Computational Education using Latent Structured Prediction

Tanja Käser ETH Zurich
Alexander G. Schwing ETH Zurich
Tamir Hazan University of Haifa
Markus Gross ETH Zurich

Abstract

Computational education offers an important add-on to conventional teaching. To provide optimal learning conditions, accurate representation of students' current skills and adaptation to newly acquired knowledge are essential. To obtain sufficient representational power we investigate suitability of general graphical models and discuss adaptation by learning parameters of a log-linear distribution. For interpretability we propose to constrain the parameter space a-priori by leveraging domain knowledge. We show the benefits of general graphical models and of regularizing the parameter space by evaluation of our models on data collected from a computational education software for children having difficulties in learning mathematics.

1 INTRODUCTION

Arithmetic skills are essential in modern society but many children experience difficulties in learning mathematics. Computer-based learning systems have the potential to offer an inexpensive extension to conventional education by providing a fear-free learning environment. To provide effective teaching, adaptation to the user's knowledge is essential. This is particularly important for students suffering from learning disabilities as the heterogeneity of these children requires a high grade of individualization.

A variety of methods are currently employed to model user knowledge and behavior. Markov Decision Processes are used for teaching planning (Brunskill and Russell, 2011; Rafferty et al., 2011) or diagnosing misconceptions (Rafferty et al., 2012). Logistic regression was proposed for modeling student learning (Rafferty and Yudelson, 2007; Yudelson and Brunskill, 2012; Rafferty et al., 2013). Furthermore, student knowledge and learning can be represented by Hidden Markov Models (HMM) (Piech et al., 2012), Bayesian networks (Brunskill, 2011; González-Brenes and Mostow, 2012b,a) or Bayesian Knowledge Tracing (BKT) (Corbett and Anderson, 1994). Bayesian Networks are also employed to model and predict students' learning styles (Kim et al., 2012), engagement states (Baschera et al., 2011; Käser et al., 2012) and goals (Conati et al., 2002). In cognitive sciences, bayesian networks are applied to model human learning (Kemp et al., 2010a; Frank and Tenenbaum, 2010; Kemp et al., 2010b) and understanding (Baker et al., 2005).

A crucial task when representing student knowledge with a probabilistic graphical model is recovery of a joint distribution over the model domain. Its goal is two-fold: The model should enable accurate prediction of student knowledge, while being interpretable. Addressing accuracy on the one hand, previous work has demonstrated the benefits of more complex student models (Lee and Brunskill, 2012; Yudelson et al., 2013). Obtaining interpretable parameters on the other hand has proven to be a challenge (Beck and Chang, 2007). This problem has been addressed by using brute-force grid search methods (Baker et al., 2010), imposing maxima on parameter values (Corbett and Anderson, 1994), contextual estimation of parameter values (Baker et al., 2008, 2010) or the analysis of expectation maximization (EM) convergence (Pardos and Heffernan, 2010b).

Our contributions are two-fold: first we introduce a framework to cope with more complex models, and second we show how to obtain interpretable results. Contrasting previous work on parameter learning with tree-structured models like HMMs we opt for more complex loopy parameterizations while noting that learning and inference is a challenge. We include a-priori domain expert knowledge via regularization with constraints to naturally enforce interpretability and show efficacy of our approach on data collected from a computer-based training program for learning mathematics (Käser et al., 2013).

Appearing in Proceedings of the 17^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2014, Reykjavik, Iceland. JMLR: W&CP volume 33.

2 BACKGROUND

Hidden Markov Models for Data Driven Education Traditionally, HMMs and BKT (Corbett and Anderson, 1994), a special case of HMM (Reye, 2004), are popular approaches for modeling student learning. The student knowledge is represented as a binary latent variable indicating whether the student masters the skill in question. Binary observations, *i.e.*, correct or wrong answers to questions, are used as surrogates to predict whether the student acquired the respective skill. Traditional BKT uses four model parameters: prior probability p_0 of knowing the skill, probability p_L of a skill transitioning from not known to known state, probability p_{slip} of making a mistake when applying a known skill and probability p_{guess} of correctly applying an unknown skill. Here, we also model the second transition probability, *i.e.*, the forget probability p_F, commonly assumed to be zero.

The learning task in BKT proceeds as follows: consider a skill $\mathcal{X}$, *e.g.*, subtraction, and the student's sequence of answers $\mathcal{Y}$ being a product space composed of multiple elements of the binary set $\{\text{correct}, \text{wrong}\}$. During learning we are interested in finding interpretable parameters $p = \{p_0, p_L, p_F, p_{slip}, p_{guess}\}$ that yield a "good" prediction of the student's answers $y \in \mathcal{Y}$ as well as information about a sequence of binary latent variables denoting whether the considered skill is mastered by the student. Learning in BKT was for example done using a brute-force grid search method (Baker et al., 2010) or expectation maximization (Pardos and Heffernan, 2010a; Wang and Heffernan, 2012).

Structured Learning for Data Driven Education We emphasize that the approach presented in the following permits to consider skills jointly within a single model. Hence we can leverage the correlations between different tasks such as addition and subtraction. To generalize the HMM models of a single skill to arbitrarily complex models, we describe the learning task as follows: consider an input space object $\mathcal{X}$, *i.e.*, a students' set of skills like subtraction, addition, multiplication *etc.*, and the corresponding task specific output space $\mathcal{Y}$, *i.e.*, a sequence of student answers. In addition we let $\mathcal{H}$ denote all the unobserved variables, *i.e.*, missing student answers and importantly the unobserved binary variables denoting whether a skill is mastered.

Further, let $\phi : \mathcal{X} \times \mathcal{Y} \times \mathcal{H} \to \mathbb{R}^F$ denote a mapping from the input, output and latent product space to the F-dimensional feature space. During learning we are interested in finding those weights $w \in \mathbb{R}^F$ that yield a "good" fit of the log-linear model distribution

$$p_w(\hat{y}, \hat{h}) \propto \exp w^\top \phi(x, \hat{y}, \hat{h}) \tag{1}$$

to a training set $\mathcal{D}$ of $|\mathcal{D}|$ input- and output-space object pairs $(x, y) \in \mathcal{X} \times \mathcal{Y}$, *i.e.*, $\mathcal{D} = \{(x_i, y_i)_{i=1}^{|\mathcal{D}|}\}$.

Performing maximum likelihood we choose the weights w such that the model assigns highest probability to the data set $\mathcal{D}$. Since the unobserved variable space does not reveal any information we marginalize it out and let $p(\hat{y} \mid x, w) \propto \sum_{\hat{h} \in \mathcal{H}} \exp w^\top \phi(x, \hat{y}, \hat{h})$. If the data is independent and identically distributed (i.i.d.), minimization of the negative log-likelihood $-\ln[p(w) \prod_{(x,y)} p(y \mid x, w)]$ yields the following optimization

$$\min_w \frac{C}{2}\|w\|_2^2 - \sum_{(x,y)\in\mathcal{D}} \ln p(y \mid x, w),$$

with a log-quadratic prior function $p(w)$.

Features To specify the feature vector ϕ we relate traditional HMMs used in computational education to the aforementioned log-linear models. Consider the transition probability p_L from a skill which is not mastered to being known by the student. This parameter is part of a conditional probability table $p(h_{r+1} \mid h_r)$ which is specified by the two probabilities p_L and the forget probability p_F. We note that $p_L = p(h_{r+1} = 1 \mid h_r = 0) = 1 - p(h_{r+1} = 0 \mid h_r = 0)$ and let $p(h_{r+1} \mid h_r = 0) \propto \exp w_{r,0}(1 - 2h_{r+1})$. We therefore obtain the feature function $\phi(h_{r+1}) = 1 - 2h_{r+1}$. We proceed similarly for the emission probabilities and therefore obtain the joint distribution as a product of the exponential terms which translates to a weighted linear combination of feature vector entries in the exponent, *i.e.*, the model given in Eq. (1). We detail the feature vector for our more complex models in Sec. 4.

Optimization Considering optimization of the aforementioned non-convex cost function we commonly follow the expectation maximization (EM) approach or more generally the concave convex procedure (CCCP). We linearize the concave term by computing its gradient at the current iterate and subsequently minimize a convex objective. This step, identical to optimizing HMMs via EM, is guaranteed to converge to a stationary point.

But contrasting the HMMs, neither linearization of the concave part nor minimization of the resulting convex objective is computationally tractable for general models. To our benefit and as indicated before and detailed below, the elements of the feature vector $\phi(x, y, h)$ typically decompose into functions depending only on a small fraction of variables. This can be employed to approximate the objective. Recently, Schwing et al. (2012) showed that a convex approximation admits more efficient learning of parameters than its non-convex counterpart.

Note that interpretability of the parameters w is not guaranteed, particularly since guarantees exist for only converging to a local optimum. However, interpretability implies some form of expectation regarding the parameters. In the following, we therefore propose to constrain the parameter space. This is useful since domain experts are capable of restricting the range of acceptable parameters, *e.g.*, it is reasonable to assume the guess probability p_{guess} to be less than 0.3.

3 LEARNING WITH CONSTRAINED PARAMETERS

Let $\bar{\ell}(x, y, w) = -\ln p(y \mid x, w)$, *i.e.*, explicitly,

$$\bar{\ell}(x,y,w) = \ln \sum_{\hat{y},\hat{h}} \exp \hat{\phi}(x,\hat{y},\hat{h},w) - \ln \sum_{\hat{h}\in\mathcal{H}} \exp \hat{\phi}(x,y,\hat{h},w)$$

while the potential is given as $\hat{\phi}(x, y, h, w) = w^\top \phi(x, y, h)$. Then we augment the learning task to read as the *constrained* optimization problem

$$\min_w \frac{C}{2}\|w\|_2^2 + \sum_{(x,y)\in\mathcal{D}} \bar{\ell}(x,y,w) \quad \text{s.t.} \quad w \in \mathcal{C}, \qquad (2)$$

with $\mathcal{C}$ denoting a convex set. Leaving the constraint set aside, this program possesses the same difficulty as the original task, *i.e.*, we minimize a non-convex objective operating on exponentially sized sets. Being interested in the quality of duality based approximations we subsequently follow Schwing et al. (2012).

We first note that an upper-bound to the program given in Eq. (2) is stated by the following cost function:

$$\frac{C}{2}\|w\|_2^2 + \sum_{(x,y)} \left(\ln \sum_{\hat{y},\hat{h}} \exp\left(\hat{\phi}(x,\hat{y},\hat{h},w)\right) - \right.$$
$$\left. - H(q_{(x,y)}) - \mathbb{E}_{q_{(x,y)}}[\hat{\phi}(x,y,\hat{h},w)] \right),$$

with H denoting the entropy and $\mathbb{E}$ indicating computation of the expectation. Importantly, the upper bound allows to divide the program into two parts which are iterated alternatingly when following the CCCP procedure: on the one hand a minimization w.r.t. the distribution $q_{(x,y)}$ ranging over the latent space $\hat{h} \in \mathcal{H}$ for every sample (x, y), often referred to as 'latent variable prediction task.' On the other hand a minimization w.r.t. the weight vector w subject to constraints $\mathcal{C}$. Both problems remain intractable without further modifications. However we notice that minimization to find the distributions $q_{(x,y)}$ directly follows (Schwing et al., 2012) and we can incorporate their approximation without further modification.

Due to the additional constraint set it is the second task which requires specific attention. The relevant excerpt from the program given in Eq. (2) reads as follows:

$$\min_{w\in\mathcal{C}} \sum_{(x,y)\in\mathcal{D}} \ln \sum_{\hat{y},\hat{h}} \exp w^\top \phi(x,\hat{y},\hat{h}) - w^\top d + \frac{C}{2}\|w\|_2^2. \qquad (3)$$

We note that the vector of empirical means $d \in \mathbb{R}^F$ contains information from the observed variables as well as information from the linearization of the concave part. This task differs from the standard structured prediction program in an additional regularization w.r.t. the constraint set $\mathcal{C}$. Although assumed to be convex subsequently, this additional regularization makes the program more challenging to solve in general. We subsequently show the approximations required to obtain an efficient algorithm based on projected gradients. To this end we first state the dual program of the task given in Eq. (3).

Claim 1 *The dual program of the constrained structured prediction task (Eq.* (3)*) reads as*

$$\max_{p_{(x,y)}\in\Delta} \sum_{(x,y)\in\mathcal{D}} H(p_{(x,y)}(\hat{y})) + \frac{C}{2}\left\|P_{\mathcal{C}}\left[z\right]\right\|_2^2 - Cz^\top P_{\mathcal{C}}\left[z\right],$$

where we maximize the entropy H of distributions $p_{(x,y)}$ restricted to the probability simplex $\Delta_{\mathcal{Y}\times\mathcal{H}}$ over the complete data space. The projection of $z = \frac{1}{C}\left(d - \sum_{(x,y),\hat{y},\hat{h}} p_{(x,y)}(\hat{y},\hat{h})\phi(x,\hat{y},\hat{h})\right)$ onto the constraint set $\mathcal{C}$ is denoted by $P_{\mathcal{C}}\left[z\right]$ and $d \in \mathbb{R}^F$ refers to the vector of empirical means.

Proof: To prove this claim, we introduce a temporary variable $g(x, \hat{y}, \hat{h}) = w^\top \phi(x, \hat{y}, \hat{h})$ to decouple the soft-max function from the norm minimization in Eq. (3). Optimizing w.r.t. both, w and g, we obtain the entropy as the conjugate dual of the soft-max. Minimizing the norm subject to constraints yields the projection of the difference between the empirical means vector d and its estimate onto the constraint set $\mathcal{C}$. We note that $\mathcal{C} = \mathbb{R}^F$ yields the solution given by Hazan and Urtasun (2010), which concludes the proof. □

The aforementioned summation over exponentially sized sets within the primal problem manifests itself in distributions $p_{(x,y)}$ over respective simplexes $\Delta_{\mathcal{Y}\times\mathcal{H}}$. Instead of working with a full joint distribution over the set of all possible solutions $\mathcal{Y}\times\mathcal{H}$, we operate with corresponding marginals $b_{(x,y)}$ for sample (x, y) and respective marginalization constraints. The marginals are chosen according to the variable dependence structure introduced within the feature vector $\phi(x, \hat{y}, \hat{h})$.

More formally, let the k-th element of the feature vector be given by $\phi_k(x, y, h) = \sum_{r\in\mathcal{R}_k} \phi_{k,r}(x, (y, h)_r)$ where r specifies a restriction of the function to a subset of the observed and unobserved variables.

Algorithm 1 (Structured Prediction with Constrained Parameter Spaces) *Let* $\tilde{\phi}_{(x,y),r}((\hat{y},\hat{h})_r) = \sum_{k:r\in\mathcal{R}_k} w_k \phi_{k,r}(x,(\hat{y},\hat{h})_r)$.
Repeat until convergence:

1. *Update Lagrange multipliers:* $\forall (x,y), r, p \in P(r), (y,h)_r$

$$\mu_{(x,y),p\to r}((y,h)_r) = \ln \sum_{(y,h)_p \setminus (y,h)_r} \exp\left(\tilde{\phi}_{(x,y),r}((y,h)_r) - \sum_{p'\in P(p)} \lambda_{(x,y),p\to p'}((y,h)_{p'}) + \sum_{r'\in C(p)\setminus r} \lambda_{(x,y),r'\to p}((y,h)_{r'})\right)$$

$$\lambda_{(x,y),r\to p}((y,h)_r) \propto \frac{1}{1+|P(r)|}\left(\hat{\phi}_{(x,y),r}((y,h)_r) + \sum_{p'\in P(r)} \mu_{(x,y),p'\to r}((y,h)_r)\right) - \mu_{(x,y),p\to r}((y,h)_r)$$

2. *Perform a gradient step and project the result onto the constraint set* $\mathcal{C}$*:*

$$w \leftarrow P_{\mathcal{C}}[w - \gamma \nabla_w f(\lambda, w)]$$

Figure 1: An algorithm for learning parameters of structured models within constrained parameter spaces.

The set of all restrictions for the k-th element of the feature vector is referred to via $\mathcal{R}_k$. All in all we therefore consider the marginals $b_{(x,y),r}((y,h)_r)$ which are required to fulfill the marginalization constraints, *i.e.*, we enforce them to be consistent amongst each other. Importantly, this means that we neglect the exponential number of constraints within the marginal polytope by adopting its local approximation (Wainwright and Jordan, 2008). In addition to usage of marginals we approximate the joint entropy $H(p_{(x,y)}) \approx \sum_r H(b_{(x,y),r})$.

To obtain an approximated convex primal we introduce Lagrange multipliers $\lambda_{(x,y),r\to p}((y,h)_r)$ for each marginalization constraint that ties together two restrictions r and p. We obtain the approximated, convex and constrained primal as follows:

$$\min_{w\in\mathcal{C},\lambda} \sum_{(x,y),r} \ln \sum_{(\hat{y},\hat{h})_r} \exp \hat{\phi}_{(x,y),r}((\hat{y},\hat{h})_r) - d^\top w + \frac{C}{2}\|w\|_2^2, \tag{4}$$

where we denote the re-parameterized potential via

$$\hat{\phi}_{(x,y),r}((\hat{y},\hat{h})_r) = \sum_{k:r\in\mathcal{R}_k} w_k \phi_{k,r}(x,(\hat{y},\hat{h})_r) + \sum_{p\in P(r)} \lambda_{r\to p}((\hat{y},\hat{h})_r) - \sum_{c\in C(r)} \lambda_{c\to r}((\hat{y},\hat{h})_c).$$

The derivation follows (Hazan and Urtasun, 2010; Schwing et al., 2012) and we recover the constraint set $\mathcal{C}$ by computing the dual for the projection $P_{\mathcal{C}}$. Intuitively we push energy λ between different restrictions such that we can find a weight vector w which minimizes the objective subject to $\mathcal{C}$.

This formulation differs from Hazan and Urtasun (2010) in that the domain for the parameters w is constrained by the convex set $\mathcal{C}$. We proceed by iterating between updates for the Lagrange multipliers λ and the model parameters w which guarantees convergence for the convex cost function. Note that optimization of the program given in Eq. (4) w.r.t. λ is unconstrained. Therefore we follow a block-coordinate descent scheme.

Let $f(w,\lambda)$ denote the cost function of the program given in Eq. (4). Fixing λ, f is a smooth, convex but non-linear function in w and a well-known method to address the constraint minimization of f w.r.t. w is the projected gradient algorithm (Rockafellar, 1970). We use the gradient of the smooth cost-function as a descent direction, perform a step and project the result onto the constraint set $\mathcal{C}$.

It is important to note that a single projection step is sufficient for convergence guarantees since block-coordinate descent methods only require to decrease the cost function at every iteration which is ensured after a single projection. We summarize this observation in the following claim.

Claim 2 *The algorithm outlined in Fig. 1 guarantees convergence of the constrained structured prediction program given in Eq.* (4).

Proof: Strong convexity admits block-coordinate descent updates (Tseng, 1993), *i.e.*, iterating between updates for weights w and Lagrange multipliers λ. The requirement of decreasing the cost function is met for the updates w.r.t. λ and also ensured by a single projection of w onto $\mathcal{C}$, which consequently proves the claim. □

Combining the structured prediction algorithm outlined in Fig. 1 with the 'latent variable prediction task' we obtain the algorithm given in Fig. 2 which we refer to as *constrained structured prediction with latent variables*.

Constrained structured prediction with latent variables
Repeat until convergence:

1. Solve the approximate 'latent variable prediction' until convergence and update the empirical means d.
2. Perform a single iteration of 'constrained structured prediction' as detailed in Fig. 1.

Figure 2: Algorithm for constrained structured prediction with latent variables.

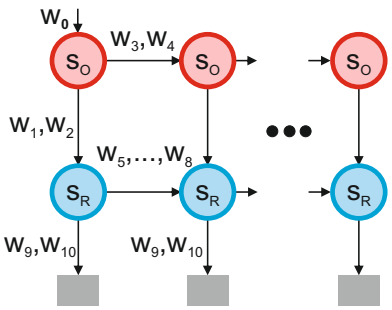

Figure 3: Structure of the graphical model used for the number understanding experiment.

4 EXPERIMENTAL EVALUATION

We evaluate our approach in three real data experiments. We compare the prediction accuracy of our learning method with and without constraints to previous work employing parameters w chosen by experts (Käser et al., 2012) and to work applying HMMs (Yudelson et al., 2013).

Data We utilize input logs from the computer-based adaptive training environment Calcularis (Käser et al., 2013) which is a software for children with developmental dyscalculia (von Aster and Shalev, 2007). Children are taught different number representations as well as arithmetic operations. The student model used in Calcularis consists of 100 different mathematical skills allowing adaptation to the child's difficulty level (Käser et al., 2012). To improve the kids learning efficiency, an accurate prediction of the modeled mathematical knowledge is essential. This is particularly important for children with learning disabilities, as their heterogeneity requires a high grade of individualization. Moreover, performance prediction in such applications is not an easy task as available data is noisy and sparse, leading to many latent variables.

The data was collected in a multi-center user study in Germany and Switzerland. From the 126 participating children (87 females, 39 males), 57 were diagnosed as having developmental dyscalculia (assessed by standardized tests at the beginning of the study), and 69 were control children (also assessed by standardized tests). The children attended 2^{nd} to 5^{th} grade of elementary school (age of diagnosed children: 8.61 ± 0.86; age of control children: 8.75 ± 0.85; $p = 0.36$) and were German speaking. The input logs from the 126 participants form our data samples.

Experimental conditions The prediction accuracy is computed as follows: given a set of observations for the Bayesian network, we predict the state of the unobserved nodes and provide the root mean squared error (RMSE), the L1 norm (L1) and the L2 norm (L2) between the true state and the predicted probability for that state. Furthermore, we also compute the classification error (CE, *i.e.*, frequency of predicted state not equaling true state) and the area under the ROC curve (AUC). When working with probabilities, appropriate calibration is essential (Gneiting et al., 2007). We assess the calibration of our models by building reliability diagrams and computing the average calibration distance (CA) between true and predicted outcome per bin. If not noted otherwise, convex learning stops when the improvement of the primal is less than 10^{-9} or the maximum number of iterations exceeds 500. In case of constraints the stopping criterion is met if the primal improves by less than $5 \cdot 10^{-6}$ or 300 iterations are exceeded. For inference, we limit the number of message passing iterations to 100.

Features The feature vector ϕ for the Bayesian network models are specified following Sec. 2. Consider the conditional probability table (CPT) describing the relationship between two skills h_a and h_b. The entries of this table are defined using two parameters p_1 and p_2, setting $p_1 = p(h_b = 1 \mid h_a = 0) = 1 - p(h_b = 0 \mid h_a = 0)$ and $p_2 = p(h_b = 1 \mid h_a = 1) = 1 - p(h_b = 1 \mid h_a = 0)$. As shown in Sec. 2, let $p(h_b \mid h_a = 0) \propto \exp w_{b,0}(1 - 2h_b)$ and $p(h_b \mid h_a = 1) \propto \exp w_{b,1}(1 - 2h_b)$, which leads to the feature function $\phi(h_b) = 1 - 2h_b$. When using a ternary conditional probability distribution including three skills h_a, h_b and h_c, we obtain four degrees of freedom: We define $p_1 = p(h_c = 1 \mid h_a = 0, h_b = 0) = 1 - p(h_c = 0 \mid h_a = 0, h_b = 0) \propto \exp w_{c,0,0}(1 - 2h_c)$ and similarly specify p_2, p_3 and p_4. Generally, we need 2^{n-1} parameters to specify a CPT including n skills.

Number understanding In a first experiment, we look at two skills taught in the number range from 0 to 100. Fig. 3 illustrates the model, where nodes colored in red represent knowledge of the concept of ordinal number understanding (s_O), *i.e.*, understanding a number as a position in a sequence. There exists no exercise for this skill, hence no observations are available. The concept of relative number understanding (s_R) is denoted by the variables highlighted in blue. Relative number understanding denotes the ability to understand a number as a difference between two numbers.

	Expert	HMM	$\mathcal{C} = \emptyset$	$\mathcal{C} = \mathcal{C}_1$	$\mathcal{C} = \mathcal{C}_2$	$\mathcal{C} = \mathcal{C}_3$	$\mathcal{C} = \mathcal{C}_4$
RMSE	0.464	0.388	0.393	0.382	0.379	0.374	**0.373**
L1	0.440	0.308	**0.265**	0.299	0.281	0.283	0.284
L2	0.227	0.170	0.184	0.163	0.163	**0.154**	**0.154**
CE	0.346	0.213	0.213	0.213	0.213	**0.202**	**0.202**
AUC	**0.625**	0.500	0.500	0.606	0.591	0.615	0.615
CA	0.432	0.073	0.135	0.072	0.084	**0.052**	0.054

Table 1: Different error measures for number understanding. Comparison of unconstrained and constrained conditions to previous work using a domain expert (Käser et al., 2012) or an HMM (Yudelson et al., 2013).

We cannot directly observe this ability, but the results of an exercise associated with it. These results are referred to by rectangles which denote the outcome of a particular "task". Every column in Fig. 3 represents a time-step and the depicted graphical model is an extract of the model used in the Calcularis software. For this experiment, we used a maximum of 50 time-steps (task outcomes) per child (mean: 22.16 ± 9.98). One child with no observations at skill s_R was excluded from the analysis.

The model representing this task employs $F = 11$ parameters w to specify the conditional probabilities that define the network illustrated in Fig. 3. Subsequently, we describe the domain knowledge introduced to constrain the parameters and the meaning of the elements of the weight vector w. Parameters w_9 and w_{10} are associated with the so called 'guess' (probability of getting a task right without knowing the associated skill) and 'slip' (getting the task wrong despite having the associated skill) probabilities which are commonly assumed to be lower than 0.3 (Corbett and Anderson, 1994). This upper bound translates to the constraints $w_9 \geq 0.4236$ and $w_{10} \leq -0.4236$. Furthermore, parameters w_5 and w_8 are associated with learning (probability that a skill is learnt from one time step to the other) and forgetting (probability of forgetting a previously learnt skill). We limit these probabilities to be lower than 0.3, yielding $w_5 \geq 0.4236$ and $w_8 \leq -0.4236$. The aforementioned constraints define the set $\mathcal{C}_1$.

We refer to set $\mathcal{C}_2$ as the constraints within the set $\mathcal{C}_1$, augmented by the following restrictions. Since w_3 and w_4 are also related to learning and forgetting, we utilize constraints identical to those for w_5 and w_8: $w_3 \geq 0.4236$ and $w_4 \leq -0.4236$. Similarly, we define $w_6 \geq 0.4236$ and $w_7 \leq -0.4236$. In addition, the hierarchical skill model of Calcularis assumes that the number understanding ability s_O is a prerequisite for relative number understanding s_R (von Aster and Shalev, 2007). Hence we restrict w_1 and w_2 by assuming that the probability of knowing s_R given s_O is larger than 0.7, while we let the probability of knowing s_R despite not knowing s_O be smaller than 0.3, which yields $w_1 \geq 0.4236$ and $w_2 \leq -0.4236$. Configurations $\mathcal{C}_3$ and $\mathcal{C}_4$ constrain the same parameters as $\mathcal{C}_1$ and $\mathcal{C}_2$, but are more restrictive by replacing 0.3 and 0.7 with 0.2 and 0.8.

After learning the model parameters using only the observed data, prediction is performed as follows: we assume "Task 1" to be given and predict the outcome of "Task 2". Afterwards we employ results from both "Task 1" and "Task 2" to predict the outcome of "Task 3" and continue to predict "Task k," $k \in \{4, 5, \ldots, 50\}$ assuming the preceding task outcomes to be given.

The performance results provided in Tab. 1 are computed using 10-fold cross validation. The most accurate results (per error measure) are marked in bold. We observe our constrained learning approach to outperform previous methods for most error metrics. Also the unconstrained optimization $\mathcal{C} = \emptyset$ yields good prediction results with the following parameter values: $w_1, ..., w_8$ are set to 0, which results in uniform distributions for the according CPTs. The parameters w_9 and w_{10} are set to values smaller than -1 (over all folds). The model therefore predicts a correct outcome with a probability higher than 0.88, independent of previous observations and the state of the hidden nodes. As the investigated skill was easy to solve for most children, this model exhibits a high prediction accuracy. It is, however, not interpretable with respect to human learning. The constrained models are generally well calibration. Note that the expert parameters generally have worse calibration. This result is not unexpected, as the expert parameters are not fit to the data.

Number representation In the second experiment, we investigate number representation tasks in the number range from 0 to 100. The graphical model is again an extract from the student model of Calcularis and is illustrated in Fig. 4. Nodes colored in green represent knowledge of the Arabic notation system (s_A). There exists no exercise for this skill, hence no observations are available. The ability to assign a number to an interval (s_S) is denoted by red circles. The task associated with this skill is to guess a num-

	Expert	HMM	$\mathcal{C} = \emptyset$	$\mathcal{C} = \mathcal{C}_1$	$\mathcal{C} = \mathcal{C}_2$	$\mathcal{C} = \mathcal{C}_3$	$\mathcal{C} = \mathcal{C}_4$
RMSE	0.541	0.456	0.474	0.445	0.448	**0.439**	**0.439**
L1	0.513	0.418	0.464	0.425	0.431	0.412	**0.410**
L2	0.300	0.215	0.227	0.201	0.202	**0.195**	**0.195**
CE	0.533	0.327	0.324	0.298	**0.274**	0.281	0.280
AUC	0.532	0.559	0.445	0.676	0.691	**0.705**	0.701
CA	0.232	0.101	0.092	0.086	0.086	0.077	**0.051**

Table 2: Different error measures for number representation. Comparison of different configurations to previous work using a domain expert (Käser et al., 2012) or an HMM (Yudelson et al., 2013).

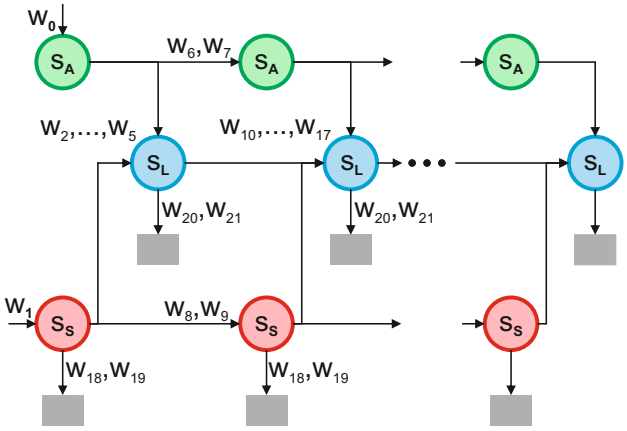

Figure 4: Structure of the graphical model used for the number representation experiment.

ber in as few steps as possible. After each guess, the child is told if the searched number is bigger or smaller than the guessed number. The ability to indicate the position of a number on a number line (s_L) is denoted by the variables highlighted in blue. This ability is a very important step in developing the mental number line representation, which is essential for number processing (von Aster and Shalev, 2007). Again, we used a maximum of 50 time-steps (task outcomes) per child (mean: 21.40 ± 10.63). For the second model, we employ $F = 22$ parameters which specify the conditional probabilities of the graphical model displayed in Fig. 4.

To constrain the parameters, we make use of the domain knowledge introduced in the first experiment. To obtain the first constrained configuration $\mathcal{C}_1$, we introduce the following constraints: $w_i \geq 0.4236$ if $i \in \{6, 8, 10, 18, 20\}$ and $w_i \leq -0.4236$ if $i \in \{7, 9, 17, 19, 21\}$. For the second configuration ($\mathcal{C}_2$), we augment the constraints of set $\mathcal{C}_1$ with the new constraints $w_i \geq 0.4236$ $\forall i \in \{2, 3, 4, 11, 12, 13\}$ and $w_i \leq -0.4236$ $\forall i \in \{5, 14, 15, 16\}$. Again, configurations $\mathcal{C}_3$ and $\mathcal{C}_4$ constrain the same parameters as $\mathcal{C}_1$ and $\mathcal{C}_2$, but are more restrictive by replacing 0.4236 and -0.4236 with 0.6913 and -0.6913.

Prediction is done successively for each time-step as described in the first experiment and the performance results given in Tab. 2 are again computed using 10-fold cross validation. The proposed constraints lead to improvement of all error measurements. We particularly highlight the decrease of the classification error by 5.3% ($CE_{C_2} = 0.274$) compared to previous work ($CE_{C_{HMM}} = 0.327$) (Yudelson et al., 2013). Furthermore, improvements in calibration are large ($CA_{HMM} = 0.101$, $CE_{C_4} = 0.051$).

Subtraction The four skills investigated in this experiment are different subtraction skills in the number range from 0 to 100. The graphical model, which is again an extract of the student model used in Calcularis, is illustrated in Fig. 5. The nodes highlighted with green color (s_1) in Fig. 5 denote a subtraction task without borrowing and a single-digit number as the subtrahend (*e.g.*, $35 - 2 =?$) while nodes in blue color (s_3) also represent a subtraction task without borrowing, but with a two-digit subtrahend (*e.g.*, $35 - 12 =?$). Purple nodes (s_2) denote subtraction with borrowing and a single-digit subtrahend (*e.g.*, $35 - 6 =?$) and the red nodes (s_4) denote the ability to do subtraction with borrowing and two two-digit numbers (*e.g.*, $35 - 6 =?$). The rectangles denote results of an exercise associated with the skills s_2, s_3 and s_4. Again, we used a maximum of 50 time-steps (task outcomes) per child (mean: 43.59 ± 10.47). To specify the conditional probabilities of the graphical model (Fig. 5), we employ $F = 33$ parameters.

The constrained configurations for this experiment follow the domain knowledge introduced in the first two experiments. More specifically, $\mathcal{C}_1$ denotes the following constraints: $w_i \geq 0.4236$ $\forall i \in \{9, 11, 15, 19, 27, 29, 31\}$ while $w_i \leq -0.4236$ $\forall i \in \{10, 14, 18, 26, 28, 30, 32\}$. The second configuration $\mathcal{C}_2$ augments the set $\mathcal{C}_1$ by adding $w_i \geq 0.4236$ $\forall i \in \{1, 3, 5, 6, 7, 12, 16, 20, 21, 22\}$ and $w_i \leq -0.4236$ $\forall i \in \{2, 4, 8, 13, 17, 23, 24, 25\}$. Again, configurations $\mathcal{C}_3$ and $\mathcal{C}_4$ constrain the same parameters as $\mathcal{C}_1$ and $\mathcal{C}_2$, but are more restrictive by replacing 0.4236 and -0.4236 with 0.6913 and -0.6913.

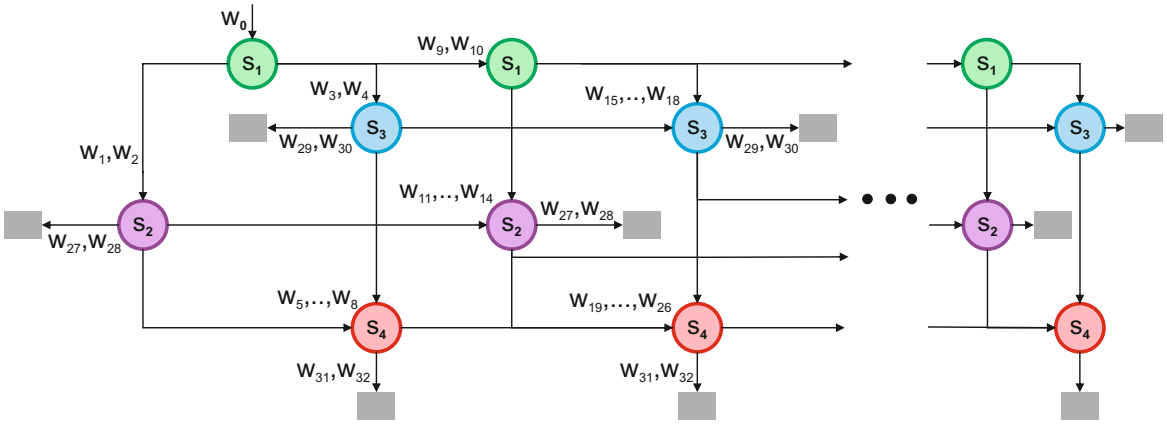

Figure 5: Structure of the graphical model used for the subtraction experiment.

	Expert	**HMM**	$\mathcal{C} = \emptyset$	$\mathcal{C} = \mathcal{C}_1$	$\mathcal{C} = \mathcal{C}_2$	$\mathcal{C} = \mathcal{C}_3$	$\mathcal{C} = \mathcal{C}_4$
RMSE	0.489	0.467	0.469	0.453	0.436	0.446	**0.433**
L1	0.460	0.422	0.445	**0.390**	**0.390**	0.391	0.392
L2	0.248	0.224	0.224	0.214	0.195	0.205	**0.192**
CE	0.398	0.327	0.325	0.313	0.287	0.302	**0.268**
AUC	0.555	0.511	0.561	0.641	0.674	0.621	**0.682**
CA	0.110	0.035	0.047	0.098	0.035	0.070	**0.029**

Table 3: Different error measures for subtraction. Comparison of different configurations to previous work using a domain expert (Käser et al., 2012) or an HMM (Yudelson et al., 2013).

Prediction is done as described in the first experiment and the performance results provided in Tab. 3 are again computed using 10-fold cross validation. We observe again significant improvements in all error measurements. We highlight the improvement of the classification error by 5.9% when learning our computational education model within a constrained parameter space ($CE_{HMM} = 0.327$, $CE_{C_4} = 0.268$). The constrained models are again well calibrated.

5 DISCUSSION

Our results demonstrate that introducing domain knowledge in the form of parameter constraints has a two-fold benefit. On one hand, the introduced parameter constraints guarantee an interpretable model. On the other hand, the proposed restrictions lead to improvement of the error metrics. Introducing restrictions on the parameter space is particularly beneficial for more complex models as well as for more difficult skills. For difficult skills where children change from the unlearnt to the learnt state after some training time, the unconstrained optimization converges to a solution closed to a uniform distribution (of correct and wrong outcomes), while the introduced domain knowledge enables more precise modeling of learning.

Compared to the HMMs used in previous work (Pardos and Heffernan, 2010a; Wang and Heffernan, 2012; Wang and Beck, 2013; Yudelson et al., 2013) our Bayesian network models are able to specify the dependencies between the different skills allowing a more precise description of the learning domain. Although learning and inference becomes more complex when introducing loopy parameterizations, our constraint optimization outperforms previous work (Yudelson et al., 2013) on prediction accuracy and also improve calibration.

6 CONCLUSION

We showed that constraining the parameter space of convex structured prediction methods maintains the design of efficient algorithms, *i.e.*, interleaving of message passing updates and parameter updates. Additionally we illustrated on real data from a study of children having dyscalculia that prediction performance and calibration improve when learning within the constrained parameter space. Furthermore, our algorithm guarantees interpretable models. In the future we plan to increase the size of our models to capture more statistics of its users.

References

Baker, C., Tenenbaum, J. B., and Saxe, R. (2005). Bayesian models of human action understanding. In *Proc. NIPS*.

Baker, R. S., Corbett, A. T., and Aleven, V. (2008). More Accurate Student Modeling through Contextual Estimation of Slip and Guess Probabilities in Bayesian Knowledge Tracing. In *Proc. ITS*.

Baker, R. S., Corbett, A. T., Gowda, S. M., Wagner, A. Z., MacLaren, B. A., Kauffman, L. R., Mitchell, A. P., and Giguere, S. (2010). Contextual Slip and Prediction of Student Performance after Use of an Intelligent Tutor. In *Proc. UMAP*.

Baschera, G.-M., Busetto, A. G., Klingler, S., Buhmann, J., and Gross, M. (2011). Modeling Engagement Dynamics in Spelling Learning. In *Proc. AIED*.

Beck, J. E. and Chang, K. M. (2007). Identifiability: A Fundamental Problem of Student Modeling. In *Proc. UM*.

Brunskill, E. (2011). Estimating Prerequisite Structure From Noisy Data. In *Proc. EDM*.

Brunskill, E. and Russell, S. (2011). Partially Observable Sequential Decision Making for Problem Selection in an Intelligent Tutoring System. In *Proc. EDM*.

Conati, C., Gertner, A., and VanLehn, K. (2002). Using bayesian networks to manage uncertainty in student modeling. *UMUAI*.

Corbett, A. T. and Anderson, J. R. (1994). Knowledge tracing: Modeling the acquisition of procedural knowledge. *UMUAI*.

Frank, M. C. and Tenenbaum, J. B. (2010). Three ideal observer models for rule learning in simple languages. *Cognition*.

Gneiting, T., Balabdaoui, F., and Raftery, A. E. (2007). Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society Series B: Statistical Methodology*.

González-Brenes, J. P. and Mostow, J. (2012a). Dynamic Cognitive Tracing: Towards Unified Discovery of Student and Cognitive Models. In *Proc. EDM*.

González-Brenes, J. P. and Mostow, J. (2012b). Topical Hidden Markov Models for Skill Discovery in Tutorial Data. *NIPS - Workshop on Personalizing Education With Machine Learning*.

Hazan, T. and Urtasun, R. (2010). A Primal-Dual Message-Passing Algorithm for Approximated Large Scale Structured Prediction. In *Proc. NIPS*.

Käser, T., Baschera, G.-M., Busetto, A. G., Klingler, S., Solenthaler, B., Buhmann, J. M., and Gross, M. (2012). Towards a Framework for Modelling Engagement Dynamics in Multiple Learning Domains. *IJAIED: Best of AIED 2011 - Part 2*.

Käser, T., Baschera, G.-M., Kohn, J., Kucian, K., Richtmann, V., Grond, U., Gross, M., and von Aster, M. (2013). Design and evaluation of the computer-based training program Calcularis for enhancing numerical cognition. *Frontiers in Developmental Psychology*.

Käser, T., Busetto, A. G., Baschera, G.-M., Kohn, J., Kucian, K., von Aster, M., and Gross, M. (2012). Modelling and optimizing the process of learning mathematics. In *Proc. ITS*.

Kemp, C., Goodman, N. D., and Tenenbaum, J. B. (2010a). Learning to Learn Causal Models. *Cognitive Science*.

Kemp, C., Tenenbaum, J. B., Niyogi, S., and Griffiths, T. L. (2010b). A probabilistic model of theory formation. *Cognition*.

Kim, E.-S., Noh, Y.-K., and Zhang, B.-T. (2012). Learning-style recognition from eye-hand movement using a dynamic Bayesian network. *NIPS Workshop on Personalizing Education on Machine Learning*.

Lee, J. I. and Brunskill, E. (2012). The Impact on Individualizing Student Models on Necessary Practice Opportunities. In *Proc. EDM*.

Pardos, Z. A. and Heffernan, N. T. (2010a). Modeling Individualization in a Bayesian Networks Implementation of Knowledge Tracing. In *Proc. UMAP*.

Pardos, Z. A. and Heffernan, N. T. (2010b). Navigating the parameter space of Bayesian Knowledge Tracing models: Visualizations of the convergence of the Expectation Maximization algorithm. In *Proc. EDM*.

Piech, C., Sahami, M., Koller, D., Cooper, S., and Blikstein, P. (2012). Modeling how students learn to program. In *Proceedings of the 43rd ACM technical symposium on Computer Science Education*.

Rafferty, A. N., Brunskill, E., Griffiths, T. L., and Shafto, P. (2011). Faster teaching by POMDP planning. In *Proc. AIED*.

Rafferty, A. N., Davenport, J., and Brunskill, E. (2013). Estimating Student Knowledge from Paired Interaction Data. In *Proc. EDM*.

Rafferty, A. N., Lamar, M., and Griffiths, T. L. (2012). Inferring learners' knowledge from observed actions. In *Proc. EDM*.

Rafferty, A. N. and Yudelson, M. (2007). Applying Learning Factors Analysis to Build Stereotypic Student Models. In *Proc. AIED*.

Reye, J. (2004). Student Modelling Based on Belief Networks. *IJAIED*.

Rockafellar, R. T. (1970). *Convex analysis*. Princeton University Press.

Schwing, A. G., Hazan, T., Pollefeys, M., and Urtasun, R. (2012). Efficient Structured Prediction with Latent Variables for General Graphical Models. In *Proc. ICML*.

Tseng, P. (1993). Dual coordinate ascent methods for non-strictly convex minimization. *J. Mathematical Programming*.

von Aster, M. G. and Shalev, R. (2007). Number development and developmental dyscalculia. *Developmental Medicine and Child Neurology*.

Wainwright, M. J. and Jordan, M. I. (2008). *Graphical models, exponential families, and variational inference*. Foundations and Trends in ML.

Wang, Y. and Beck, J. (2013). Class vs. Student in a Bayesian Network Student Model. In *Proc. AIED*.

Wang, Y. and Heffernan, N. T. (2012). The student skill model. In *Proc. ITS*.

Yudelson, M. and Brunskill, E. (2012). Policy Building - An Extension To User Modeling. In *Proc. EDM*.

Yudelson, M. V., Koedinger, K. R., and Gordon, G. J. (2013). Individualized Bayesian Knowledge Tracing Models. In *Proc. AIED*.