A Systematic Review of Artificial Intelligence Techniques Applied to Smart Learning Environments

Luis M. Vaquero

February 2021

1 Introduction

THIS IS a NOT PEER-REVIEWED WORK IN PROGRESS: AN ONGOING VERSION OF THE MANUSCRIPT THAT I WILL UPDATE IN SUCCESSIVE VERSIONS

The selection of the best content for each stage of development is one of the hardest problems in education. The recent covid-19 pandemic has made these difficulties more apparent since parents had to support their children learning activities at home.

Artificial Intelligence (AI) and Machine Learning (ML) have become essential for Intelligent Tutoring Systems (ITS) and Massive Open Online Courses (MOOCs).

Previous analyses on the usage of ML techniques for ITSs have focused on specific algorithms and their application to small datasets [73], which used to be the case before MOOCs were broadly available.

Increased data availability has enabled the application of data-intensive techniques. [146] review 64 recent paper using Machine Learning (ML) to predict student performance. Their results show an ample majority of papers focusing on University-level studies and supervised learning and recommender system algorithms. [52] also show this clear trend towards data-intensive methods of delivery and more data hungry optimisation methods (see Table 1).

We reviewed 300 papers in the area of ITSs and structured their contents according to the 3 goals of modelling and predicting the performance of learners.

The functionality of Smart Learning systems can be broken down into several logical steps and horizontal considerations (such as security, ethics, and explainability):

- Knowledge tracing, the problem of estimating how a student's knowledge changes over time as they interact with content
- Adaptive behaviour

	1st Wave *1960-70*	*2nd Wave* *1970-2015*	*3rd Wave* *2015-2020*
Method of delivery	In classroom - computer assisted instruction	Intelligent tutoring systems	Massive Open Online Courses
Performance Prediction	Statistical analysis and mathematical psychology	Machine Learning	Deep Learning
Path Sequencing (optimisation)	Heuristic decision processes	Reinforcement Learning	Deep Reinforcement Learning

Table 1: An Overview of Recent Progress in Smart Learning Systems

1. what hints and nudges or support work best
2. matching student skills to the best content for their current abilities

- Assessment assistance - gathering actionable insights
- Ethics

The remaining of this paper is structured along these logical steps and horizontal considerations: 1) estimating student knowledge to fine tune support for learning (see Section 2), 2) facilitating adaptive behavior of the system presenting the content (Section 3) and 3) generating interpretable and actionable insights [134] (Section 4). Then, we dive into ethical aspects that plague learning systems and frameworks that have tried to provide a comprehensive view in Section 5.

2 Knowledge Tracing

Knowledge tracing aims to predict the outcomes of students over questions as they are interacting with a learning platform.

2.1 Domain Definition

Performance can be modelled at a very high level as follows [64]:

$$r - t \simeq f(h_t), h_t \simeq g(h_t 1) \tag{1}$$

where $r_t \in 0, 1$ is the response of the learner at step t (a value of 1 usually corresponds to a correct response and 0 corresponds to an incorrect one). h_t is an unobserved variable that represents the learner's current knowledge. The f and g functions model how responses and learner's knowledge evolve.

In addition to predicting learner performance, and ITS also needs to have information about the content available and how the skills of a learner match the skills required to learn a specific piece of content in a productive and non-frustrating way.

A knowledge component (KC) is a fact, skill, or principle required to succeed at a particular task or problem step. We refer to this specialised form of a cognitive model as a knowledge component model (KC model), and it is sometimes referred to alternatively in the literature as a Q-matrix (e.g., [7]). KC models are typically evaluated in conjunction with a statistical model. The statistical model uses the KC model mapping to make inferences about student knowledge based on performance across a variety of observable tasks/items.

2.2 Early Performance Prediction Methods

2.2.1 Regression Models

Linear/logistic regression has been utilised for performance and timing prediction in small datasets consisting of just a few tens of thousands of interactions [11].

Logistic regression models have been broadly used and tend to differ in their feature vectors.

In the Item Response Theory (IRT) [185], each student s is assumed to have an unobserved ability, represented by the scalar a_s. Each item i is assumed to have an unobserved difficulty level, represented by a scalar d_i. IRT specifies the probabilistic relationship between the predicted response, R_{si} and a_s and d_i. The simplest IRT is the one-parameter logistic (1PL) model, which includes just one item-associated parameter:

$$Pr(R_{si} = 1) = 1/(1 + exp(d_i a_s) \tag{2}$$

The 2PL model includes parameters that allow for student ability to have a non-uniform influence across items. The 3PL model adds an item-associated parameter for guessing in the case that the right answer is possible (like multiple choice questions).

In Performance Factors Analysis (PFA) methods, the probability of learning is computed using the retrospective data about number of failures and successes [132]. The feature vectors of the regression take this form:

$$Pr(R_{si} = 1) = 1/(1 + exp(-(\sum_k \beta_k + \gamma_k c_{s,k} + \rho_k f_{s,k})) \tag{3}$$

where β_k is the easiness of task k, $c_{s,k}$ is the number of correct answers of learner s on task k, and $f_{s,k}$ is the number of wrong answers of learner s on k prior to this attempt. The PFA model was better at distinguishing correct from incorrect responses, but it cannot adapt to present better items to students based on their prior performance [66].

DAS3H combines IRT and PFA features by adding continuous time-window features [33]:

$$Pr(R_{si}=1) = 1/(1+exp(-(d_i a_s+\sum_k \beta_k+\sum_k \sum_{w=0}^{W-1} \theta_{k,2w+1}\Phi(c_{s,k,w})-\theta_{k,2w+2}\Phi(a_{s,k,w}))) \tag{4}$$

where w indexes a set of expanding time windows, $c_{s,k,w}$ is the number of correct answers of learner s on task k in time window w and $\Phi(x) = log(1+x)$ re-scales the counts [33].

In the general model for KT models described above, IRT models rely on sigmoidal f functions and h_t is a real number. In a more recent refinement, the SPARFA-Trace method [96] used a simple affine transformation model as the explicit knowledge evolution model g

IRT models and their derivatives, in general, have to determine the entire interaction trajectory for each student with item parameters [193] and need large samples for calibration [156] and tend to be regarded as difficult to implement in an adaptive environment, but remained used because of their explainability.

2.2.2 Markov Models

Bayesian Knowledge Tracing (BKT) [42, 179] consists of a simple Hidden Markov Model where the performance prediction was dependent on several binary variables: two representing performance (probability of learning P(T)) and two latent variables representing the knowledge on a single skill (probability of guessing $P(G)$ and of slipping $P(S)$). This is shown in Figure 1.

The model assumes the learner never forgets a learned skill and every new question has a fixed probability of helping to learn. BKT is usually fit with expectation-maximization or brute-force search. While the model is trained, the probability of mastery is updated with Bayesian inference after every response. Determining knowledge priors is essential for a smooth usage of the system in the early stages.

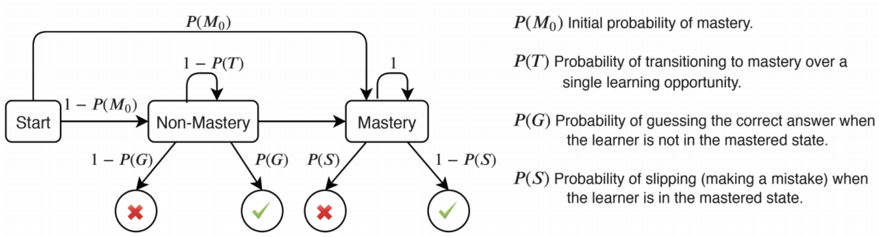
$P(M_0)$
$1 - P(T)$
1
Start
$1 - P(M_0)$
Non-Mastery
Mastery
$1 - P(G)$
$P(G)$
$P(S)$
$1 - P(S)$
$P(M_0)$ Initial probability of mastery.
$P(T)$ Probability of transitioning to mastery over a single learning opportunity.
$P(G)$ Probability of guessing the correct answer when the learner is not in the mastered state.
$P(S)$ Probability of slipping (making a mistake) when the learner is in the mastered state.

Figure 1: Markov process representation of Bayesian Knowledge Tracing (BKT). Mastery and Non-Mastery are latent states and arrow values are the probabilities of transition and observation. Taken from [63]

This initial model has been extended in a wave of works improving on several aspects such as: adding probability of forgetting, joint computation of the multiple skills (vs single skills) [199, 89], skill importance [23, 65], skill combinations like in Conjunctive Knowledge Tracing (CKT), which combines skills and knowledge to predict student performance [93], personalisation (multiple priors for student knowledge [131] or individual student modelling [99]), or timing [127].

BKT+ combines personalisation, forgetting, and automatically discovering the Markovian model, delivering a better performance than other BKT models at the expense of a more complex implementation and slower fitting times [90]. Similar temporal extensions were done by [143].

Markov process methods, like Bayesian Knowledge Tracing, lag behind other approaches. Availability of large scale datasets has made deep learning models for learner ready to compete with Markov process and logistic regression models [63]. The authors explore performance over 9 datasets to conclude that logistic regression is preferable with datasets of moderate size, while Deep Knowledge Tracing does better on large size datasets or if precise temporal information matters most.

In this type of early models, knowledge (h_t) is a binary number that shows if the learner has acquired the knowledge required for grasping a (single) concept as covered by a question. $r+t$ is also binary so $fandg$ become modulators based on slipping, learning, and forgetting probabilities[93, 131, 99, 127, 90].

2.2.3 Recommender Methods

An intelligent tutoring system can be seen as a type of recommender system: students are users and questions are items to the recommendation systems.

The ability to use recommender system techniques builds in two assumptions: 1) students with similar knowledge levels will perform similarly in solving problems; 2) a student will have similar performance on two problems with KCs.

Collaborative filtering (CF) involves inferring a relatively compact set of latent variables that can predict or explain the observed data. [14] automatically define a logistic function (IRT) to fit the model of the student responses over different items. GPCF for Gaussian Process with Collaborative Filtering builds on the idea of collaborative filtering. Each student is modelled with a temporal Gaussian Process (GP) joining predictions from other GPs to make accurate predictions [44].

[159] Matrix Factorization (MF) performance prediction is particularly appealing because it does not require tagging involved skills in tasks. However, MF's difficult interpretability does not allow to show the student's s state evolution. Kalman Filters have used in a way that the student state is updated at each interaction and it models the individual progress of the students over time that could be later used to develop novel sequencing policies.

For CF, factorization machines (FMs) were designed to provide a way to encode side information about items or users into the model. [179, 177] employ FMs in their regression form for student modeling, where they use root mean squared error as the main error metric. [186] proposed to use a Deep FM model to minimise feature engineering in 2_{nd} language learners in the Duolingo app.

[187] show how FMs encompass most of the previously existing models (additive factor model, PAF, and multidimensional IRT) as particular cases.

[50] use a rank-based tensor factorisation method (RBTF) to predict the performance by considering the gradual learning of students as a ranking problem.

[3] use a combination of Latent Dirichlet Allocation (LDA) to find the similarity between topics in the descriptions of their projects and profiles of learners and cosine similarity for differences between KCs in the projects to build a similarity matrix between students and courses.

2.3 Modern Methods

Modern methods since 2010 have focused on adding non-linear models to approximate the knowledge evolution function g.

2.3.1 Deep-Learning Methods

[137] introduced deep learning into the educational data mining landscape. As shown in Figure 2, DKT is a recurrent neural network (RNN). At time step t, the previous interaction $x_{t1} = (q_{t1}, a_{t1})$ is one-hot encoded as $u_t = \phi(x_{t1}) = \delta(q_{t1} + a_{t1}I)$, where I is the total number of items. DKT uses an RNN variant with long short-term memory (LSTM) cells, which can better exploit long-term dependencies in the data.

Only small performance improvements have been achieved since DKT [210, 205, 98]. [205] introduced regularisation terms to the loss function of the original DKT to account for the ability of the model to reconstruct student's knowledge (if a student performs well on a KC, the prediction of that KC's mastery level should increase) and to keep consistency in the predicted performance for across time-steps. [98]

The dynamic key-value memory networks (DKVMN) work extended DKT by using an external memory matrix (H_t) to model knowledge so that it has two main elements: the static, "key" matrix that contains a fixed representation of each concept, and a dynamic, "value" matrix that contains the dynamically changing knowledge levels of each learner on each concept [210].

[91] showed how a fine-tuned BKT incorporating additional elements like forgetfulness or skill relationships brings performance very close to that of DKT. [29] also show how adding a graph of dependencies between KCs can improve performance.

When an exercise is tagged under two or more skill, DKT encodes it as a sequence of multiple single skills (rather than as a new combined skill). Special attention is required before using DKT to ensure multiple skill sequencing is processed correctly [198]. qDKT [171] uses a graph Laplacian regulariser to tale question similarity information into account for a question being related to more than one skill.

Exercise-aware Knowledge Tracing(EKT) [108] outperforms DKT using a

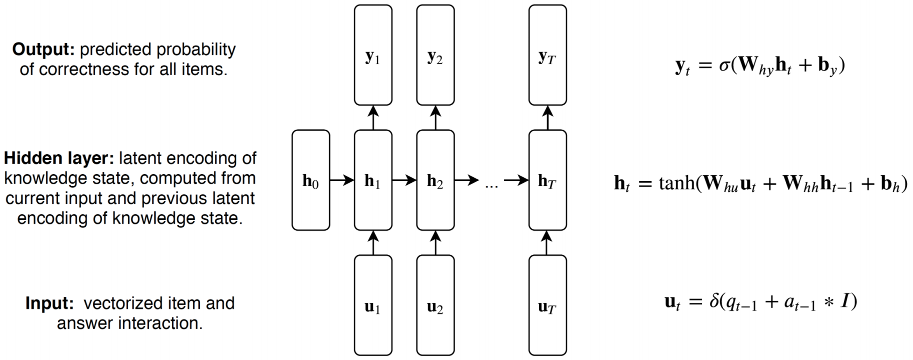
Output: predicted probability of correctness for all items.
Hidden layer: latent encoding of knowledge state, computed from current input and previous latent encoding of knowledge state.
Input: vectorized item and answer interaction.
$\mathbf{y}_1$
$\mathbf{y}_2$
$\mathbf{y}_T$
$\mathbf{h}_0$
$\mathbf{h}_1$
$\mathbf{h}_2$
...
$\mathbf{h}_T$
$\mathbf{u}_1$
$\mathbf{u}_2$
$\mathbf{u}_T$
$\mathbf{y}_t = \sigma(\mathbf{W}_{hy}\mathbf{h}_t + \mathbf{b}_y)$
$\mathbf{h}_t = \tanh(\mathbf{W}_{hu}\mathbf{u}_t + \mathbf{W}_{hh}\mathbf{h}_{t-1} + \mathbf{b}_h)$
$\mathbf{u}_t = \delta(q_{t-1} + a_{t-1} * I)$

Figure 2: Recurrent neural network representation of Deep Knowledge Tracing (DKT). Taken from [63]

simple Bidirectional Long-Short Term Memory (Bi-LSTM) model with an attention mechanism built on the top layer, similar to [208].

The main advantage of the Transformer over RNN is its ability to learn long-range dependencies, which is a limitation of DKT [63]. Thus, Transformer models have been the first to consistently improve the performance of DKT and BKT+. As learners practice different items, skills are automatically correlated with the responses given to previous items and attention mechanisms learn inter-dependencies among items and interactions. [142] modified the basic transformer to learn a matrix that relates KCs to question items and handles cases where several KCs are associated with a single item. The temporal aspects or learning and their relation to students forgetting prior knowledge has also been explored as extensions to the original RNN in DKT [124].

[129] use self-attentive knowledge tracing, which builds on attention mechanisms to relate query with responses. [35] used a Transformer encoder to pre-train students' interactions. Choi et al. [34] experimented with an encoder-decoder model for knowledge tracing that applies deep self-attentive layers to exercises and responses separately, reporting gains in the AUC under 2%. Building on this work, Choi at al. by considering the time taken for a student to respond and the time passed between subsequent learning activities [168]. Mao et al. add an exponential time decay to gate the hidden previous state of an LSTM to account for the time in between activity of the user [112]. [142] report that the original Transformer gains 10% AUC simply by adding the interaction-skill mapping and time-bias to its structure. [130] follow a similar reasoning by modelling relations between exercises and time elapsed since the last interaction into the self-attention mechanism they used in [129].

Some models [129, 108] have shallow attention layers (only one attention layer in some cases) that cannot learn the relationship between items and interactions. They also link items with attention queries and interactions with attention keys and values. [34]encode exercise/item embeddings as queries, keys

	Attention Mechanism
[129]	multi-headed self-attention model, using question embeddings for mapping queries whereas response embeddings for key and value mapping.
[34]	
[142]	classic single headed attention modified to learn KC-item relationship including time gap bias correction
[212, 76]	classic multi-headed attention
[64]	two self-attentive encoders, one for contextualised questions and other for contextualised answers (knowledge) and a monotonic multihead attention linking learner's future responses with assessment questions to their past responses
[211]	attention applied to relating input and output sequences and updating an external memory base

Table 2: A Summary of the Attention Mechanisms Employed in Knowledge Tracing

and values, and pass this to a repeated self-attention mechanism. The decoder takes the sequential input of response embeddings as queries, keys and values and finally applies self-attention and attention alternatively to the output of the encode.

Attention being one key element to many transformer models, [64] proposed attentive knowledge tracing (AKT), which uses attention-based neural network models with a focus on interpretability. More specifically, attention weights are combined with exponential decay and a context-aware relative distance measure, in addition to features representing question similarity. The Rasch (1PL IRT) model is employed to regularise KCs and question embeddings.

Attention has also been used to handle the so-called coldstart problem: not enough data from learners to train a useful/non-frustrating model (e.g. when a new course is launched). Zhao et al. [211] use attention-based mechanisms to alleviate the problem of cold start, based on a Neural Turing Machine.

Table 2 shows how most papers have relied on simple attention mechanisms built from question-answer embeddings. [64] have extended simple attention by including a representations of past questions and responses too. Unlike [129], these authors employ question embeddings for mapping both queries and keys. They also added an exponential decay factor into the historical representation to reduce the weight given to questions far in back in the past. These changes seem to have an important effect by up to 6% in AUC in some cases.

[129] split the question sequence into segments and train the model on batches of segments, thus ignoring information from previous segments and making their model less suited for long sentences. KT-XL [76] aims to enable Transformer architectures cope with long questions. The authors also split long sentences into segments and (after applying the embedding and positional encoding) the hidden state sequence computed for a previous segment is kept in order to reuse it together with the next segment.

The advantages of some transformer models have been questioned by [63], mentioning that DKT outperforms [129], probably because of the usage of too small datasets for attention mechanisms to be able to find meaningful relation-

Paper	Temporal Modelling	Model
[132]	Decay function	PFA
[124]	Modified input considering: repeated time gap,sequence time gap, and past trial counts	DKT
[143, 90]	Additional states and conditional probability transitions	BKT
[168]	Modified attention	Transformer
[180]	tensor factorisation	matrix factorisation
[112]	Exponential time decay in gate	LSTM
[130]	Attention weights corresponding to previous interactions are used for the prediction of the outcome of the next exercise	Transformer
[175]	Attention over the past interactions using cosine similarity between the past interactions and the next exercise	RNN

Table 3: A Summary on Methods Used to Model How Learners Forget

ships.

[200] have recently used convolutional networks to model long term memory decay in students as a key factor to predict their performance. Their Convolutional Knowledge Tracing (CKT) model consists of 3D Convolutional Network fed forward in parallel to an LSTM. The outputs of these are joined via an an update gate. A final LSTM produces the final prediction.

Table 3 shows a summary of the main methods used by different KT models to mimic the observed forgetting in skills that can be observed in learners after periods of not practicing enough. Many modern modelling techniques have incorporated these mechanisms¿ For instance, simple decay functions were used in classic PFAs, additional states in BKT, and modified attention mechanisms in Transformer-based methods.

2.3.2 Graph Neural Networks

HMMs and Dynamic Bayesian Networks (DBNs) are generative models called Probabilistic Graphical Models (PGM) that model conditional dependencies (i.e., interactions)between random variables. Formally, a PGM is a graph $G = (X, E)$, where the random variables X are represented as nodes in a graph,and the conditional dependencies between X are described by the edges E.

Incorporating prior knowledge about the graph structure improves performance and interpretability [10]. Traditional methods for student's performance prediction usually ignore the relationships between multiple courses, multiple students, external contents and how students acquire knowledge across them.

Graph Neural Networks (GNNs) are deep neural networks that were specifically designed for graphs. They build on the idea of convolution either done as a simple multiplication in the spectral domain or building on the idea of convolution as pattern matching in the spatial/topological domain (see Table 5 for a classification of works).

Spectral methods operate by building the graph normalised Laplacian (a representation of the smoothness of the features of the neighbourhood of a node) $\Delta = I - D^{-1/2}AD^{-1/2}$ where $D = diag \sum_{j \neq i}(A_{i,j})$. The Spectrum of the Laplacian corresponds to a diagonal matrix Λ that has the eigenvalues λ_k as elements in the diagonal and the Fourier functions correspond to a matrix Φ containing the eigenvectors ϕ_k as columns. For the normalised Laplacian the eigenvalues are bounded $0 \leq \lambda_1 \leq \lambda_2 .... \leq \lambda_n \leq \lambda_{max} = 2$. Thus, $\Delta = \Phi^T \Lambda \Phi$.

Thus, any vector h is taken to the spectral domain simply by $\mathcal{F}(h) = \Phi^T h = \hat{h}$ and the reverse Fourier transform is calculated as $\mathcal{F}^{-1}(\hat{h}) = \Phi \hat{h}$. A convolution $w * h$ of two vectors w and v in the spectral domain is just a pointwise product of their Fourier transforms: $\mathcal{F}^{-1}(\mathcal{F}(w) \circledast \mathcal{F}(h)) = \Phi^T w \Delta v$.

Graph spectral convolution is simply a succession of layers where:

$$h^{l+1} = \eta(w^l * h^l) = \eta(\Phi \hat{w}^l \Lambda \Phi^T h^l) \tag{5}$$

where $\hat{w}^l(\Lambda)$ is learned by backpropagation.

Topological GNNs can also be seen as a classic message passing system, where the state of a node n at time t is represented by h_t^n. This is calculated by the application of a function q, which depends on the previous state h_{t-1}^n and an aggregation function of all the messages $\bigcup_{k, \forall n_j / n \rightarrow n_j} f_t(h_{t-1}^n, k, h_{t-1}^{n_j})$ rendering:

$$h_t^n = q(h_{t-1}, \bigcup_{k, \forall n_j / n \rightarrow n_j} f_t(h_{t-1}^n, k, h_{t-1}^{n_j})) \tag{6}$$

where k is the type of edge connecting node n and its j neighbour.

Table 4 how very similar graph neural network models were used to support several functions. In some cases the graph is pre-created (while in some other instances it is inferred from the data). Also, the type of graph considered changes across works: in some cases the graph includes just relationships between skills (see Subsection 2.6), while in some cases a more complete graph including relationships between students and interactions is taken into consideration. It is worth mentioning that GCN are often used as a form of feature engineering which output is fed into a subsequent model such as a RNN and an LSTM.

[125] incorporate Graph Neural Networks into their KT model and propose Graph-based Knowledge Tracing (GKT). KC become graph nodes and graph edges represent dependencies between concepts. GKT aggregates the node features of the KCs related to the question the student responded to; GKT also updates the knowledge level of the student for those KCs. The updating process employs a multilayer perceptron (MLP) layer, an erase gate such as the one in DKVMN [210], and a classic gated recurrent unit (GRU). In [125], the graph structure is learned in parallel with the optimisation of the performance prediction.

	Type of Model	Optimisation	Evaluation	Graph
GKT [125]	GCN	Minibatch stochastic gradient descent	AUC	Relationships between KCs
GIKT [201]	GCN + LSTM	Minibatch stochastic gradient descent	AUC	Relationships between KCs and questions
R^2GCN [101]	GCN + residual layer to convolutional networks	Minibatch stochastic gradient descent	AUC	Graph of questions, students, and interactions
AGCN [78]	GCN	Minibatch stochastic gradient descent	mean absolute error (MAE) and percentage of tick accuracy (PTA)	Relationships between Courses taken
HGNN [181]	GCN + RNN	Minibatch stochastic gradient descent	AUC	Relationships between Courses taken

Table 4: A Summary of Recent Graph Neural Network Models Used in Knowledge Tracing

The work by Nakagawa et al. fails to consider the dynamic structure of the graph, which reflects the knowledge acquisition change over time steps, which could be added by considering an additional attention layer like in [25].

Attention-based Graph Convolutional Networks (AGCN) [78] learn a graph embedding of the graph of frequently taken courses and then apply the attention mechanism to generate a weighted embedding for predicted scores. Courses taken in the current term are fully connected to courses taken in the next term; 1 represents connected, 0 otherwise in the graph adjacency matrix. AAGCNs work well for homogeneous graphs with a single type of node and edges.

[101] build a heterogeneous large graph, consisting of questions, students, and the interactions (in the form of mouse journeys) between them to formulate the prediction of 4 scores per question as a classification problem. This is done by adapting GNNs to include a residual connection to different convolutional layers and original features; hence the name of the model: Residual Relational Graph Neural Network (R^2GCN).

GNNs have also been used as a way to create more complex embeddings that are then fed into other KT model, often a form of recurrent neural network to extract knowledge from temporal dependencies. [201] proposed the Graph-based Interaction model for Knowledge Tracing (GIKT), which employs a graph convolutional network (GCN) to incorporate question-skill correlations via embedding propagation into any previous KT model (e.g a model based in RNNs). Building on this idea, Hierarchical graph neural networks (HGNN) are proposed to account for the hierarchical nature of some exercises and to propagate semantic features of exercises into problem schema embeddings [181]. Karimi et al [88] built Deep Online Performance Evaluation (DOPE), which also feeds a student to course relational graph (using student clicks to represent how students have interacted with the course) into a GNN to obtain embeddings.

Spectral Graph Convnet	***Spatial Graph Convnet***
	[125, 78, 201, 101, 181, 88]

Table 5: Type of Graph Neural Network Used to Support the KT Problem.

Graph Creation

[125] propose to learn the graph structure concurrently with the optimisation of the performance prediction task; this is done through a Parametric Adjacency Matrix (PAM) that is learnt together with Multi-head Attention to infer the edge weights between two nodes based on node features.

Other approaches build a graph of related exercise by measuring the cosine similarity of the BERT embeddings of two exercises [181]. The same authors also build a transition probability matrix, where $A_{i,j}$ is 1 if $n_{i,j}/\sum_{k=1}^{\epsilon} n_{i,k}w_e$ where $n_{i,j}$ represents the number of times exercise j is answered immediately after exercise i and w_e is a hyperparameter controlling the number of edges in the generated graph.

[136] propose classic association rule mining on text to infer the structure of precedence relationships between skills.

Table 6 shows a summary of the models utilised for KT in the literature.

2.4 Features Considered

All the families of models described above have followed a similar evolution process, starting from simple models that consider mainly exercise execution performance, growing into richer models that include fine grained student interactions and timings.

IRT also includes just response features [185], but some techniques learn from a common matrix including a representation of student knowledge in combination with exercise features [46]. The Multidimensional Item Response Theory(MIRT) [202] enabled several features to be considered but it still failed to exploit semantic representation from question texts. Deep Item Response Theory (DIRT) starts with classic feature engineering: a proficiency vector reflects student proficiency on each KC; they also compute an embedding representation for the text of the exercises. A deep neural net is then used to relate texts KCs. These are fed into the item response function to predict student performance [31]. DAS3H dded timing features into the family of IRT-based models [33].

BKT includes skill-specific features only, with no consideration to features to model student behaviour [42]. In order to consider the variability of students, some extensions to BTK include student-specific parameters (such as the initial knowledge mastery level and the learning speed). [206] proposed a personalised BKT model that includes student-specific parameters (e.g. starting knowledge level and the learning speed per student). CKT added a representation of knowledge [93] and this was followed by individualised features per learner [131,

	Type of Model	Optimisation	Evaluation	Skill Dependency
BKT [42]	HMM	Curve fitting	MAE	Assumes independent skills
IBKT [99]	HMM	Minibatch stochastic gradient descent	RMSE	Assumes independent skills
IRT, PFA [185, 132, 33, 96]	Regression	Curve fitting	RMSE	Assumes independent skills
DBN [89]	DBN	Constrained latent structure	AUC, RMSE, Brier Score	Hardcoded skill dependencies
[159]	Matrix factorisation	Gradient descent	RMSE	Discover the inter-skill dependencies
[177]	Factorisation machines (regression)	Curve fitting	RMSE	Assumes independent skills
DKT [137]	LSTM	Minibatch stochastic gradient descent	AUC, Accuracy	Discover the inter-skill similarities/dependencies
GAKT [175]	LSTM	Minibatch stochastic gradient descent	AUC	DNA
EKT [108]	bi-LSTM	Minibatch stochastic gradient descent	AUC,	Explicit model of relationships between KCs
NAKTM [208]	LSTM	Minibatch stochastic gradient descent	AUC	NA
Knowledge Query Network(KQN) [98]	LSTM/GRU	Minibatch stochastic gradient descent	AUC	probabilistic skill similarity: relates the cosine and Euclidean distances between the skill to the odds ratios for the corresponding skills
T-LSTM [112]	modified LSTM to account for time between interactions	Minibatch stochastic gradient descent	AUC, precision, recall, F1	NA
DKVMN [210]	Memory Augmented NN	Minibatch stochastic gradient descent	AUC, Accuracy	Discover the inter-skill similarities/dependencies
FM [186]	Factorisation Machine	Negative log-likelihood	AUC, Accuracy	Discover the inter-skill similarities/dependencies
Transformer Models [129, 35, 35]	Transformer	Minibatch stochastic gradient descent	AUC	Discover inter skill dependencies
Transformer [142]	Transformer	Minibatch stochastic gradient descent	AUC	Discovery of latent interaction-skill mapping and time-bias
AKT [64]	Attention model with decay and IRT for KC regularisation	Minibatch stochastic gradient descent	AUC	Discover the inter-skill similarities/dependencies
CKT [200]	Convolutional nets and LSTMs	Minibatch stochastic gradient descent	AUC	NA
KT-XL [76]	Transformer XL	Minibatch stochastic gradient descent	AUC	NA
GKT [125]	Graph Convolutional Network	Minibatch stochastic gradient descent	AUC	Not explicitly
AGCN [78]	Graph Convolutional Network with multi head attention	Minibatch stochastic gradient descent	AUC	Implicit through course dependencies

Table 6: Overview of Recent Models Used in Knowledge Tracing

Exercise features	Text/video features	Timing	Static learner	Dynamic learner
[185, 42]	[29, 97, 137, 31, 129, 76, 136]	[127, 58, 29, 90, 143, 64, 213]	[93, 202, 206, 131, 99]	[210, 35, 58, 213, 78, 136]

Table 7: A Summary of the Type of Features Used by Many Recent Works.

99]), only too make timing of those interactions available for consideration into the model [127]. BKT+ and similar works combine personalisation and timing into a single Markovian model [90, 143].

[210] (DKVMN) uses a static key matrix (for KCs) and a dynamic value matrix (for updates on the level of knowledge of the student) and [29] use students exercise records and text of exercises to model student state.

Deep neural networks replace the need to do feature engineering for a need to explore different neural architectures and hyperparameter tuning. DKT adds timing between exercises into consideration [137] and other works have recently started to include more fine grained student interactions with the learning system [35]. [108] use attention to combine student features with exercise features. [129, 76] include specific text processing into neural architectures.

Most previous works have focused on graded/assessed materials that make it easier to associate higher grades with knowledge acquisition. There are other contents that students may use that also contribute to knowledge creation but are not easily linked to knowledge acquisition through an assessment. [213] provides a factorisation method that tries to generalise how to bring all types of content in to the same latent space.

Graph-based techniques allow to include dependencies between some of the features to be considered. For instance, AGCN [78] enables to find course dependencies (which could be related to implicit skill dependencies).

As can be observed in 7, the initial seminal works in the area dealt with performance results only [185, 42]. A second wave of efforts started to include timing effects and the influence of forgetting [127, 58, 29, 90, 143, 64, 213]. More recent papers have tried to include dynamic learner features [210, 35, 58, 213, 78] and even include audiovisual based information from the content the learners go through [93, 202, 206, 131, 99].

2.5 Interpretability

Explanatory modelling efforts tend to start with "clean" independent variables that have either simple functions or map to clearly defined concepts. Then all the dependent variables can also be mapped to well-defined constructs. One consistent theme across findings is that feature representations motivated by interpretable, theoretical frameworks have been among the most promising [153]. Finally, explanatory models tend to have fewer estimated parameters

No existing KT method truly excels at both future performance prediction and interpretability. Early KT methods exhibit excellent interpretability but do not provide state-of-the-art performance on future learner performance prediction.

Recent KT methods based on deep learning excel at that, but offer limited interpretability. There have been a number of works that have used simpler methods to try to explain the decisions made by more complex models. For example, [109] use the layer-wise relevance propagation (LRP) method to interpret DKT models. The authors backpropagate the relevance from the model's output layer to its input layer. [204], used the DKVMN by [210] to determine the student ability and the item difficulty over time. The IRT model is then employed to determine the probability that a student will answer an item correctly using her estimated ability and the item's difficulty.

[164] explore the idea of creating an ensemble of 22 different models (e.g. IRT, deep tracing, gradient boosting) and mention that differences in the probabilistic weight of those groups of models could be used to increase explainability, without adding enough details on how this would be implemented.

2.6 Item to Skill Labelling

The item-skill labelling is a close concept to Q-matrices in psychometrics, which is a binary (or real valued to indicate strength) matrix depicting the mapping of assessment items to skills.

KT typically requires experts to define the Q-matrix relating item to skill, this can be done in two main ways: 1) specific modelling or 2) by manually labelling the skills associated to an item (often through structured interviews or think-aloud protocols).

As examples of (1), knowledge can be represented by a Hidden Markov Model (HMM) for each skill [42] or the knowledge-learning-instruction (KLI) framework [92] represents a KC as a fine-grained decomposition of the knowledge targeted in the instruction (set of labels), together with a mapping that specifies which items (questions) involve which KCs.

In the case of manual labelling (2), [24, 96**?**] require experts to annotate every item, with some items having thousands of items to label. [121] showed that crowd workers are useful to identify a subset of KCs and pre-training was not needed, hence demonstrating how labelling can be crowdsourced (a larger sale example of (2)). Novice students tend to label items by using superficial features, such as "words in problem text" while senior students use a more principled approach to categorising items into KCs, such as "conservation of momentum" [67], which may make scaling manual efforts difficult as finding experts may be difficult and the labels obtained from a crowd may end up been more superficial.

[107] adopted semi-automation in the form of Bayesian approach that uses available expert labels but an item depends on a single KC in a binary manner, which may not be representative of many learning scenarios. Stamper and Koedinger (2011) [174] proposed another semi-automated approach, iterating as follows: 1) fitting statistical model with a pre-built KC model, 2) inspecting

learning curve visualisations and best parameter estimates, 3) identifying and tweaking problematic KCs and 4) re-fit the statistical model with the revised KC model and evaluate for improvements. This process still requires notorious human effort. Learning Factors Analysis aimed to automate part of this refinement of KCs by using combinatorial search across potential KCs extracted from previous cognitive models [24]. Refinement of KCs can also be approached as a classification problem [119].

There are more automated approaches. [8] inferred the mapping between items and latent skills by taking several initial mappings and using a search algorithm to enhance the mapping. [220] have tried to extract and score KCs from books chapters using classic techniques [7].

[67] present 3 different approaches to automating item labelling. One based on matrix factorisation and clustering (where each cluster defines the item to KC mapping), a second one based on dynamic HMMs (sampling from the HMM's forward-backward probabilities to find skills), and the third one relying on superficial clues (e.g. similar texts) to group items together (again assuming items in the same group require similar KCs). This third method just requires an expert-defined feature extraction function to quantify the similarity between any pair of items. The last two methods are not easy to interpret (figuring our actual KCs in the groups/samples requires human expert interpretation).

Collaborative filtering has been employed to assess an expert-generated Q-matrix [55] or to build the Q-matrix from scratch [47, 48, 176]. LogCF can retrieve from scratch item-skill associations from scratch by including learner actions and timings into a deep collaborative filter model [28].

Given an item response matrix $R \in \mathbb{R}^{mxn}$ consisting of item responses of m learners to n items, matrix factorisation decomposes R into two low-rank matrices $U \in \mathbb{R}^{mk}$ and $V \in \mathbb{R}^{nxk}$, which represent the learner-skill and the item-skill association matrices of rank k, respectively.

Matrix-factorisation techniques build on the assumption that student patterns of correct and incorrect responses have less variance within skills than between skills. This variance differential is exploited infer the latent skill structure of the items. Matrix factorisation approaches been shown to have high accuracy when working with few relatively independent latent skills on static data, but does not work well when data changes frequently and there are hundreds of partially overlapped skills to be found [170].

Despite the interpretability of lower dimensionality matrices in the factorisation process,the linear combination (the inner-products) of learner-skill and item-skill associations cannot fully explain the complexity of learner-item interactions. Collaborative filtering approaches have been extended with deep learner to ease the relationship between learner by exploiting information sources beyond performance data [28]

There is a trade-off between full automation and interpretability. Automated approaches tend to be less interpretable. For instance, [190] not only aim to extract KCs automatically, but also they created a directed graph, where each node is a KC and directed edges represent their prerequisite dependencies. They identify words in books and extract embeddings and relate the words to

Wikipedia anchors and measure distance of the words in the table of concepts of the books to figure out relevant KCs and their relationships at the same time. A similar method is used for Youtube lectures used on subtitle information [6]. This method would keep explainability and automation, but it may not be as easy to do for purely visual content or content where external books or Wikipedia elements are not available.

Cognitive Representation Learner(CogRL) is a fully automated KC extraction technique that uses representation learning to automatically find KCs for items in domains that are are highly visual or probabilistic domains including many exceptions[26]. The authors train a domain-specific CNN (or RNN) to extract a vectorial representation of the images or text in a domain. This representation is fed into a final layer that predicts the right answer for any item in the domain. The vectorial representations for all the problems in a domain are then used as a estimated Q-matrix relating problems with KCs. These automated representations are not really interpretable KCs.

[50] avoid the problem of creating interpretable KCs altogether by learning student, problem, and time-based biases as a single lower-dimensional space and they add a rank-based constraint to account for forgetting of concepts. This results in potentially less interpretable insights (even when latent variables are kept non-negative).

SimStudent is an intelligent agent that inductively learns knowledge, in the form of rules (each rule mapping to a single KC), by observing a tutor solve sample problems and by solving problems on its own and receiving feedback [102]. These production rules linked to a single KC tend to be more explainable, but learning systems may elicit hundreds of them, making the interpretability of them all and their interaction a bit less clear.

[170] propose to use correlated topic modeling to exploit the linguistic features of the items in the content of mathematics problems to extract the set of latent skills associated to each item. While the results are more interpretable than the results obtained with variance difference techniques, understanding partial overlaps between extracted topics can be difficult.

[98] aims at improving the interpretability of the interaction between items and skills by encoding student learning activities into knowledge state and skill vectors, and modelling the interactions between the two types of vectors with the dot product.

Beyond the level of automation, the ability of some models to deal with *longitudinal data* (data captured over a sustained and representative time period): "do not distinguish between poor performance at early time steps and poor performance after a lot of practice" [67]. This includes matrix-based methods (such as [191, 7, 97, 180]). Matrix factorisation uses a student to exercise matrix whose cells indicate whether a student has answered an exercise correctly. The goal is to obtain a vector for each student characterising the degree to which the student has learned each KC and a vector for each exercise indicating what KCs are required by each items. The lack of longitudinal data support highlighted by [67] has been worked around via adding dynamical models of knowledge state or by extending the matrix into a tensor whose 3^{rd} dimension to capture time.

2.7 Performance Metrics

Probabilistic understanding of errors tends to rely on Root Mean Square Error (RMSE), which has also been shown to be highly correlated with the log-likelihood function. This is, in turn, related with a key indicator of knowledge mastery: the 'moment of knowledge acquisition' [133]. The Brier Score (a RMSE without the squared error) is useful to gain insight into the reliability, resolution and uncertainty [156].

Most models do, however, use the overall AUC.

Per-skill and across-skill AUCs differ in two regards. First, per-skill AUC weighs all skills equally in the final computation, whereas across-skill AUC weighs all trials equally. Because some skills have far more trials than others, and because one would expect any model to perform more poorly on skills for which there are less data, the per-skill AUC tends to be lower than the across-skill AUC.Second, when skills are separated via the per-skill AUC, the averaged AUC score does not reflect a model's ability to predict relative accuracy of one skill versus another; in contrast, the across-skill AUC improves to the degree that model predictions capture the relative accuracy across skills. Thus,the per-skill AUC will tend to be lower, as long as a model can predict the relative difficulty of skills—the sort of base rate statistic that should be readily learned.

When calculated by averaging per-skill AUCs and assigning an even weight to all skills, the AUC value is lower than when evenly weighting all trials. DKT [137] and DKVMN [210] employ the latter as deep learning models may not have the skill notation of each question [156].

Overfitting is a common source of error in machine learning models and Cross Validation (CV) is, therefore, a key element to mitigate the impact of overfitting. The number of folds is determined by user stratification: the mean performance of students in a fold is approximately equal across folds [156].

3 Adaptive Behaviour

Having a prediction of the performance of a student giving a task is not enough to make actionable decisions. There is a number of tasks that an ITS needs to make to enhance the experience of learners.

3.1 Learning Path Recommendation

KT models are commonly applied in curriculum sequencing (recommendation of a dynamic, optimal sequence for a learner) and mastery learning (estimation of the when a given skill will be acquired) frameworks [152].

A simple heuristic to use performance prediction from students to create a learning itinerary could be to tackle the tasks with higher probability of success, but this would come with plenty of inconveniences (eg. students not challenged, stuck in the same kind of lessons). Thus, schedules/recommendations must

Work	*Optimised Metrics in Sequencing*
[106]	Normalised learning gain $NLG = \frac{PosttestscorePretestscore}{1-Pretestscore}$
[215, 167, 3, 209, 45, 9, 53, 192, 162, 144, 148],	Learning performance adjusted with post-test scores between control and RL policy groups
[111]	Number of levels completed
[145, 100, 147, 103, 4, 166]	Overall completion time
[12, 157]	Completion time per item
[158]	Total knowledge (number of KCs mastered)
[158, 82]	Total number of items
[45]	Engagement (interaction time and number of log-ins, as well as their reported opinion)
[9]	Dropout rates
[38]	Personalisation (propose more difficult exercises sooner and keep proposing easier exercises longer)

Table 8: A Summary of the Main Metrics Optimised by the applied Heuristic/Policy Compared to a Baseline

balance competing priorities of introducing new items and reviewing old items in order to maximise learning[1].

Table 8 offers an overview of the metrics that different works have tried to optimise in their sequencing efforts. AS can be observed, they are mainly centered in completion time and/or performance relative to a baseline (often a reasonable policy chosen by a human).

[52] review available work and categorise the sequencing tasks based on the type of content:

- Paired-associate learning tasks: e.g. learning the translation of a word across languages; these techniques date back to the first wave of smart learning systems (see Table 1) and rely on use of statistical psychological models of human learning.
- Concept learning tasks: e.g. a student is presented with examples that either belong or do not belong to an initially unknown concept, and the goal is to learn what constitutes that concept. The parts that constitute the concept are mutually interdependent, but in a very particular way.
- Sequencing interdependent content: different areas of content are interdependent
- Sequencing activity types: how to sequence the types of activities students engage with rather than the content itself
- Maximizing other objectives

There are many kinds of personalised learning path recommendation algorithms. See [52] and tables therein for a detailed view on the works using the content classification above. In this section, we take a more technology orientated view providing a classification framework for technologies employed in sequencing.

[1] See https://fs.blog/2018/12/spacing-effect/

3.1.1 Expert-based

Existing spaced repetition systems rely on heuristics for review scheduling. The Leitner system[100, 147] uses a network of first-in-first-out queues to coarsely prioritise items by novelty and difficulty.

Pimsleur [140], SuperMemo [197], Anki [59], and Mnemosyne [1] use layers of handcrafted rules to decide when to next review an item and to prioritise items within a session. [122] use a threshold-based policy that selects the item with predicted recall likelihood closest to some fixed threshold.

3.1.2 Automated Learning Path Extraction

Hill climb search takes a greedy approach, starting with any sequence and then evaluating neighbouring sequences to see if this one is fitter, stopping when no more neighbouring sequences are available [163].

[103, 4] apply constraint optimisation algorithms focusing on restrictions such as learning time, learning style, and learning difficulty, thus proposed a universal solution for the learning path. For instance, [103] build on rule mining association with topological sorting of the related rule sets extracted from student test records and skills to test associations. These sorted subgraphs correspond to learning paths with different learning features.

DKT has been used to sort out sequences of skills depending on skill mastery [209]. The authors create pairs of skills and use DKT to predict the performance of each pair of exercises. Kahn's algorithm to generate a topological order from the directed acyclic graph created by selecting candidate skills from the predicted performance of the skills latent to each exercise.

Sequencing based on matrix factorisation finding latent features for the skills needed to do the activity and the skills of the students [158]. [95] also build on recommender system techniques to propose a self-directed learner model.

[175]bidirectional LSTM to learn each exercise representation from its text description without any expertise and in-formation loss. Then, we propose a new LSTM architecture to trace student states (i.e., knowledge states) in their sequential exercising process with the combination of exercise representations. For making final predictions, we design two strategies under EERNN, i.e.,EERNNM with Markov property and EERNNA with Attention mechanism

[45] Bayesian Knowledge Tracing (BKT) model that incorporates partial credit scores, reasoning about multiple attempts to solve problems, and integrating item difficulty.

[135] suggested that mastery learning tends to be threshold-dependent and it is very difficult to automatically define an optimal threshold value. The DKT and DKVMN have been suggested to be more useful for curriculum sequencing [156].

3.1.3 Reinforcement Learning Scheduling

Reinforcement Learning (RL) models can learn complex relationships between course activities, learner actions, and educational outcomes.

Reference	*Model*
[12, 106, 192, 38, 162, 157, 82, 215]	Model-free
[167, 215]	MDP
[107]	DASH
[111, 166, 144]	POMDP
[22, 155, 9]	(Deep) neural net
[106]	Genetic algorithms for mutating a seed set of environmental rules

Table 9: A Summary of Model-Free vs Model-based RL Techniques

At each time step t, an agent observes E in state s_t; it chooses an action a_t from a discrete set of possible actions; and E provides a scalar reward r_t and evolves into next state s_{t+1}.

Future rewards are discounted by $\gamma \in [0, 1]$ so that the return at time-step t is defined as $R_t = \sum_{t'=t}^{T} \gamma^{t'-t} r_t$, where T is the last time-step in the episode. The goal is to maximise R_t, which is equivalent to finding the optimal action-value function $Q(s, a)$ for all states. It can be calculated as $Q(s, a) = max_{\pi}[R_t | s_t = s, a_t = a, \pi]$ and $Q(s, a)$ must follow the Bellman Equation, π being an optimal policy.

Table 9 shows the main types of model-based and model-free techniques used for content sequencing. As can be observed, there is an increasing number of works relying on model-free approaches to cope with finding optimal policies in complex environments (including deep Q-learning). We will divide this subsection in Online and Offline algorithms as the impact of not having initial data may lead to frustration in learners and less future engagement with the learning system.

Offline RL

Dynamic Bayes Net (DBN) have been used to adapt content design as a sequential decision making problem, where an optimised policy maps states of student knowledge to the most effective lesson (problem or hint) to pose next. [70] show how this method performs on par with hand-tuned expert policies.

As early as 20+ years ago there were already works in this direction. [12] showed how TD(0)-based reinforcement learning was useful to derive a teaching policy that meets the specified educational goal.

Students that experience a sense of control over their own learning. Sanz et al. use deep reinforcement learning to decide whether to expose students with a worked example or a problem solving task at any point in time. They showed these two options are not radically different to students choosing freely [155]. [166] follow a similar process by using constrained action-based partially observable Markov decision processes (POMDP) and optimising for learning rate and time for high performing and low performing students.

[9] schedule educational activities in real time for a large online course through active learning. The authors used a controlled experiment with over

1000 learners to show the learnt policy delivers learning gains using fewer educational activities and with lower dropout rates. This means they chose a reward function that prioritises performance and penalises the inclusion of additional activities.

[22] use a fully connected neural network with a final softmax layer returning the level of mastery of each available KC. Assuming N KCs, the input is a N-dimensional vector that represents the user learning state and there is a value associated to the degree of the mastery of the KC, and the value range is [0,1]. The reward function is modelled as the difference between the learner's mastery degree of two adjacent recommended KCs.

[53] provide a method to help determine when a policy suggesting sequence of contents should or should not be deployed. [192] offer a control theoretic view on how the next best sequence can be computed. Their model is not numerical so can scale and compute faster.

Representing the state space like in DKT (one-hot encoding for each combination of educational activity and score within a learner trace) resulted in a huge state space so the authors simply ignored the ordering of previous activities.

RL algorithms require hundreds of steps to learn simple policies which can make them not ideal for learning courses with just a few thousand learners. [9] propose to use proximal policy optimisation (PPO), a policy gradient method that leverages deep neural networks to reduce the dimensionality of the state/action space and also the number of samples an algorithm needs to converge [160]. PPO is an actor-critic algorithm, where the actor (a neural net itself) decides the action to take and the critic (another neural net) estimates the expected reward from an action on current state.

[167] show how more efficient policies can be beneficial for students who need extra help and effort.

Online RL

Some methods do not require a wealth of data (given reasonable priors) to start working and can, hence, be more effective and avoid the “cold start” problem that plagues many smart learning systems [189].

In multi-armed bandit (MAB) scenarios, agents solve an online decision problem in which they learn to maximise a reward over time based on experience.

At each step, the agent chooses one of a finite set of actions A, and receives reward r which is related to the chosen action alone and is independently and identically distributed (IID) given the action.

The policy to choose the action is aimed at minimising expected regret (difference between using an exploratory policy and always choosing the best action).

Clement et al. used expert initialisation of a MAB algorithm called EXP4, which highlighted what activities were at the right level to be presented to the students [38].

[162] used MABs to personalise the selection of the next question to pose to the student. Given a set of target questions they estimate the expected learning gains for each question and use an exploration-exploitation strategy to choose

the best next question to ask. This way, they maintain a personalised ranking over the difficulties of each question and update the ranking immediately based on learner's progress.

[144] review the literature to show that MAB experiments lead to higher average benefits to students than traditional experimental designs, provided there are twice as many participants to keep the needed statistical power. They also show how longer MAB experiments assign fewer students to a less effective condition than typical practice (short experiment followed by applying the winning strategy to all future students).

Rafferty et al. also conclude that MAB assignments increase false positive rates, "especially if there are temporal biases in when students enter the experiment", which can happen when students can choose when to participate in the experiment [144].

Model-free reinforcement learning has also been explored to develop a review scheduling algorithm that does not explicitly model the student and learns a policy from raw data of student interactions [148]. In a similar model-less approach, Sarma and Ravindran and Iglesias et al. used an artificial neural network to build a Q function off student interactions with the learning environment [157, 82, 215]. These works rely on the Q-learning algorithms to estimate the values of state-action pairs (the optimal action from any state is the one with the highest Q-value). Martin and Arroyo follow a V-value function approach where they detect policy stability periodically and recalculate values when needed [113].

Many of the previous systems rely on simulations or a wealth of available data. Genetic-Based Reinforcement Learning (GBML) can learn a set of rules from the environment based on a genetic-based optimiser for rule discovery [106].

3.1.4 Automated Difficulty Recognition

Automatic difficulty recognition from low-level features are classification methods like SVMs, decision trees or ensembles of those [83]. These authors define an layered stacking of SVMs of methods for task sequencing building on the perceived difficulty of the task, which is calculated based on low-level features extracted from a log-file of interactions (without explicit user feedback, like in [150]. This makes extracting difficulty easier and minimises cost in sensors, privacy aspects with adding mikes or cameras and enables learning from anywhere [195].

3.2 Transfer Learning

A domain D consists of two components: a feature space X and a marginal probability distribution $Pr(X)$ where $X = x_{i}{}_{i=1}^{n} \subset X$. Given a specific domain, a task T consists of two components: a label space Y and a conditional probability distribution $Pr(Y|X)$ where $Y = y_{i}{}_{i=1}^{n} \subset Y$. Considering a source course S and target course T, the feature spaces of the source and target domains DS and DT are the same but the feature distributions are different $Pr(XS), Pr(XT)$.

However, the prediction tasks TS and TT for the two domains are the same, as the conditional distributions coincide $Pr(XS|YS) = Pr(XT|YT)$. In transfer learning the training of the target predictive function $f_T()$ in the target domain DT is supplemented using the knowledge in the source domain DS, where $DS \neq DT$ and $TS = TT$. At training time, source data $xS_i, yS_{inS}^{i=1}$ and the unlabeled target domain features $XT = xT_{inT}^{i=1}$ are available.

There are 3 approaches for transductive feature learning:

- instance-based methods motivated by importance sampling: try to reweight samples in the source domain which are then used in the target domain for training in an attempt to correct for marginal distribution differences
- representation-based methods: transforming one or both of the domains to a common latent feature space — usually of a low dimension — that has predictive qualities while reducing the marginal distribution between the domains
- parameter-based methods: try to transfer knowledge through the shared parameters of the source and target domain learner models

[49] try to predict dropout levels by using 3 techniques: 1) auto-encoders, 2) a passive approach using transductive principal component analysis, and 3) an active approach that uses a correlation alignment loss term. These models learn a representation space that is common to both source and target domains which can then capture the common characteristics of the two distributions by simply creating a deep representation from raw click stream data. In a similar study, [188] defined a set of features based on the event logs of two MOOC systems and explored dropout prediction by applying a learning model built on one system to the other system and blending data from both systems. [154] employ TL to detect emotions expressed by learners on video lessons. They learn affect representation by training a ResNet-50 on a facial affect classification AffectNet and extracting a fixed-size embedding and compressing it into a lower-dimension by learning a fully-connected neural network layer. This compact representation is concatenated with embeddings of head pose and gaze, and fed into another fully-connected neural network.

Hunt et al. [80] predict user graduation rates using a TL-enabled version of AdaBoost as an example of instance-based TL. Also in the instance-based set of methods, [30] also use LSTM auto-encoders and use minimisation of the domain-specific knowledge state distribution discrepancy under maximum mean discrepancy (MMD) measurement to then fine-tune weights. [77] showed how case-based reasoning to create a shared feature space could be used in conjunction to 4 different instance-based TL algorithms (MultiSource,TrAdaboost, TrAdaboost, and TransferBoost).

Domain adaptation (DA) is a particular case of transfer learning (TL) which utilises labeled data in one or more relevant source domains to execute new tasks

Type of TL	*References*
representation based	[49]
instance based	[30, 80, 77, 207]
parameter-based	[19, 110, 182, 188, 154]

Table 10: A Summary of Transfer Learning Techniques

in target domains. All DA settings share a commonality: the label set in the source domain is the same as the label set in the target domain.

[207]adjusts a classifier trained on the source domain (relying on dissimilar samples of the source domain) to the target domain (based on the most confident examples of the target domain).

With regards to parameter-based methods, Boyer and Veeramachaneni show how courses might evolve differently over time, even if they have context and structure and this is due to courses having different students and instructors [19]. They also note the existence of features that cannot be transferred. [19] ensembled different methods that mix predictive models built on different data sources. [110] use a refinement of naive TL, where they study the portability of performance prediction models within blended university courses by training a classifier per dataset and then testing if the trained model is applicable to other courses in the same degree or courses that exhibit similar user behaviour in Moodle.

[182] form pairs of datasets and then train a deep neural network on the first dataset of each pair and, subsequently, it is applied on the second dataset of the pair for further training after a predefined number of epochs.

Table 10 shows the works that have attempted to use TL in the educational domain and the type of techniques they used. As can be observed,

3.3 Hints and Feedback

Unlike sequential activities typical of MOOCs, ITS let some degrees of freedoms for students or teachers to select the next activity. Manual guides, on-line systems to personalised assistance from academic advisers are classic tools to inform the decision on when and how to offer hints and feedback. Content-specific hint systems can be very specialised and can leverage content-specific features for hint suggestion (see [116] for example), we will try to keep this section as general as possible.

Automatic hints help students reduce the time spent on problems as they were guided to prevent common pitfalls or poor strategies [173]. A hint is generally deemed to be more efficient if it reduces the time it takes the student to learn the material. Different types of hints take different amount of time to deliver and takes the student a different amount. The most effective feedback for learners should be credible, specific (not evaluative), and infrequent, but delivered at the right time [169]. Hints tend to be ignored by students [39].

[155] suggesting that simpler explanations of the pedagogical policy can improve learning performance. This is a list of example of hints suggested by a number of systems in the space:

- *Hazard hints*: messages indicating that most students who attempt the same are not successful have proven to be helpful in guiding students [62].
- *Critical timing*: Kaplan-Meier survival estimators show several declining steps that can point at critical times for intervention [56].
- *Sequential recommendations and repetition intervals*: [212] use a simple autoencoder to incorporate auxiliary learner attributes and feeds a basic attention module to factor the most important features from learner attributes and interactions for performance prediction and next best learning activity recommendations. The authors find Top N similar learners who have achieved mastery and then identify the steps taken from those similar learners to achieve mastery. This allows the system to extract and sort the next learning activity. Similarly, [105] design sequential interventions on top of BKT. Some of these interventions can be as simple as eliciting a next step from the student (vs telling her directly) or ask her to justify her reasoning. [32] build a reinforcement learning system to assess which of these two strategies works best in the wild.
- *Content specific*: [172, 37, 116] add application-specific messages to guide students through common pitfalls
- *Content review recommendations*: guide students to (1) topics to review, (2) similar questions to attempt and (3) easier questions [51].
- *Edit recommendations*: a type of contents specific types (e.g. in programming an abstract syntax tree distance measure can be used to provide specific changes) [218] or calculating edits made by successful students in similar states [128, 71].

The Hint Factory takes prior student transaction to create a Markov decision processes (MDP) from interaction networks, where vertices are ob-served student problem-solving states (snapshots of an on-going or completed proof) and assign scores to problem-solving states [172]. The continuous hint factory calculates the distance between the student's trajectory and those of all students who found a correct solution, using dynamic time warping (DTW). Then, the desired state is found using Gaussian Process Regression on the DTW distances and it then employs a tree grammar to select only valid edits for a given programming language [128].

An alternative approach would be about directly learning hints from a group of experts. [139] took a group of seven computer science educators to label hundreds of student programs with their suggested partial solution for the student to move towards used as a gold standard for comparison with a number of alternatives. These labels could have also be used to train a model to make suggestions made on them.

Beside the types of hints, these feedback mechanisms can also be classified according to the approach they take with regards to figuring out the hints.

- *Wisdom of the crowd techniques*: the action taken by a majority of successful learners that are similar to a learner is the most appropriate one [212, 172, 37, 218, 71]. These include clustering solutions [79, 72, 117], statistical maximisation of the probability of success by finding likely sequences [219, 27], and hint factory-based approaches [172, 128]
- *Desirable path*: creating a graph of actions where nodes represent intermediate and edges denote single changes/actions and calculating the shortest path. This can be calculated in a number of ways such as assuming smallest expected time (Poisson distribution of time events), Markov-zero most likely paths [139].
- *Goal Chasing*: a goal is set and the system finds the best next recommendation. For instance, time to complete the assignment may be one driver and the recommendation for this goal and, say, long-term memory consolidation, would differ.
- *Learning in embedding space*: compactly represent the space of errors and solutions and can provide hints to students with trajectories that have never been observed [217, 138, 128]

The use of historical student data (wisdom of the crowd and desirable path techniques) means that future students would need to stay on previously-seen paths and students learning off mainstream paths would not receive hints, even if they were not particularly far from a solution. One could infer what hints to give to students learning off common paths by clustering original steps in the path into equivalent groups and offering hints relative to the cluster any new observed step is closer to.

In a similar vein, [79, 72] have tried to infer clusters to propagate hints relative to a most common correct solution. [114] cluster students into learning levels and use reinforcement learning to choose subsequences of all possible hints for a problem. Some authors propose a semi-supervised technique for hint generation that clusters answers; instructors then have to manually label one correct answer per cluster [87]. Clustering-based feedback is not very good at offering an idea on the next best action that is personalised to an individual learner [149].

Wisdom of the crowd and desirable path techniques do not scale well as the number of potential intermediate results and actions/hints/edits grow. They then require exponentially more data to cope with the increased solution space to find appropriate hints [57]. The continuous hint factory [128] learns a regression function as the hint policy which can identify the most likely hint as a vector in an embedding space and then translates this vector back into a human-readable edit.

Embedding learning techniques as less data hungry than other approaches but still fall short when student trajectories are very different from the norm. [57] propose a reinforcement-learning approach to cope with first students in a course. [217] evaluated RL for determining the best course of interventions on a RL simulated students. They explored student learning speed based on: 1) no advice, 2) early advice, 3) mistake correcting or 4) a RL teacher that also adapts and learns. MDPs have been proposed to learn an intervention policy highlighting the most effective tutor turn-taking behaviours [120]. Mitchell et al. learned key design considerations about intervention timing: maintaining tutor engagement during student problem solving and avoiding multiple consecutive interventions.

Recommendations may need to change as the course or as a the course is repeated over the years and different cohorts take it. [194] introduced an LSTM to account for concept drift in recommendations of next activities in sequential courses.

Bayesian inverse planning takes as input a set of step-by-step actions from a learner and returns a posterior distribution over possible levels of mastery for a set of skills. Inverse planning treats algebraic equation solving as a MDP, in which people choose actions to bring them closer to the goal of solving an equation with as few steps as possible.

[145] showed that completing feedback obtained via Bayesian inverse planning was associated with performance improvements, but personalised feedback was not associated with reliably more improvement.

3.3.1 Affective Learning

Affective learning aims to teach learners by highlighting the importance of effort and perseverance and the idea that intelligence is malleable rather than a fixed trait. Affect and mental state of the learner influence the learning process: positive affective states (such as surprise, satisfaction or curiosity) are known to contribute towards learning, negative affective states (e.g. frustration) can be detrimental for learning [68].

Emotion, engagement or affect can be identified in speech by processing distinct linguistic features, like n-grams and bag-of-words, and low-level features like prosodic features or disfluencies [13, 84, 83]. The advantage of using low-level features is that words are not needed and they can be simpler to interpret.

Capturing all actions of the students interacting with the system (see e.g. [115]) like features extracted from a log-file, eye tracking [40, 83], body gestures [41], gaze to predict boredom [85], face expression or head movement to predict disengagement or frustration [196], keystroke dynamics identified by analysing the rhythm of the typing patterns of persons on a keyboard [60], or information from physiological sensors (e.g. skin conductance is related to high energy) as in [165, 5, 196].

[69] adopt a graph-based approach to modelling, querying and visualising student-system interactions where nodes are odes represent occurrences of key indicators that are detected and edges point to the next event. They applied

association rule mining to reason about the sequences of events in the graph.

Once current pre-dispositon towards learning has been identified, the next step is to identify how to respond to improve motivation and learning. Computer agents do affect motivation [196].

[68] - adapt type of feedback and presentation based on state. The types of affective actions taken are:

1. AFFECT BOOSTS
2. AFFIRMATION prompts
3. INSTRUCTIVE feedback
4. OTHER PROBLEM SOLVING
5. REFLECTIVE prompts
6. TALK ALOUD prompts
7. TASK SEQUENCE prompts

List 1: An Example Set of Affective Actions [69]

Woolf et al. [196] test the actions in List 2 to determine gender and performance differences that affect the response to affective feedback (women more susceptible and low performing students being less susceptible).

As can be observed in Table 11, there are many studies that detect emotions based on one or several type of learner input, but most of them fail to use that to produce actions that could result in improved learner performance oe experience.

1. Attribution interventions: learner motivation is directly rooted in their beliefs about why they succeed or fail. Sent before tasks are started.
2. Effort affirmation: acknowledge effort after students obtain a correct solution.
3. Strategic Interventions: making students more effective problem solvers and motivating them for learning in general

List 2: Overview of Affective Actions [196]

It seems like creating companions that trust and support learners have a deep impact on self-efficacy beliefs [118]. This is especially true when an established relationship between virtual companion and the student is formed.

4 Assessment Assistance

Automated essay scoring (AES) is a compelling topic in Learning Analytics (LA) for the primary reason that recent advances in AI find it as a good testbed to explore artificial supplementation of human creativity. The key metrics of performance in this task are Pearson's correlation, Spearman's correlation, Kendall's Tau, and quadratic weighted Kappa (QWK).

Reference	*Emotion Detection Model*	*Action*
[69]	Graph + association rule discovery	As in List 1
[85]	Random Forests (RF), Naïve Bayes, Logistic Regression, and Support Vector Machines	NA
[68]	dynamic Bayesian networks	As in List 1
[115, 41]	Bayesian network	NA
[13]	non-linear regression function	NA
[84]	support vector machine	NA
[196, 5]	stepwise regression, Markov chains	As in List 2
[60]	decision trees	NA
[165]	support vector machine and K-nearest neighbors	NA
[86]	several classifiers	NA

Table 11: A Summary of Affective Models and Actions Taken as a Consequence of Detecting the Affective Emotion.

Before the mid 2010s, many works were relying on carefully handcrafted features for AES [178]. See [81] for a comprehensive review including earlier works.

Zhao et al. rely on a fully connected forward neural network augmented with external memory keeping a map between works with a given score and the score they were marked with [214].

[17] used a collection of ensemble methods (including deep neural networks) to determine whether assessment tasks can be automated. They showed considerable divergences between rubic and holistic scores and also between human and machine scores. The best performance is sometimes obtained with a shallow network including just two layers [18]. This is similar to the results by Nguyen and Dery (2020) who showed how a simple neural network model using 300 dimensional Glove as initialisation to the embedding layer offered better results than deep LSTM models [126].

Other authors show how more complex models perform better, such as convolutional recurrent networks [178, 21]. Uria et al. (2019) showed how transformer-based architectures (BERT and XLNET) outperform simple BOW-based or recurrent models [151].

Kumar and Boulanger (2020) use Shapley Additive Explanations (SHAP) to try to understand how some models make their rubic score predictions [94].

Uto and Okano point out that having datasets where the training data has been scored by biased raters makes deep neural models very prone to poor performance [183]. They propose to integrate IRT models to deal with rater bias within training data and compare this model against a convolutional recurrent model and BERT.

Discourse structure and coherence important aspects of essay writing and are of-ten explicitly a part of grading rubrics. Handcrafted features are being incorporated into neural models to improve performance (see [123, 184]) .

5 Ethics

The European Union General Data Protection Regulation (GDPR) became law in 2018 building in a wealth of prior legislation and new trends. Being a General law, it is applied across industries and, therefore, the ethics of learning analytics are not explicitly covered there.

Non Governmental Organisations (NGOs), such as DataEthics[2] are also supporting conferences and standards for ethical considerations in data analyses. Other organisations are also aligning into Data Ethics standards (e.g. use of personal data in AI [2]).

In 2015 in the UK, the Joint Information Systems Committee (JISC) released a Code of practice for learning analytics. This code provides a set of recommendations to enhance existing information management processes to ensure the responsible, effective, legal, and ethical use of learning analytics [161].

The elements in Table 12 above are very formal and miss some important considerations: [75] builds on GDPR and believes data ethics is not just a checklist "formally framed in countless statements,documents and mission statements from a multitude of sources, including governments,intergovernmental organisations, consultancy firms, companies, non-governmental organisations, independent experts and academics" but also part of an ongoing informal conversation, cultural positioning, and an ethos that is part of the day to day life of the organisation.

UNICEF highlights the potential for "severe, long-lasting and differential impacts on children, child rights need to be firmly integrated onto the agendas of global debates about ethics and data science". The classical elements of consent in the list and table above are less applicable to children [15].

These considerations should be governing design and data capture decisions for children in their different emotional development stages. For instance: "We excluded the possibility of collecting an exploratory corpus because making practice with very easy and very difficult exercises in random order could be frustrating for the students, who could be children" [158].

A truly ethical data framework for learning analytics for children would need to consider the implications of errors and outliers, dealing with their individual needs[15].

There are two key elements that converge to make Ethical Frameworks based on checklists effective for children learning[15]:

[2] https://dataethics.eu

Paper	Consent and Purpose	Transparency	Privacy	Data Quality	Ownership
[141]	“clear and accurate information is provided about what data is or may be collected, why and how it is collected, how it is stored and how it is used; and (b) agreement is freely given to the practice(s) described.”	“regarding the purposes for which data will be used, under what conditions, access to data, and the protection of an individual’s identity”	As in [54] below	Refers to the now withdrawn UK Cabinet office recommendations for ethical data science[3], which itself points to the need to robust data to feed models	As in [54] below
[54]	Explicit, clear, concise, inconsequential	Explain, legitimate, involve users in your data processes	“Privacy should not been seen as a burden but rather as a valuable service we can offer to build trusting relationships with our stakeholders”, Anonymise, Aggregate	NA	“ask for consent, and to strictly monitor who has access to data”
[15]	“The EU GDPR states that children deserve specific protection of their personal data and calls for parental consent for the processing of personal data of children under 16. Also, children ‘evolving capacities’ with regards to informed consent tend not to be considered appropriately”		“given the current level of uncertainty around insuring continued data privacy and the particular vulnerabilities of children, we need to err on the side of caution with data generated on and by children”	NA	Data providers, collectors, analysers, users
[161]	“Collection and use of data for these may require further measures, such as data protection impact assessments and obtaining additional consent. Options for granting consent must be clear and meaningful, explaining the expected consequences of granting or withholding consent. Students should be able easily to amend their decisions subsequently”	“Institutions should clearly describe the processes involved in producing the analytics to students and staff”	“Avoid the identification of individuals from metadata or re-identification of individuals by aggregating multiple data sources”	“Inaccuracies in the data are understood and minimised. The implications of incomplete datasets are understood. The optimum range of data sources is selected. Spurious correlations are avoided”	“Students have a legal right to be able to correct inaccurate personal data held about themselves”
[43]	“clear and transparent information on the purposes for data collection so that they are in a position to give informed consent. They should also potentially be provided with the option to opt-out”	“what data is collected, the purposes of the data collection and how the data will be used, as well as how the data are processed, stored and share”	“self-determination in that individuals are imbued with the capacity to determine their level of privacy or disclosure of personal information”	NA	“data belongs to the owner of the data collection tool [who is], typically also the data client and beneficiary”

Table 12: A Summary of the Main Data Ethics Frameworks Applied to Learning Analytics

- *technical developments*: e.g. real time nature of actions and hints exposed to students do not lent themselves to the rigorous scrutiny and the need to deal with statistical anomalies in analyses in a constructive and tactful manner.
- *inefficiency of traditionally privacy techniques*: individual explicit consent opting out and anonymity, have lost much of their effectiveness.
- *long term impact* of decisions made on children who are the first generations to have a a digital life from their early infancy and the unknowns about technology evolution will have on data being collected today. And the role of families, parents and tutors in this needs to be clearly established.

6 Public Datasets

Public datasets are available like ASSISTments [61] and Bridge and Algebra [4]. They are composed by log files of ITS, i.e. recordings of the score of a student in the tasks he or she accomplished, number of hints and domain information. Example of domain information are the domain of knowledge (fractions, additions, etc.), number of skills required to solve the exercise and other information necessary in order to individuate an univoque step, if we are talking of a multiple step tasks.

There are nine classic real-world datasets commonly used as benchmarks: 4 datasets from the ASSISTment intelligent tutoring system, 2 datasets from the KDD Cup2010 EDM Challenge: Algebra I 2005-2006 and Bridge to Algebra 2006-2007, 1 dataset from middle-school students practicing Spanish, one from a college-level statics from the PSLC DataShop, and a dataset from middle-school students practicing math on the Squirrel AI tutoring system [63].

The the PSLC DataShop includes a single directory for data available in the pace[5].

The *algebra05, bridge06*, and *assistments09* datasets in terms of learners per item and learners per KC and deep learning methods, like DKT, severely overfit.

[63] also report the proportion of consecutive interactions involving consecutive item numbers (most datasets include a sequential item number) in the material is related to the degree to which learners progress sequentially through the material.

The number of interactions per learner also seems to affect learner performance [63]. Most datasets present a long-tail with up to a few hundreds of interactions for the typical student. But the *bridge06* and *spanish* contain thousands of interactions per learner. DKT performs worse than logistic regression, which uses historical counts as a feature, since it does not maintain long-term information [63].

[4] http://pslcdatashop.web.cmu.edu/KDDCup/rules_data_format.jsp

[5] https://pslcdatashop.web.cmu.edu/index.jsp?datasets=public

The Open University Learning Analytics Dataset (OULAD) comprises 22 courses for years 2013 and 2014 and 32593 students together with metadata such as student demographic details, assessment results, and daily interactions with the university's VLEs (10,655,280 entries)[6].

7 Discussion

In certain education problems the teacher has an educational goal that can be formulated as θ_*. For example, a botanist may want to teach students to categorise flowers according to the petal and sepal length. The botanist has the correct decision boundary, but she cannot transfer into the student's brain. Instead, she teaches by picking informative flower samples to show the students. *Machine teaching* (MT) aims to optimise the choice of flower samples to efficiently learn θ_*. Using MT, a teacher with full knowledge about the learning dynamics of the students can teach a target concept to the entire classroom using $O(mind, Nlog(1/\epsilon))$ examples, where d is the ambient dimension of the problem, N is the number of learners, and ϵ is an accuracy parameter [203].

female students respond more positively (more engaged and less frustrated) to empathetic feedback than male students; Currently there is no gold standard for either labeling a person's emotional state or for responding to it. One approach to recognizing emotion is to triangulate among three different inputs: sensor data, student self-reports, and human observation of students.

For user emotion modeling, researchers and developers widely refer to Russell's two-dimension "circumplex model of affect", where emotions are seen as combinations of arousal and valence. Another model, known as OCC (following their creators' name: Ortony, Clore, Collins), specifies 22 emotion categories based on emotional reactions to situations constructed either as a) goals of relevant events, b) actions of an accountable agent, or c) attitudes of attractive or unattractive objects.

Offering appropriate feedback to learners is not an easy task. A recent analysis has shown how many educational games for early-years learners are inconsistent and not proactive when providing error feedback, often promoting trial and error strategies, lack of support for learning the game mechanics, and a preference for task-oriented rewards [221].

While most studies on TL show some advantages to using transfer learning for predicting a number of features (e.g. dropout, graduation rates, performance), some experimental results indicated that the produced learning models did not always perform as intended. [19]

Li et al. show through a series of MAB simulations that adaptive personalisation can be a double-edged sword. On the one hand, real-time adaptation can provide better student experience, but on the other hand an intelligent learning system would take more time to adapt and there would be increased variability. Thus, more personalised models may not always be preferable [104].

[6] https://analyse.kmi.open.ac.uk/open_dataset

Table 13: Characteristics of Publicly Available Academic Datasets

Datasets	algebra05	bridge06	assist09	assist12	assist17	statistics	squirrel	spanish	assist15
Interactions	607025	1817476	278868	2711602	934638	189297	6003641	578726	658887
Learners	574	1146	3241	29018	1708	282	24500	182	14657
Items	173113	129236	17709	53086	3162	1223	20201	409	NA
KCs	112	493	124	265	102	98	742	221	100
Median KCs per Item	1.36	1.01	1.2	1	1.23	1	1	1	NA
Median Learners per Item	85	101	120	35	18	4	28	1	NA
Median Learners per KC	1	4	10	22	90	136	137	178	NA
Timestamps	Y	Y	N	Y	Y	N	Y	N	N
Median Days per Learner	84	162	NA	52	184	NA	30	NA	NA
Median Interactions per Learner	581	1373	32	59	489	635	154	2924	31

Table 14: Characteristics of Publicly Available Academic Datasets (continued)

Datasets	EdNet [36][7]	STAT F2011[8]	KDD 2010[9]	Kalboard 360[10]	Duolingo[11]	AICFE [29][12]	CriticalLang [74] [13]	PISA 15[14]
Interactions	131441538	190000	4420000	NA	TBD.	125726	tbd	tbd
Learners	784309	330	3287	500	TBD	2648	669498	19334
Items	13,169	1224	1379	NA	TBD	655	95	183
KCs	293.	81	899	NA	TBD	21	tbd	tbd
Median KCs per Item	TBD	TBD	TBD	NA	TBD	1.2	tbd	tbd
Median Learners per Item	TBD	TBD	TBD	NA	TBD	102	tbd	tbd
Median Learners per KC	TBD	TBD	TBD	NA	TBD	105	tbd	tbd
Timestamps	Y	TBD	TBD	NA	TBD	Y	tbd	tbd
Median Days per Learner	TBD	TBD	TBD	NA	TBD	1	tbd	tbd
Median Interactions per Learner	TBD	TBD	TBD	NA	TBD	. 43	tbd	tbd

Boulanger et al. more than 1,000 hand-graded essays per writing construct would be necessary to adequately train the predictive student models on automated essay scoring, provided that all score categories are equally or fairly represented in the sample dataset [16]. Cader et al. explore data augmentation techniques to improve the performance of assessment prediction models without requiring more data [20]. Despite the abandoning of all feature engineering that came with the advent of deep neural networks for assessment support, recent work is concatenating handcrafted features to intermediate representations to improve performance [184].

We expect to see an increasing number of works benefiting from MT in classroom settings.

[216]

Some concerns exist about whether students truly need hints and whether pushing the student towards the answer is the right objective as learning is often associated with persevering on difficult goals, rather than achieving right answers quickly.

References

[1] The mnemosyne project, 2006.

[2] Ieee p7006 - personal data ai agent working group, 2017.

[3] K. Abhinav, V. Subramanian, A. Dubey, Padmaraj Bhat, and Aditya Divakaruni Venkat. Lecore: A framework for modeling learner's preference. In *EDM*, 2018.

[4] N. V. Ánh, Nguyen Viet Ha, and Ho Sy Dam. Constructing a bayesian belief network to generate learning path in adaptive hypermedia system. *Journal of Computer Science and Cybernetics*, 24, 2012.

[5] Ivon Arroyo, Beverly Park Woolf, Winslow Burleson, Kasia Muldner, Dovan Rai, and Minghui Tai. A multimedia adaptive tutoring system for mathematics that addresses cognition, metacognition and affect. *Int. J. Artif. Intell. Educ.*, 24(4):387–426, 2014.

[6] Mehmet Cem Aytekin, Stefan Räbiger, and Yücel Saygin. Discovering the prerequisite relationships among instructional videos from subtitles. In Anna N. Rafferty, Jacob Whitehill, Cristóbal Romero, and Violetta Cavalli-Sforza, editors, *Proceedings of the 13th International Conference on Educational Data Mining, EDM 2020, Fully virtual conference, July 10-13, 2020*. International Educational Data Mining Society, 2020.

[7] Tiffany Barnes. Evaluation of the q-matrix method in understanding student logic proofs. In Geoff Sutcliffe and Randy Goebel, editors, *Proceedings of the Nineteenth International Florida Artificial Intelligence Research Society Conference, Melbourne Beach, Florida, USA, May 11-13, 2006*, pages 491–496. AAAI Press, 2006.

[8] Tiffany Barnes, Donald Bitzer, and Mladen Vouk. Experimental analysis of the q-matrix method in knowledge discovery. In *Proceedings of the 15th International Conference on Foundations of Intelligent Systems*, ISMIS'05, page 603–611, Berlin, Heidelberg, 2005. Springer-Verlag.

[9] Jonathan Bassen, Bharathan Balaji, Michael Schaarschmidt, Candace Thille, Jay Painter, Dawn Zimmaro, Alex Games, Ethan Fast, and John C. Mitchell. Reinforcement learning for the adaptive scheduling of educational activities. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, page 1–12, New York, NY, USA, 2020. Association for Computing Machinery.

[10] Peter Battaglia, Jessica Blake Chandler Hamrick, Victor Bapst, Alvaro Sanchez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Caglar Gulcehre, Francis Song, Andy Ballard, Justin Gilmer, George E. Dahl, Ashish Vaswani, Kelsey Allen, Charles Nash, Victoria Jayne Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matt Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. Relational inductive biases, deep learning, and graph networks. *arXiv*, 2018.

[11] J. Beck and B. Woolf. High-level student modeling with machine learning. In *Intelligent Tutoring Systems*, 2000.

[12] Joseph Beck, Beverly Park Woolf, and Carole R. Beal. Advisor: A machine learning architecture for intelligent tutor construction. In *AAAI/IAAI*, pages 552–557, 2000.

[13] Joseph E. Beck. Engagement tracing: Using response times to model student disengagement. In *Proceedings of the 2005 Conference on Artificial Intelligence in Education: Supporting Learning through Intelligent and Socially Informed Technology*, page 88–95, NLD, 2005. IOS Press.

[14] Yoav Bergner, S. Droschler, S. Rayyan, D. Seaton, G. Kortemeyer, and D. Pritchard. *Model-based collaborative filtering analysis of student response data: machine-learning item response theory.* 2012.

[15] Gabrielle Berman and Kerry Albright. Children and the data cycle: Rights and ethics in a big data world, 2017.

[16] David Boulanger and Vivekanandan Kumar. Deep learning in automated essay scoring. In Roger Nkambou, Roger Azevedo, and Julita Vassileva, editors, *Intelligent Tutoring Systems - 14th International Conference, ITS 2018, Montreal, QC, Canada, June 11-15, 2018, Proceedings*, volume 10858 of *Lecture Notes in Computer Science*, pages 294–299. Springer, 2018.

[17] David Boulanger and Vivekanandan Kumar. Shedding light on the automated essay scoring process. In Michel C. Desmarais, Collin F.

Lynch, Agathe Merceron, and Roger Nkambou, editors, *Proceedings of the 12th International Conference on Educational Data Mining, EDM 2019, Montréal, Canada, July 2-5, 2019*. International Educational Data Mining Society (IEDMS), 2019.

[18] David Boulanger and Vivekanandan Kumar. Shaped automated essay scoring: Explaining writing features' contributions to english writing organization. In Vivekanandan Kumar and Christos Troussas, editors, *Intelligent Tutoring Systems - 16th International Conference, ITS 2020, Athens, Greece, June 8-12, 2020, Proceedings*, volume 12149 of *Lecture Notes in Computer Science*, pages 68–78. Springer, 2020.

[19] Sebastien Boyer and Kalyan Veeramachaneni. Transfer learning for predictive models in massive open online courses. In Cristina Conati, Neil Heffernan, Antonija Mitrovic, and M. Felisa Verdejo, editors, *Artificial Intelligence in Education*, pages 54–63, Cham, 2015. Springer International Publishing.

[20] Andrzej Cader. The potential for the use of deep neural networks in e-learning student evaluation with new data augmentation method. In Ig Ibert Bittencourt, Mutlu Cukurova, Kasia Muldner, Rose Luckin, and Eva Millán, editors, *Artificial Intelligence in Education*, pages 37–42, Cham, 2020. Springer International Publishing.

[21] Changzhi Cai. Automatic essay scoring with recurrent neural network. In *Proceedings of the 3rd International Conference on High Performance Compilation, Computing and Communications*, HP3C '19, page 1–7, New York, NY, USA, 2019. Association for Computing Machinery.

[22] D. Cai, Y. Zhang, and B. Dai. Learning path recommendation based on knowledge tracing model and reinforcement learning. In *2019 IEEE 5th International Conference on Computer and Communications (ICCC)*, pages 1881–1885, 2019.

[23] Hao Cen, Kenneth Koedinger, and Brian Junker. Learning factors analysis – a general method for cognitive model evaluation and improvement. In *Proceedings of the 8th International Conference on Intelligent Tutoring Systems*, ITS'06, page 164–175, Berlin, Heidelberg, 2006. Springer-Verlag.

[24] Hao Cen, Kenneth Koedinger, and Brian Junker. Learning factors analysis – a general method for cognitive model evaluation and improvement. In *Proceedings of the 8th International Conference on Intelligent Tutoring Systems*, ITS'06, page 164–175, Berlin, Heidelberg, 2006. Springer-Verlag.

[25] Abdessamad Chanaa and Nour-Eddine El Faddouli. Predicting learners need for recommendation using dynamic graph-based knowledge tracing. In Ig Ibert Bittencourt, Mutlu Cukurova, Kasia Muldner, Rose Luckin, and Eva Millán, editors, *Artificial Intelligence in Education*, pages 49–53, Cham, 2020. Springer International Publishing.

[26] Devendra Singh Chaplot, Christopher MacLellan, Ruslan Salakhutdinov, and Kenneth R. Koedinger. Learning cognitive models using neural networks. *CoRR*, abs/1806.08065, 2018.

[27] R. Chaturvedi. Task-based example miner for intelligent tutoring systems. In *PhD Dissertation. University of Windsor*, 2016.

[28] Fu Chen and Ying Cui. LogCF: Deep Collaborative Filtering with Process Data for Enhanced Learning Outcome Modeling. *Journal of Educational Data Mining*, 12(4):66–99, December 2020.

[29] Penghe Chen, Yu Lu, Vincent W. Zheng, and Yang Pian. Prerequisite-driven deep knowledge tracing. In *IEEE International Conference on Data Mining, ICDM 2018, Singapore, November 17-20, 2018*, pages 39–48. IEEE Computer Society, 2018.

[30] Song Cheng, Qi Liu, and Enhong Chen. Domain adaption for knowledge tracing, 2020.

[31] Song Cheng, Qi Liu, Enhong Chen, Zai Huang, Zhenya Huang, Yiying Chen, Haiping Ma, and Guoping Hu. Dirt: Deep learning enhanced item response theory for cognitive diagnosis. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, page 2397–2400, New York, NY, USA, 2019. Association for Computing Machinery.

[32] Min Chi, Kurt VanLehn, and Diane J. Litman. Do micro-level tutorial decisions matter: Applying reinforcement learning to induce pedagogical tutorial tactics. In Vincent Aleven, Judy Kay, and Jack Mostow, editors, *Intelligent Tutoring Systems (1)*, volume 6094 of *Lecture Notes in Computer Science*, pages 224–234. Springer, 2010.

[33] Benoît Choffin, Fabrice Popineau, Y. Bourda, and J. Vie. Das3h: Modeling student learning and forgetting for optimally scheduling distributed practice of skills. *ArXiv*, abs/1905.06873, 2019.

[34] Youngduck Choi, Youngnam Lee, Junghyun Cho, Jineon Baek, Byungsoo Kim, Yeongmin Cha, Dongmin Shin, Chan Bae, and Jaewe Heo. Towards an appropriate query, key, and value computation for knowledge tracing, 2020.

[35] Youngduck Choi, Youngnam Lee, Junghyun Cho, Jineon Baek, Dongmin Shin, Seewoo Lee, Jonghun Shin, Chan Bae, Byungsoo Kim, and Jaewe Heo. Assessment modeling: Fundamental pre-training tasks for interactive educational systems, 2020.

[36] Youngduck Choi, Youngnam Lee, Dongmin Shin, Junghyun Cho, Seoyon Park, Seewoo Lee, Jineon Baek, Chan Bae, Byungsoo Kim, and Jaewe Heo. Ednet: A large-scale hierarchical dataset in education. In Ig Ibert Bittencourt, Mutlu Cukurova, Kasia Muldner, Rose Luckin, and Eva

Millán, editors, *Artificial Intelligence in Education*, pages 69–73, Cham, 2020. Springer International Publishing.

[37] Sammi Chow, Kalina Yacef, Irena Koprinska, and James Curran. Automated data-driven hints for computer programming students. In *Adjunct Publication of the 25th Conference on User Modeling, Adaptation and Personalization*, UMAP '17, page 5–10, New York, NY, USA, 2017. Association for Computing Machinery.

[38] Benjamin Clément, Didier Roy, Pierre-Yves Oudeyer, and Manuel Lopes. Multi-Armed Bandits for Intelligent Tutoring Systems. *Journal of Educational Data Mining*, 7(2):20–48, June 2015.

[39] Cristina Conati, Natasha Jaques, and Mary Muir. Understanding attention to adaptive hints in educational games: An eye-tracking study. *Int. J. Artif. Intell. Educ.*, 23(1-4):136–161, 2013.

[40] Cristina Conati, Sébastien Lallé, Md. Abed Rahman, and Dereck Toker. Comparing and combining interaction data and eye-tracking data for the real-time prediction of user cognitive abilities in visualization tasks. *ACM Trans. Interact. Intell. Syst.*, 10(2):12:1–12:41, 2020.

[41] Cristina Conati and Heather Maclaren. Empirically building and evaluating a probabilistic model of user affect. *User Model. User Adapt. Interact.*, 19(3):267–303, 2009.

[42] Albert T. Corbett and John R. Anderson. Knowledge tracing: Modelling the acquisition of procedural knowledge. *User Model. User-Adapt. Interact.*, 4(4):253–278, 1995.

[43] Linda Corrin, Gregor Kennedy, Sarah French, Simon Buckingham Shum, Kristy Kitto, Abelardo Pardo, Deborah West, Negin Mirriahi, and Cassandra Colvin. The ethics of learning analytics in australian higher education: Discussion paper, 2019.

[44] Antoine Cully and Yiannis Demiris. Online knowledge level tracking with data-driven student models and collaborative filtering. *IEEE Trans. Knowl. Data Eng.*, 32(10):2000–2013, 2020.

[45] Yossi Ben David, Avi Segal, and Ya'akov (Kobi) Gal. Sequencing educational content in classrooms using bayesian knowledge tracing. In *Proceedings of the Sixth International Conference on Learning Analytics amp; Knowledge*, LAK '16, page 354–363, New York, NY, USA, 2016. Association for Computing Machinery.

[46] Jimmy de la Torre. Dina model and parameter estimation: A didactic. *Journal of Educational and Behavioral Statistics*, 34(1):115–130, 2009.

[47] Michel C. Desmarais. Mapping question items to skills with non-negative matrix factorization. *SIGKDD Explor. Newsl.*, 13(2):30–36, May 2012.

[48] Michel C. Desmarais and Rhouma Naceur. A matrix factorization method for mapping items to skills and for enhancing expert-based q-matrices. In H. Chad Lane, Kalina Yacef, Jack Mostow, and Philip Pavlik, editors, *Artificial Intelligence in Education*, pages 441–450, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[49] Mucong Ding, Yanbang Wang, Erik Hemberg, and Una-May O'Reilly. Transfer learning using representation learning in massive open online courses. In *Proceedings of the 9th International Conference on Learning Analytics & Knowledge, LAK 2019, Tempe, AZ, USA, March 4-8, 2019*, pages 145–154. ACM, 2019.

[50] T.N. Doan and S. Sahebi. Rank-based tensor factorization for student performance prediction. *12th International Conference on Educational Data Mining (EDM).*

[51] A.K. Dominguez, Kalina Yacef, and J.R. Curran. Data mining for individualised hints in elearning. pages 91–100, 01 2010.

[52] Shayan Doroudi, V. Aleven, and Emma Brunskill. Where's the reward? *International Journal of Artificial Intelligence in Education*, 29:568–620, 2019.

[53] Shayan Doroudi, Vincent Aleven, and Emma Brunskill. Robust evaluation matrix: Towards a more principled offline exploration of instructional policies. In *Proceedings of the Fourth (2017) ACM Conference on Learning @ Scale*, L@S '17, page 3–12, New York, NY, USA, 2017. Association for Computing Machinery.

[54] Hendrik Drachsler and Wolfgang Greller. Privacy and analytics: It's a delicate issue a checklist for trusted learning analytics. In *Proceedings of the Sixth International Conference on Learning Analytics amp; Knowledge*, LAK '16, page 89–98, New York, NY, USA, 2016. Association for Computing Machinery.

[55] Guillaume Durand, Nabil Belacel, and Cyril Goutte. Evaluation of expert-based q-matrices predictive quality in matrix factorization models. In Gráinne Conole, Tomaz Klobucar, Christoph Rensing, Johannes Konert, and Élise Lavoué, editors, *Design for Teaching and Learning in a Networked World - 10th European Conference on Technology Enhanced Learning, EC-TEL 2015, Toledo, Spain, September 15-18, 2015, Proceedings*, volume 9307 of *Lecture Notes in Computer Science*, pages 56–69. Springer, 2015.

[56] Michael Eagle and Tiffany Barnes. Data-driven feedback beyond next-step hints. In John C. Stamper, Zachary A. Pardos, Manolis Mavrikis, and Bruce M. McLaren, editors, *Proceedings of the 7th International Conference on Educational Data Mining, EDM 2014, London, UK, July 4-7, 2014*, pages 444–446. International Educational Data Mining Society (IEDMS), 2014.

[57] Aleksandr Efremov, Ahana Ghosh, and Adish Singla. Zero-shot learning of hint policy via reinforcement learning and program synthesis. In Anna N. Rafferty, Jacob Whitehill, Cristóbal Romero, and Violetta Cavalli-Sforza, editors, *Proceedings of the 13th International Conference on Educational Data Mining, EDM 2020, Fully virtual conference, July 10-13, 2020*. International Educational Data Mining Society, 2020.

[58] Chaitanya Ekanadham and Yan Karklin. T-SKIRT: online estimation of student proficiency in an adaptive learning system. *CoRR*, abs/1702.04282, 2017.

[59] Damien Elmes. Anki, 2005.

[60] Clayton Epp, Michael Lippold, and Regan L. Mandryk. Identifying emotional states using keystroke dynamics. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, page 715–724, New York, NY, USA, 2011. Association for Computing Machinery.

[61] Mingyu Feng, Neil Heffernan, and Kenneth R Koedinger. Addressing the assessment challenge with an online system that tutors as it assesses, Jun 2009.

[62] Davide Fossati, Barbara Di Eugenio, Stellan Ohlsson, Christopher Brown, Lin Chen, and David Cosejo. I learn from you, you learn from me: How to make ilist learn from students. In *Proceedings of the 2009 Conference on Artificial Intelligence in Education: Building Learning Systems That Care: From Knowledge Representation to Affective Modelling*, page 491–498, NLD, 2009. IOS Press.

[63] T. Gervet, K. Koedinger, J. Schneider, and T. Mitchel. When is deep learning the best approach to knowledge tracing?". In *JEDM— Journal of Educational Data Mining- vol 12*, page 31–54, 2020.

[64] Aritra Ghosh, Neil Heffernan, and Andrew S. Lan. Context-aware attentive knowledge tracing. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery amp; Data Mining*, KDD '20, page 2330–2339, New York, NY, USA, 2020. Association for Computing Machinery.

[65] Yue Gong, Joseph E. Beck, and Neil T. Heffernan. Comparing knowledge tracing and performance factor analysis by using multiple model fitting procedures. In *Proceedings of the 10th International Conference on Intelligent Tutoring Systems - Volume Part I*, ITS'10, page 35–44, Berlin, Heidelberg, 2010. Springer-Verlag.

[66] Jose Gonzalez-Brenes and Yun Huang. Your model is predictive– but is it useful? theoretical and empirical considerations of a new paradigm for adaptive tutoring evaluation. In *The 8th International Conference on Educational Data Mining*, 2015.

[67] José P. González-Brenes. Modeling skill acquisition over time with sequence and topic modeling. In Guy Lebanon and S. V. N. Vishwanathan, editors, *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2015, San Diego, California, USA, May 9-12, 2015*, volume 38 of *JMLR Workshop and Conference Proceedings*. JMLR.org, 2015.

[68] Beate Grawemeyer, Manolis Mavrikis, Wayne Holmes, Sergio Gutiérrez-Santos, Michael Wiedmann, and Nikol Rummel. Affective learning: improving engagement and enhancing learning with affect-aware feedback. *User Modeling and User-Adapted Interaction*, 27(1):119–158, March 2017.

[69] Beate Grawemeyer, Alex Wollenschlaeger, Sergio Gutierrez-Santos, Wayne Holmes, Manolis Mavrikis, and Alexandra Poulovassilis. Using graph-based modelling to explore changes in students' affective states during exploratory learning tasks. In Xiangen Hu, Tiffany Barnes, Arnon Hershkovitz, and Luc Paquette, editors, *Proceedings of the 10th International Conference on Educational Data Mining, EDM 2017*, pages 382–383, United States, January 2017. International Educational Data Mining Society. 10th International Conference on Educational Data Mining, EDM 2017 ; Conference date: 25-06-2017 Through 28-06-2017.

[70] Derek Green, Thomas Walsh, Paul Cohen, and Yu-Han Chang. Learning a skill-teaching curriculum with dynamic bayes nets. volume 2, 01 2011.

[71] Sebastian Gross, Bassam Mokbel, Barbara Hammer, and Niels Pinkwart. How to select an example? a comparison of selection strategies in example-based learning. pages 340–347, 06 2014.

[72] Sebastian Gross, Xibin Zhu, Barbara Hammer, and Niels Pinkwart. Cluster based feedback provision strategies in intelligent tutoring systems. In *Proceedings of the 11th International Conference on Intelligent Tutoring Systems*, ITS'12, page 699–700, Berlin, Heidelberg, 2012. Springer-Verlag.

[73] Wilhelmiina Hämäläinen and Mikko Vinni. Comparison of machine learning methods for intelligent tutoring systems. In *Proceedings of the 8th International Conference on Intelligent Tutoring Systems*, ITS'06, page 525–534, Berlin, Heidelberg, 2006. Springer-Verlag.

[74] J. Hartshorne, B. Tenenbaum, J., and S. Pinker. A critical period for second language acquisition: Evidence from 2/3 million english speakers. *Cognition*, 177:263–277, 2018.

[75] Gry Hasselbalch. Making sense of data ethics. the powers behind the data ethics debate in european policymaking. *Internet Policy Review*, 8(2):1–19, 2019.

[76] Yu He, Xinying Hu, Zhongtian Xu, Sun, and Guangzhong. *KT-XL: A Knowledge Tracing Model for Predicting Learning Performance Based on*

Transformer-XL, page 175–179. Association for Computing Machinery, New York, NY, USA, 2020.

[77] Nguyen Duy Hoang, Vo Thi Ngoc Chau, and Hua Phung Nguyen. Combining transfer learning and co-training for student classification in an academic credit system. In Tru Cao and Yo-Sung Ho, editors, *2016 IEEE RIVF International Conference on Computing & Communication Technologies, Research, Innovation, and Vision for the Future, RIVF 2016, Hanoi, Vietnam, November 7-9, 2016*, pages 55–60. IEEE, 2016.

[78] Qian Hu and H. Rangwala. Academic performance estimation with attention-based graph convolutional networks. In *Proceedings of The 12th International Conference on Educational Data Mining*, volume abs/2001.00632, pages 69–78, 2019.

[79] Jonathan Huang, Chris Piech, Andy Nguyen, and J. Leonidas Guibas. Syntactic and functional variability of a million code submissions in a machine learning mooc. *AIED Workshops*, 2013.

[80] Xin J. Hunt, Ilknur Kaynar Kabu, and Jorge Silva. Transfer learning foreducation data. In *KDD Workshop*, 2017.

[81] Mohamed Abdellatif Hussein, Hesham Hassan, and Mohammad Nassef. Automated language essay scoring systems: A literature review. *PeerJ Computer Science*, 5:e208, 2019. n/a.

[82] A. Iglesias, Paloma Martínez, and F. Fernández. An experience applying reinforcement learning in a web-based adaptive and intelligent educational system. *Informatics Educ.*, 2:223–240, 2003.

[83] Ruth Janning, Carlotta Schatten, and Lars Schmidt-Thieme. Perceived task-difficulty recognition from log-file information for the use in adaptive intelligent tutoring systems. *Int. J. Artif. Intell. Educ.*, 26(3):855–876, 2016.

[84] Ruth Janning, Carlotta Schatten, Lars Schmidt-Thieme, Gerhard Backfried, and Norbert Pfannerer. An SVM plait for improving affect recognition in intelligent tutoring systems. In *26th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2014, Limassol, Cyprus, November 10-12, 2014*, pages 202–209. IEEE Computer Society, 2014.

[85] Natasha Jaques, Cristina Conati, Jason M. Harley, and Roger Azevedo. Predicting affect from gaze data during interaction with an intelligent tutoring system. In Stefan Trausan-Matu, Kristy Elizabeth Boyer, Martha Crosby, and Kitty Panourgia, editors, *Intelligent Tutoring Systems*, pages 29–38, Cham, 2014. Springer International Publishing.

[86] Ajjen Joshi, Danielle Allessio, John J. Magee, Jacob Whitehill, Ivon Arroyo, Beverly Park Woolf, Stan Sclaroff, and Margrit Betke. Affect-driven

learning outcomes prediction in intelligent tutoring systems. In *14th IEEE International Conference on Automatic Face & Gesture Recognition, FG 2019, Lille, France, May 14-18, 2019*, pages 1–5. IEEE, 2019.

[87] Shalini Kaleeswaran, Anirudh Santhiar, Aditya Kanade, and Sumit Gulwani. Semi-supervised verified feedback generation, 2016.

[88] Hamid Karimi, Tyler Derr, Jiangtao Huang, and Jiliang Tang. Online academic course performance prediction using relational graph convolutional neural network. In Anna N. Rafferty, Jacob Whitehill, Cristóbal Romero, and Violetta Cavalli-Sforza, editors, *Proceedings of the 13th International Conference on Educational Data Mining, EDM 2020, Fully virtual conference, July 10-13, 2020.* International Educational Data Mining Society, 2020.

[89] Tanja Kaser, Severin Klingler, Alexander G. Schwing, and Markus Gross. Dynamic bayesian networks for student modeling. *IEEE Trans. Learn. Technol.*, 10(4):450–462, October 2017.

[90] M. Khajah, Robert V. Lindsey, and M. Mozer. How deep is knowledge tracing? *ArXiv*, abs/1604.02416, 2016.

[91] Mohammad Khajah, Robert V. Lindsey, and Michael C. Mozer. How deep is knowledge tracing? In *Proceedings of the 9th International Conference on Educational Data Mining, EDM 2016, Raleigh, North Carolina, USA, June 29 - July 2, 2016*, 2016.

[92] Kenneth R. Koedinger, Albert T. Corbett, and Charles Perfetti. The knowledge-learning-instruction framework: Bridging the science-practice chasm to enhance robust student learning. *Cognitive Science*, 36(5):757–798, 2012.

[93] K.R. Koedinger, P.I. Pavlik, J. Stamper, T. Nixon, and S. Ritter. Avoiding problem selection thrashing with conjunctive knowledge tracing. In *EDM*, pages 91–108. www.educationaldatamining.org, 2010.

[94] Vivekanandan Kumar and David Boulanger. Explainable automated essay scoring: Deep learning really has pedagogical value. *Frontiers in Education*, 5:186, 2020.

[95] T B Lalitha and P S Sreeja. Personalised self-directed learning recommendation system. *Procedia Computer Science*, 171:583–592, 2020. Third International Conference on Computing and Network Communications (CoCoNet'19).

[96] Andrew S. Lan, Christoph Studer, and Richard G. Baraniuk. Time-varying learning and content analytics via sparse factor analysis. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, page 452–461, New York, NY, USA, 2014. Association for Computing Machinery.

[97] Andrew S. Lan, Christoph Studer, and Richard G. Baraniuk. Time-varying learning and content analytics via sparse factor analysis. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, page 452–461, New York, NY, USA, 2014. Association for Computing Machinery.

[98] Jinseok Lee and Dit-Yan Yeung. Knowledge query network for knowledge tracing: How knowledge interacts with skills. In *Proceedings of the 9th International Conference on Learning Analytics amp; Knowledge*, LAK19, page 491–500, New York, NY, USA, 2019. Association for Computing Machinery.

[99] Jung In Lee and Emma Brunskill. The impact on individualizing student models on necessary practice opportunities. In Kalina Yacef, Osmar R. Zaïane, Arnon Hershkovitz, Michael Yudelson, and John C. Stamper, editors, *EDM*, pages 118–125. www.educationaldatamining.org, 2012.

[100] S. Leitner and R. Totter. *So lernt man lernen.* Angewandte Lernpsychologie ein Weg zum Erfolg. Herder, 1972.

[101] Haotian Li, Huan Wei, Yong Wang, Yangqiu Song, and Huamin Qu. Peer-inspired student performance prediction in interactive online question pools with graph neural network. In *Proceedings of the 29th ACM International Conference on Information amp; Knowledge Management*, CIKM '20, page 2589–2596, New York, NY, USA, 2020. Association for Computing Machinery.

[102] Nan Li, Noboru Matsuda, William Cohen, and Kenneth Koedinger. Integrating representation learning and skill learning in a human-like intelligent agent. *Artificial Intelligence*, 219, 02 2015.

[103] Yancong Li, Zengzhen Shao, Xiao Wang, Xuechen Zhao, and Yanhui Guo. A concept map-based learning paths automatic generation algorithm for adaptive learning systems. *IEEE Access*, 7:245–255, 2019.

[104] Zhaobin Li, Luna Yee, Nathaniel Sauerberg, Irene Sakson, J. Williams, and Anna N. Rafferty. Getting too personal(ized): The importance of feature choice in online adaptive algorithms. In *EDM*, 2020.

[105] Chen Lin and Min Chi. Intervention-bkt: Incorporating instructional interventions into bayesian knowledge tracing. In *Proceedings of the 13th International Conference on Intelligent Tutoring Systems - Volume 9684*, ITS 2016, page 208–218, Berlin, Heidelberg, 2016. Springer-Verlag.

[106] Hsuan Ta Lin, Po Ming Lee, and Tzu-Chien Hsiao. Online pedagogical tutorial tactics optimization using genetic-based reinforcement learning. *The Scientific World Journal*, 2015, January 2015.

[107] Robert V. Lindsey, Jeffery D. Shroyer, Harold Pashler, and Michael C. Mozer. Improving students' long-term knowledge retention through personalized review. *Psychological Science*, 25(3):639–647, 2014. PMID: 24444515.

[108] Qi Liu, Zhenya Huang, Yu Yin, Enhong Chen, Hui Xiong, Yu Su, and Guoping Hu. EKT: exercise-aware knowledge tracing for student performance prediction. *IEEE Trans. Knowl. Data Eng.*, 33(1):100–115, 2021.

[109] Yu Lu, Deliang Wang, Qinggang Meng, and Penghe Chen. Towards interpretable deep learning models for knowledge tracing. In Ig Ibert Bittencourt, Mutlu Cukurova, Kasia Muldner, Rose Luckin, and Eva Millán, editors, *Artificial Intelligence in Education*, pages 185–190, Cham, 2020. Springer International Publishing.

[110] Javier López-Zambrano, Juan A. Lara, and Cristóbal Romero. Towards portability of models for predicting students' final performance in university courses starting from moodle logs. *Applied Sciences*, 10(1), 2020.

[111] Travis Mandel, Yun-En Liu, Sergey Levine, Emma Brunskill, and Zoran Popovic. Offline policy evaluation across representations with applications to educational games. In *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-Agent Systems*, AAMAS '14, page 1077–1084, Richland, SC, 2014. International Foundation for Autonomous Agents and Multiagent Systems.

[112] Ye Mao, Samiha Marwan, Thomas W. Price, Tiffany Barnes, and Min Chi. What time is it? student modeling needs to know. In Anna N. Rafferty, Jacob Whitehill, Cristóbal Romero, and Violetta Cavalli-Sforza, editors, *Proceedings of the 13th International Conference on Educational Data Mining, EDM 2020, Fully virtual conference, July 10-13, 2020.* International Educational Data Mining Society, 2020.

[113] Kimberly Martin and Ivon Arroyo. Agentx: Using reinforcement learning to improve the effectiveness of intelligent tutoring systems. volume 3220, pages 564–572, 08 2004.

[114] Kimberly N. Martin and Ivon Arroyo. Agentx: Using reinforcement learning to improve the effectiveness of intelligent tutoring systems. In James C. Lester, Rosa Maria Vicari, and Fábio Paraguaçu, editors, *Intelligent Tutoring Systems, 7th International Conference, ITS 2004, Maceiò, Alagoas, Brazil, August 30 - September 3, 2004, Proceedings*, volume 3220 of *Lecture Notes in Computer Science*, pages 564–572. Springer, 2004.

[115] Manolis Mavrikis. Data-driven prediction of the necessity of help requests in iles. In Wolfgang Nejdl, Judy Kay, Pearl Pu, and Eelco Herder, editors, *Adaptive Hypermedia and Adaptive Web-Based Systems, 5th International Conference, AH 2008, Hannover, Germany, July 29 - August 1, 2008.*

Proceedings, volume 5149 of *Lecture Notes in Computer Science*, pages 316–319. Springer, 2008.

[116] Jessica McBroom, Irena Koprinska, and Kalina Yacef. A survey of automated programming hint generation – the hints framework, 2019.

[117] Jessica McBroom, Kalina Yacef, and Irena Koprinska. Scalability in online computer programming education: Automated techniques for feedback, evaluation and equity. In Anna N. Rafferty, Jacob Whitehill, Cristóbal Romero, and Violetta Cavalli-Sforza, editors, *Proceedings of the 13th International Conference on Educational Data Mining, EDM 2020, Fully virtual conference, July 10-13, 2020*. International Educational Data Mining Society, 2020.

[118] Scott W. McQuiggan, Bradford W. Mott, and James C. Lester. Modeling self-efficacy in intelligent tutoring systems: An inductive approach. *User Model. User Adapt. Interact.*, 18(1-2):81–123, 2008.

[119] Sein Minn, Michel C. Desmarais, and Shunkai Fu. Refinement of a q-matrix with an ensemble technique based on multi-label classification algorithms. In Katrien Verbert, Mike Sharples, and Tomaz Klobucar, editors, *Adaptive and Adaptable Learning - 11th European Conference on Technology Enhanced Learning, EC-TEL 2016, Lyon, France, September 13-16, 2016, Proceedings*, volume 9891 of *Lecture Notes in Computer Science*, pages 165–178. Springer, 2016.

[120] Christopher M. Mitchell, Kristy Elizabeth Boyer, and James C. Lester. A markov decision process model of tutorial intervention in task-oriented dialogue. In H. Chad Lane, Kalina Yacef, Jack Mostow, and Philip Pavlik, editors, *Artificial Intelligence in Education*, pages 828–831, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[121] Steven Moore, Huy A. Nguyen, and John Stamper. Towards crowdsourcing the identification of knowledge components. In *Proceedings of the Seventh ACM Conference on Learning @ Scale*, L@S '20, page 245–248, New York, NY, USA, 2020. Association for Computing Machinery.

[122] M. Mozer and Robert V. Lindsey. Predicting and improving memory retention: Psychological theory matters in the big data era. 2016.

[123] Farah Nadeem, Huy Nguyen, Yang Liu, and Mari Ostendorf. Automated essay scoring with discourse-aware neural models. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 484–493, Florence, Italy, August 2019. Association for Computational Linguistics.

[124] Koki Nagatani, Qian Zhang, Masahiro Sato, Yan-Ying Chen, Francine Chen, and Tomoko Ohkuma. Augmenting knowledge tracing by considering forgetting behavior. In *The World Wide Web Conference*, WWW '19,

page 3101–3107, New York, NY, USA, 2019. Association for Computing Machinery.

[125] Hiromi Nakagawa, Yusuke Iwasawa, and Yutaka Matsuo. Graph-based knowledge tracing: Modeling student proficiency using graph neural network. In *IEEE/WIC/ACM International Conference on Web Intelligence*, WI '19, page 156–163, New York, NY, USA, 2019. Association for Computing Machinery.

[126] Huyen Nguyen and Lucio Deryn. eural networks for automated essay grading, 2016.

[127] Bahador B. Nooraei, Zachary A. Pardos, Neil T. Heffernan, and Ryan Shaun Joazeiro de Baker. Less is more: Improving the speed and prediction power of knowledge tracing by using less data. In Mykola Pechenizkiy, Toon Calders, Cristina Conati, Sebastián Ventura, Cristóbal Romero, and John C. Stamper, editors, *EDM*, pages 101–110. www.educationaldatamining.org, 2011.

[128] Benjamin Paaßen, Barbara Hammer, Thomas William Price, Tiffany Barnes, Sebastian Gross, and Niels Pinkwart. The continuous hint factory - providing hints in vast and sparsely populated edit distance spaces. *EDM*, abs/1708.06564, 2017.

[129] Shalini Pandey and George Karypis. A self-attentive model for knowledge tracing. In Collin F. Lynch, Agathe Merceron, Michel Desmarais, and Roger Nkambou, editors, *EDM 2019 - Proceedings of the 12th International Conference on Educational Data Mining*, EDM 2019 - Proceedings of the 12th International Conference on Educational Data Mining, pages 384–389. International Educational Data Mining Society, January 2019. 12th International Conference on Educational Data Mining, EDM 2019 ; Conference date: 02-07-2019 Through 05-07-2019.

[130] Shalini Pandey and Jaideep Srivastava. Rkt: Relation-aware self-attention for knowledge tracing. In *Proceedings of the 29th ACM International Conference on Information amp; Knowledge Management*, CIKM '20, page 1205–1214, New York, NY, USA, 2020. Association for Computing Machinery.

[131] Zachary A. Pardos and Neil T. Heffernan. Modeling individualization in a bayesian networks implementation of knowledge tracing. In Paul De Bra, Alfred Kobsa, and David Chin, editors, *User Modeling, Adaptation, and Personalization*, pages 255–266, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[132] Philip I. Pavlik, Hao Cen, and Kenneth R. Koedinger. Performance factors analysis –a new alternative to knowledge tracing. In *Proceedings of the 2009 Conference on Artificial Intelligence in Education: Building*

Learning Systems That Care: From Knowledge Representation to Affective Modelling, page 531–538, NLD, 2009. IOS Press.

[133] Radek Pelánek. A brief overview of metrics for evaluation of student models. In *EDM*, 2014.

[134] Radek Pelánek. Bayesian knowledge tracing, logistic models, and beyond: An overview of learner modeling techniques. *User Modeling and User-Adapted Interaction*, 27(3–5):313–350, December 2017.

[135] Radek Pelánek and Jiří Řihák. Experimental analysis of mastery learning criteria. In *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*, UMAP '17, page 156–163, New York, NY, USA, 2017. Association for Computing Machinery.

[136] Y. Pian, Y. Lu, P. Chen, and Q. Duan. Coglearn: A cognitive graph-oriented online learning system. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pages 2020–2023, 2019.

[137] Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas Guibas, and Jascha Sohl-Dickstein. Deep knowledge tracing. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 505–513, Cambridge, MA, USA, 2015. MIT Press.

[138] Chris Piech, Jonathan Huang, Andy Nguyen, Mike Phulsuksombati, Mehran Sahami, and Leonidas Guibas. Learning program embeddings to propagate feedback on student code. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 1093–1102. JMLR.org, 2015.

[139] Chris Piech, Mehran Sahami, Jonathan Huang, and Leonidas Guibas. Autonomously generating hints by inferring problem solving policies. In *Proceedings of the Second (2015) ACM Conference on Learning @ Scale*, L@S '15, page 195–204, New York, NY, USA, 2015. Association for Computing Machinery.

[140] Paul Pimsleur. A memory schedule. *The Modern Language Journal*, 51(73-75):71–132, 1967.

[141] Paul Prinsloo and Sharon Slade. Ethics and Learning Analytics: Charting the (Un)Charted. In Charles Lang, George Siemens, Alyssa Friend Wise, and Dragan Gaševic, editors, *The Handbook of Learning Analytics*, pages 49–57. Society for Learning Analytics Research (SoLAR), Alberta, Canada, 1 edition, 2017.

[142] Shi Pu, Michael Yudelson, Lu Ou, and Yuchi Huang. Deep knowledge tracing with transformers. In Ig Ibert Bittencourt, Mutlu Cukurova, Kasia Muldner, Rose Luckin, and Eva Millán, editors, *Artificial Intelligence in Education*, pages 252–256, Cham, 2020. Springer International Publishing.

[143] Yumeng Qiu, Yingmei Qi, Hanyuan Lu, Zachary A. Pardos, and Neil T. Heffernan. Does time matter? modeling the effect of time with bayesian knowledge tracing. In Mykola Pechenizkiy, Toon Calders, Cristina Conati, Sebastián Ventura, Cristóbal Romero, and John C. Stamper, editors, *Proceedings of the 4th International Conference on Educational Data Mining, Eindhoven, The Netherlands, July 6-8, 2011*, pages 139–148. www.educationaldatamining.org, 2011.

[144] Anna Rafferty, Huiji Ying, and Joseph Williams. Statistical consequences of using multi-armed bandits to conduct adaptive educational experiments. *Journal of Educational Data Mining*, 11(1):47–79, Jun 2019. The file is in PDF format. If your computer does not recognize it, simply download the file and then open it with your browser.

[145] Anna N. Rafferty, Rachel A. Jansen, and Thomas L. Griffiths. Using inverse planning for personalized feedback. pages 472–477, January 2016. 9th International Conference on Educational Data Mining, EDM 2016 ; Conference date: 29-06-2016 Through 02-07-2016.

[146] Juan Rastrollo-Guerrero, Juan A. Gomez-Pulido, and Arturo Domínguez. Analyzing and predicting students' performance by means of machine learning: A review. *Applied Sciences*, 10:1042, 02 2020.

[147] Siddharth Reddy, Igor Labutov, Siddhartha Banerjee, and Thorsten Joachims. Unbounded human learning: Optimal scheduling for spaced repetition. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 1815–1824, New York, NY, USA, 2016. Association for Computing Machinery.

[148] Siddharth Reddy, S. Levine, and Anca D. Dragan. Accelerating human learning with deep reinforcement learning. 2017.

[149] Kelly Rivers and Kenneth R. Koedinger. Automating hint generation with solution space path construction. In *12th International Conference on Intelligent Tutoring Systems - Volume 8474*, ITS 2014, page 329–339, Berlin, Heidelberg, 2014. Springer-Verlag.

[150] J. Robison, S. McQuiggan, and J. Lester. Evaluating the consequences of affective feedback in intelligent tutoring systems. In *2009 3rd International Conference on Affective Computing and Intelligent Interaction and Workshops*, pages 1–6, 2009.

[151] Pedro Uria Rodriguez, Amir Jafari, and Christopher M. Ormerod. Language models and automated essay scoring. *CoRR*, abs/1909.09482, 2019.

[152] Cristóbal Romero and Sebastián Ventura. Educational data science in massive open online courses. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, 7(1), 2017.

[153] Carolyn Rose and Alla Tovares. *What Sociolinguistics and Machine Learning Have to Say to Each Other About Interaction Analysis*, pages 289–299. 04 2015.

[154] Nataniel Ruiz, Mona Jalal, Vitaly Ablavsky, Danielle Allessio, John J. Magee, Jacob Whitehill, Ivon Arroyo, Beverly Park Woolf, Stan Sclaroff, and Margrit Betke. Leveraging affect transfer learning for behavior prediction in an intelligent tutoring system. *CoRR*, abs/2002.05242, 2020.

[155] Markel Sanz Ausin, Mehak Maniktala, Tiffany Barnes, and Min Chi. Exploring the impact of simple explanations and agency on batch deep reinforcement learning induced pedagogical policies. In Ig Ibert Bittencourt, Mutlu Cukurova, Kasia Muldner, Rose Luckin, and Eva Millán, editors, *Artificial Intelligence in Education*, pages 472–485, Cham, 2020. Springer International Publishing.

[156] A. Sapountzi, Sandjai Bhulai, I. Cornelisz, and C. Van Klaveren. Dynamic knowledge tracing models for large-scale adaptive learning environments. *International Journal on Advances in Intelligent Systems*, 12(12):93–110, June 2019.

[157] B. H. Sreenivasa Sarma and Balaraman Ravindran. Intelligent tutoring systems using reinforcement learning to teach autistic students. In Alladi Venkatesh, Timothy A. Gonsalves, Andrew Monk, and Kathy Buckner, editors, *Home Informatics and Telematics: ICT for The Next Billion - Proceedings of IFIP TC 9, WG 9.3 HOIT 2007 Conference, August 22-25, 2007, Chennai, India*, volume 241 of *IFIP*, pages 65–78. Springer, 2007.

[158] Carlotta Schatten, Ruth Janning, and Lars Schmidt-Thieme. Vygotsky based sequencing without domain information: A matrix factorization approach. *CSEDU (Selected Papers)*, pages 35–51, 2014.

[159] Carlotta Schatten and Lars Schmidt-Thieme. Student progress modeling with skills deficiency aware kalman filters. In *Proceedings of the 8th International Conference on Computer Supported Education*, CSEDU 2016, page 31–42, Setubal, PRT, 2016. SCITEPRESS - Science and Technology Publications, Lda.

[160] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017.

[161] Bailey P. Sclater, N. Jisc: Code of practice for learning analytics, 2015.

[162] Avi Segal, Yossi Ben David, Joseph Jay Williams, Kobi Gal, and Yaar Shalom. Combining difficulty ranking with multi-armed bandits to sequence educational content. In Carolyn Penstein Rosé, Roberto Martínez Maldonado, Heinz Ulrich Hoppe, Rose Luckin, Manolis Mavrikis, Kaska

Porayska-Pomsta, Bruce M. McLaren, and Benedict du Boulay, editors, *Artificial Intelligence in Education - 19th International Conference, AIED 2018, London, UK, June 27-30, 2018, Proceedings, Part II*, volume 10948 of *Lecture Notes in Computer Science*, pages 317–321. Springer, 2018.

[163] Ayon Sen, Purav Patel, Martina A. Rau, Blake Mason, Robert Nowak, Timothy T. Rogers, and Jerry Zhu. For teaching perceptual fluency, machines beat human experts. In Chuck Kalish, Martina A. Rau, Xiaojin (Jerry) Zhu, and Timothy T. Rogers, editors, *Proceedings of the 40th Annual Meeting of the Cognitive Science Society, CogSci 2018, Madison, WI, USA, July 25-28, 2018.* cognitivesciencesociety.org, 2018.

[164] Tirth Shah, Lukas Olson, Aditya Sharma, and Nirmal Patel. Explainable knowledge tracing models for big data: Is ensembling an answer?, 2020.

[165] Liping Shen, Minjuan Wang, and Ruimin Shen. Affective e-learning: Using "emotional" data to improve learning in pervasive learning environment. *J. Educ. Technol. Soc.*, 12(2):176–189, 2009.

[166] Shitian Shen, Markel Sanz Ausin, Behrooz Mostafavi, and Min Chi. Improving learning amp; reducing time: A constrained action-based reinforcement learning approach. In *Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization*, UMAP '18, page 43–51, New York, NY, USA, 2018. Association for Computing Machinery.

[167] Shitian Shen and Min Chi. Reinforcement learning: The sooner the better, or the later the better? In *Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization*, UMAP '16, page 37–44, New York, NY, USA, 2016. Association for Computing Machinery.

[168] Dongmin Shin, Yugeun Shim, Hangyeol Yu, Seewoo Lee, Byungsoo Kim, and Youngduck Choi. Saint+: Integrating temporal features for ednet correctness prediction, 2020.

[169] Valerie J. Shute. Focus on formative feedback. *Review of Educational Research*, 78(1):153–189, 2008.

[170] Stefan Slater, Ryan Baker, Ma. Victoria Almeda, Alex Bowers, and Neil Heffernan. Using correlational topic modeling for automated topic identification in intelligent tutoring systems. In *Proceedings of the Seventh International Learning Analytics amp; Knowledge Conference*, LAK '17, page 393–397, New York, NY, USA, 2017. Association for Computing Machinery.

[171] Shashank Sonkar, Andrew S. Lan, Andrew E. Waters, Phillip Grimaldi, and Richard G. Baraniuk. qdkt: Question-centric deep knowledge tracing. In Anna N. Rafferty, Jacob Whitehill, Cristóbal Romero, and Violetta Cavalli-Sforza, editors, *Proceedings of the 13th International Conference on Educational Data Mining, EDM 2020, Fully virtual conference, July 10-13, 2020.* International Educational Data Mining Society, 2020.

[172] John Stamper, Tiffany Barnes, Lorrie Lehmann, and Marvin Croy. The hint factory: Automatic generation of contextualized help for existing computer aided instruction. 01 2008.

[173] John C. Stamper, Michael Eagle, Tiffany Barnes, and Marvin J. Croy. Experimental evaluation of automatic hint generation for a logic tutor. *Int. J. Artif. Intell. Educ.*, 22(1-2):3–17, 2013.

[174] John C. Stamper and Kenneth R. Koedinger. Human-machine student model discovery and improvement using datashop. In *Proceedings of the 15th International Conference on Artificial Intelligence in Education*, AIED'11, page 353–360, Berlin, Heidelberg, 2011. Springer-Verlag.

[175] Yu Su, Qingwen Liu, Qi Liu, Zhenya Huang, Yu Yin, Enhong Chen, Chris H. Q. Ding, Si Wei, and Guoping Hu. Exercise-enhanced sequential modeling for student performance prediction. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 2435–2443. AAAI Press, 2018.

[176] Yuan Sun, Shiwei Ye, Shunya Inoue, and Yi Sun. Alternating recursive method for q-matrix learning. In John C. Stamper, Zachary A. Pardos, Manolis Mavrikis, and Bruce M. McLaren, editors, *Proceedings of the 7th International Conference on Educational Data Mining, EDM 2014, London, UK, July 4-7, 2014*, pages 14–20. International Educational Data Mining Society (IEDMS), 2014.

[177] Mack Sweeney, Jaime Lester, Huzefa Rangwala, and Aditya Johri. Next-Term Student Performance Prediction: A Recommender Systems Approach. *Journal of Educational Data Mining*, 8(1):22–51, September 2016. The file is in PDF format. If your computer does not recognize it, simply download the file and then open it with your browser.

[178] Kaveh Taghipour and Hwee Tou Ng. A neural approach to automated essay scoring. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1882–1891, Austin, Texas, November 2016. Association for Computational Linguistics.

[179] Nguyen Thai-Nghe, Lucas Drumond, T. Horváth, and L. Schmidt-Thieme. Using factorization machines for student modeling. In *UMAP Workshops*, 2012.

[180] Nguyen Thai-nghe, Lucas Drumond, Artus Krohn-grimberghe, Ros Nanopoulos, and Lars Schmidt-thieme. Factorization techniques for predicting student performance. In *In Educational Recommender Systems and Technologies: Practices and Challenges (In*, 2011.

[181] Hanshuang Tong, Yun Zhou, and Zhen Wang. HGKT : Introducing problem schema with hierarchical exercise graph for knowledge tracing. *CoRR*, abs/2006.16915, 2020.

[182] Maria Tsiakmaki, Georgios Kostopoulos, Sotiris Kotsiantis, and Omiros Ragos. Transfer learning from deep neural networks for predicting student performance. *Applied Sciences*, 10(6), 2020.

[183] Masaki Uto and Masashi Okano. Robust neural automated essay scoring using item response theory. In Ig Ibert Bittencourt, Mutlu Cukurova, Kasia Muldner, Rose Luckin, and Eva Millán, editors, *Artificial Intelligence in Education*, pages 549–561, Cham, 2020. Springer International Publishing.

[184] Masaki Uto, Yikuan Xie, and Maomi Ueno. Neural automated essay scoring incorporating handcrafted features. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6077–6088, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics.

[185] Willem J. van der Linden and R.K. Hambleton. *Handbook of modern item response theory.* Springer, 1997.

[186] Jill-Jênn Vie. Deep factorization machines for knowledge tracing. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 370–373, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[187] Jill-Jênn Vie and Hisashi Kashima. Knowledge Tracing Machines: Factorization Machines for Knowledge Tracing. In *Proceedings of the 33th AAAI Conference on Artificial Intelligence*, pages 750–757, 2019.

[188] Massimo Vitiello, Simon Walk, Vanessa Chang, Rocael Hernández, Denis Helic, and Christian Guetl. MOOC dropouts: A multi-system classifier. In Élise Lavoué, Hendrik Drachsler, Katrien Verbert, Julien Broisin, and Mar Pérez-Sanagustín, editors, *Data Driven Approaches in Digital Education - 12th European Conference on Technology Enhanced Learning, EC-TEL 2017, Tallinn, Estonia, September 12-15, 2017, Proceedings*, volume 10474 of *Lecture Notes in Computer Science*, pages 300–314. Springer, 2017.

[189] Fangju Wang. Learning teaching in teaching: Online reinforcement learning for intelligent tutoring. In James J. (Jong Hyuk) Park, Ivan Stojmenovic, Min Choi, and Fatos Xhafa, editors, *Future Information Technology*, pages 191–196, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[190] Shuting Wang, Alexander Ororbia, Zhaohui Wu, Kyle Williams, Chen Liang, Bart Pursel, and C. Lee Giles. Using prerequisites to extract concept maps from textbooks. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16,

page 317–326, New York, NY, USA, 2016. Association for Computing Machinery.

[191] A. E. Waters, A. S. Lan, and C. Studer. Sparse probit factor analysis for learning analytics. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8776–8780, 2013.

[192] J. Whitehill and J. Movellan. Approximately optimal teaching of approximately optimal learners. *IEEE Transactions on Learning Technologies*, 11(2):152–164, 2018.

[193] Kevin H. Wilson, Yan Karklin, Bojian Han, and Chaitanya Ekanadham. Back to the basics: Bayesian extensions of irt outperform neural networks for proficiency estimation, 2016.

[194] Chris Wong. Sequence based course recommender for personalized curriculum planning. In Carolyn Penstein Rosé, Roberto Martínez-Maldonado, H. Ulrich Hoppe, Rose Luckin, Manolis Mavrikis, Kaska Porayska-Pomsta, Bruce McLaren, and Benedict du Boulay, editors, *Artificial Intelligence in Education*, pages 531–534, Cham, 2018. Springer International Publishing.

[195] Beverly Woolf, Winslow Burleson, Ivon Arroyo, Toby Dragon, David Cooper, and Rosalind Picard. Affect-aware tutors: recognising and responding to student affect. *International Journal of Learning Technology*, 4(3-4):129–164, 2009.

[196] Beverly Park Woolf, Ivon Arroyo, David Cooper, Winslow Burleson, and Kasia Muldner. *Affective Tutors: Automatic Detection of and Response to Student Emotion*, pages 207–227. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[197] P. Woźniak and E. J. Gorzelańczyk. Optimization of repetition spacing in the practice of learning. *Acta neurobiologiae experimentalis*, 54 1:59–62, 1994.

[198] Xiaolu Xiong, Siyuan Zhao, Eric Van Inwegen, and Joseph Beck. Going deeper with deep knowledge tracing. In Tiffany Barnes, Min Chi, and Mingyu Feng, editors, *Proceedings of the 9th International Conference on Educational Data Mining, EDM 2016, Raleigh, North Carolina, USA, June 29 - July 2, 2016*, pages 545–550. International Educational Data Mining Society (IEDMS), 2016.

[199] Yanbo Xu and Jack Mostow. Comparison of methods to trace multiple subskills: Is lr-dbn best? In Kalina Yacef, Osmar R. Zaïane, Arnon Hershkovitz, Michael Yudelson, and John C. Stamper, editors, *EDM*, pages 41–48. www.educationaldatamining.org, 2012.

[200] Shanghui Yang, Mengxia Zhu, Jingyang Hou, and Xuesong Lu. Deep knowledge tracing with convolutions, 2020.

[201] Yang Yang, Jian Shen, Yanru Qu, Yunfei Liu, Kerong Wang, Yaoming Zhu, Weinan Zhang, and Yong Yu. GIKT: A graph-based interaction model for knowledge tracing. *CoRR*, abs/2009.05991, 2020.

[202] Lihua Yao and R. Schwarz. A multidimensional partial credit model with associated item and test statistics: An application to mixed-format tests. *Applied Psychological Measurement*, 30:469 – 492, 2005.

[203] Teresa Yeo, Parameswaran Kamalaruban, Adish Singla, Arpit Merchant, Thibault Asselborn, Louis Faucon, Pierre Dillenbourg, and Volkan Cevher. Iterative classroom teaching. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):5684–5692, Jul. 2019.

[204] Chun-Kit Yeung. Deep-irt: Make deep learning based knowledge tracing explainable using item response theory. In Michel C. Desmarais, Collin F. Lynch, Agathe Merceron, and Roger Nkambou, editors, *Proceedings of the 12th International Conference on Educational Data Mining, EDM 2019, Montréal, Canada, July 2-5, 2019*. International Educational Data Mining Society (IEDMS), 2019.

[205] Chun-Kit Yeung and Dit-Yan Yeung. Addressing two problems in deep knowledge tracing via prediction-consistent regularization. In *Proceedings of the Fifth Annual ACM Conference on Learning at Scale*, L@S '18, New York, NY, USA, 2018. Association for Computing Machinery.

[206] Michael V. Yudelson, Kenneth R. Koedinger, and Geoffrey J. Gordon. Individualized bayesian knowledge tracing models. In H. Chad Lane, Kalina Yacef, Jack Mostow, and Philip Pavlik, editors, *Artificial Intelligence in Education*, pages 171–180, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[207] Ziheng Zeng, Snigdha Chaturvedi, Suma Bhat, and Dan Roth. Diad: Domain adaptation for learning at scale. In *Proceedings of the 9th International Conference on Learning Analytics amp; Knowledge*, LAK19, page 185–194, New York, NY, USA, 2019. Association for Computing Machinery.

[208] J. Zhang, Y. Mo, C. Chen, and X. He. Neural attentive knowledge tracing model for student performance prediction. In *2020 IEEE International Conference on Knowledge Graph (ICKG)*, pages 641–648, 2020.

[209] Jiani Zhang and Irwin King. Topological order discovery via deep knowledge tracing. In Akira Hirose, Seiichi Ozawa, Kenji Doya, Kazushi Ikeda, Minho Lee, and Derong Liu, editors, *Neural Information Processing*, pages 112–119, Cham, 2016. Springer International Publishing.

[210] Jiani Zhang, Xingjian Shi, Irwin King, and Dit-Yan Yeung. Dynamic key-value memory networks for knowledge tracing. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, page 765–774,

Republic and Canton of Geneva, CHE, 2017. International World Wide Web Conferences Steering Committee.

[211] Jinjin Zhao, Shreyansh Bhatt, Candace Thille, Neelesh Gattani, and Dawn Zimmaro. Cold start knowledge tracing with attentive neural turing machine. In *Proceedings of the Seventh ACM Conference on Learning @ Scale*, L@S '20, page 333–336, New York, NY, USA, 2020. Association for Computing Machinery.

[212] Jinjin Zhao, Shreyansh Bhatt, Candace Thille, Dawn Zimmaro, and Neelesh Gattani. Interpretable personalized knowledge tracing and next learning activity recommendation. In *Proceedings of the Seventh ACM Conference on Learning @ Scale*, L@S '20, page 325–328, New York, NY, USA, 2020. Association for Computing Machinery.

[213] Siqian Zhao, Chunpai Wang, and Shaghayegh Sahebi. Modeling knowledge acquisition from multiple learning resource types. In Anna N. Rafferty, Jacob Whitehill, Cristóbal Romero, and Violetta Cavalli-Sforza, editors, *Proceedings of the 13th International Conference on Educational Data Mining, EDM 2020, Fully virtual conference, July 10-13, 2020*. International Educational Data Mining Society, 2020.

[214] Siyuan Zhao, Yaqiong Zhang, Xiaolu Xiong, Anthony Botelho, and Neil Heffernan. A memory-augmented neural model for automated grading. In *Proceedings of the Fourth (2017) ACM Conference on Learning @ Scale*, L@S '17, page 189–192, New York, NY, USA, 2017. Association for Computing Machinery.

[215] Guojing Zhou, J. Wang, Collin Lynch, and Min Chi. Towards closing the loop: Bridging machine-induced pedagogical policies to learning theories. In *EDM*, 2017.

[216] Xiaojin Zhu, Adish Singla, Sandra Zilles, and Anna N. Rafferty. An Overview of Machine Teaching. *arXiv e-prints*, page arXiv:1801.05927, January 2018.

[217] Matthieu Zimmer, Paolo Viappiani, and Paul Weng. Teacher-Student Framework: a Reinforcement Learning Approach. In *AAMAS Workshop Autonomous Robots and Multirobot Systems*, Paris, France, May 2014.

[218] Kurtis Zimmerman and Chandan R. Rupakheti. An automated framework for recommending program elements to novices. In *Proceedings of the 30th IEEE/ACM International Conference on Automated Software Engineering*, ASE '15, page 283–288. IEEE Press, 2015.

[219] Kurtis Zimmerman and Chandan Raj Rupakheti. An automated framework for recommending program elements to novices (N). In Myra B. Cohen, Lars Grunske, and Michael Whalen, editors, *30th IEEE/ACM International Conference on Automated Software Engineering, ASE 2015,*

Lincoln, NE, USA, November 9-13, 2015, pages 283–288. IEEE Computer Society, 2015.

[220] Neil L. Zimmerman and Ryan S. Baker. Mining knowledge components from many untagged questions. In *Proceedings of the Seventh International Learning Analytics amp; Knowledge Conference*, LAK '17, page 566–567, New York, NY, USA, 2017. Association for Computing Machinery.

[221] Laura §, Asimina Vasalou, Kay Berkling, Wolmet Barendregt, and Manolis Mavrikis. *A Critical Examination of Feedback in Early Reading Games*, page 1–12. Association for Computing Machinery, New York, NY, USA, 2018.