and S. O'Leary. Mining interactions in immersive learning environments for real-time student feedback. *Australasian J. of Educational Technology*, 29(2), 2013.

[15] Knewton. Knewton adaptive learning: Building the world's most powerful recommendation engine for education. *online*, June 2012.

[16] Y. Koren and J. Sill. OrdRec: an ordinal model for predicting personalized item rating distributions. In *Proc. of the 5th ACM Conf. on Recommender Systems*, pages 117–124, Oct. 2011.

[17] 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.

[18] A. S. Lan, A. E. Waters, C. Studer, and R. G. Baraniuk. Sparse factor analysis for learning and content analytics. *J. of Machine Learning Research, to appear*, 2014.

[19] T. P. Minka. From hidden Markov models to linear dynamical systems. Technical report, MIT, 1999.

[20] T. P. Minka. Expectation propagation for approximate Bayesian inference. In *Proc. 17th Conf. on Uncertainty in Artificial Intelligence*, pages 362–369, Aug. 2001.

[21] OpenStaxTutor. https://openstaxtutor.org/, 2013.

[22] 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, June 2010.

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

[24] S. Roweis and Z. Ghahramani. Learning nonlinear dynamical systems using the expectation–maximization algorithm. *Kalman Filtering and Neural Networks*, 6:175–220, 2001.

[25] J. Silva and L. Carin. Active learning for online Bayesian matrix factorization. In *Proc. 18th Intl. Conf. on Knowledge Discovery and Data Mining*, pages 325–333, Aug. 2012.

[26] N. Thai-Nghe, T. Horvath, and L. Schmidt-Thieme. Factorization models for forecasting student performance. In *Proc. 4th Intl. Conf. on Educational Data Mining*, pages 11–20, July 2011.

[27] 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.

[28] C. Wang, J. Tang, J. Sun, and J. Han. Dynamic social influence analysis through time-dependent factor graphs. In *Intl. Conf. on Advances in Social Networks Analysis and Mining*, pages 239–246, July 2011.

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

[30] 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). *J. of Experimental Psychology*, 79(2):226, Feb. 1969.

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

[32] H. Yu et al. Feature engineering and classifier ensemble for KDD cup 2010. *JMLR Workshop and Conference Proceedings*, 2010.