Learning Representations of Student Knowledge and Educational Content

Siddharth Reddy[1,2]   Igor Labutov[3]   Thorsten Joachims[1]

[1]Department of Computer Science
Cornell University
Ithaca, NY, USA

[2]Knewton, Inc.
New York, NY, USA

[3]Department of Electrical and Computer Engineering
Cornell University
Ithaca, NY, USA

Machine Learning for Education Workshop, ICML 2015

Motivation

How can we guide students through large, heterogeneous collections of educational content to help them achieve their goals?

- Content = lessons and assessments
- Goal = passing an assessment
- Guide = recommending personalized lesson sequences

Motivation

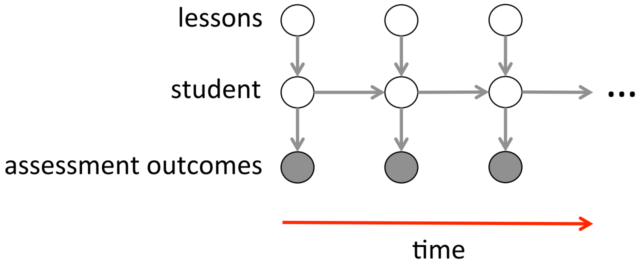
How can we guide students through large, heterogeneous collections of educational content to help them achieve their goals?

- Content = lessons and assessments
- Goal = passing an assessment
- Guide = recommending personalized lesson sequences

General Approach

1. Alice completed Lesson B
2. Alice passed Assessment C
3. Alice failed Assessment D
4. Alice completed Lesson E
5. Alice passed Assessment D
6. ...

1. Bob completed Lesson B
2. Bob failed Assessment C
3. Bob failed Assessment D
4. Bob completed Lesson E
5. Bob passed Assessment C
6. ...



Related Work

Existing frameworks:

- Bayesian Knowledge Tracing (BKT)
- Item Response Theory (IRT)
- Sparse Factor Analysis (SPARFA)

	knowledge tracing	lesson effects	evaluations
DKT	RNN	-	outcome prediction
T-SKIRT	input-output HMM	-	outcome prediction
SPARFA-Trace	input-output HMM	skill gains	outcome prediction
Embedding	input-output HMM	skill gains, prereqs	outcome prediction, lesson recommendation

Student and Content Representations

- **Student** = a set of latent skill levels that vary over time
- **Lesson module** = a vector of skill gains, and a set of prerequisite skill requirements
- **Assessment module** = a set of skill requirements
- A student can be tested on an assessment module, which has a **pass-fail result**. The likelihood of passing should be high when a student has skill levels that exceed the assessment requirements, and vice-versa.
- A student can complete lesson modules to learn over time, though the **skill gains** from a lesson module are modulated by **prerequisite knowledge**

Student and Content Representations

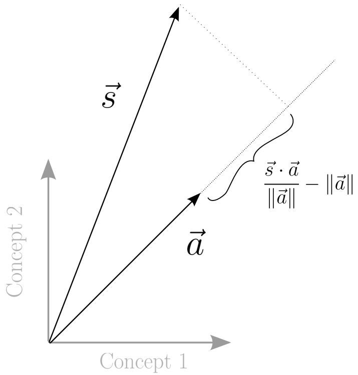
- **Student** = a set of latent skill levels that vary over time
- **Lesson module** = a vector of skill gains, and a set of prerequisite skill requirements
- **Assessment module** = a set of skill requirements
- A student can be tested on an assessment module, which has a **pass-fail result**. The likelihood of passing should be high when a student has skill levels that exceed the assessment requirements, and vice-versa.
- A student can complete lesson modules to learn over time, though the **skill gains** from a lesson module are modulated by **prerequisite knowledge**

Modeling Assessment Outcomes

For student $\vec{s}_t \in \mathbb{R}^d$, assessment $\vec{a} \in \mathbb{R}^d$, and result $R \in \{0, 1\}$,

$$R \sim \mathrm{Bernoulli}(\phi(\Delta(\vec{s}_t, \vec{a})))$$

where ϕ is the logistic function and $\Delta(\vec{s}_t, \vec{a}) = \frac{\vec{s}_t \cdot \vec{a}}{||\vec{a}||} - ||\vec{a}|| + \gamma_s + \gamma_a$



Modeling Student Learning from Lessons

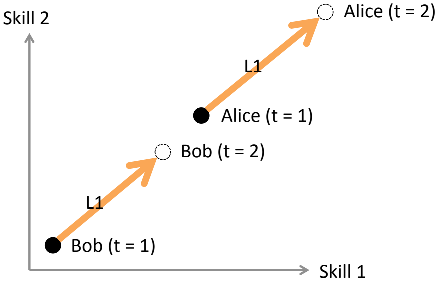
For student $\vec{s}_t \in \mathbb{R}^d$ who worked on a lesson with skill gains $\vec{\ell} \in \mathbb{R}^d_+$ and no prerequisites at time $t+1$, the updated student state is

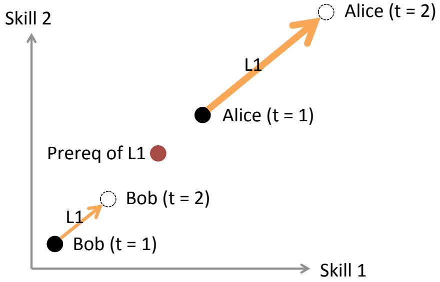
$$\vec{s}_{t+1} \sim \mathcal{N}\left(\vec{s}_t + \vec{\ell}, \Sigma\right)$$

where the covariance matrix $\Sigma = I_d \sigma^2$ is diagonal. For a lesson with prerequisites $\vec{q} \in \mathbb{R}^d$,

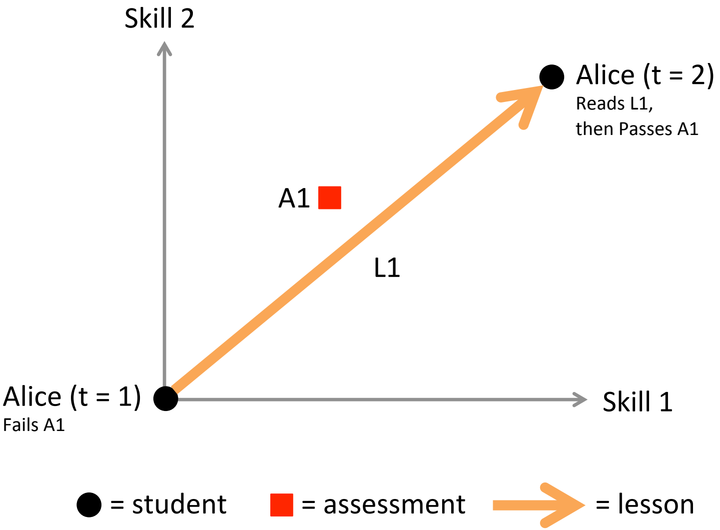
$$\vec{s}_{t+1} \sim \mathcal{N}\left(\vec{s}_t + \vec{\ell} \cdot \phi(\Delta(\vec{s}_t, \vec{q})), \Sigma\right)$$

where $\Delta(\vec{s}_t, \vec{q}) = \frac{\vec{s}_t \cdot \vec{q}}{||\vec{q}||} - ||\vec{q}||$

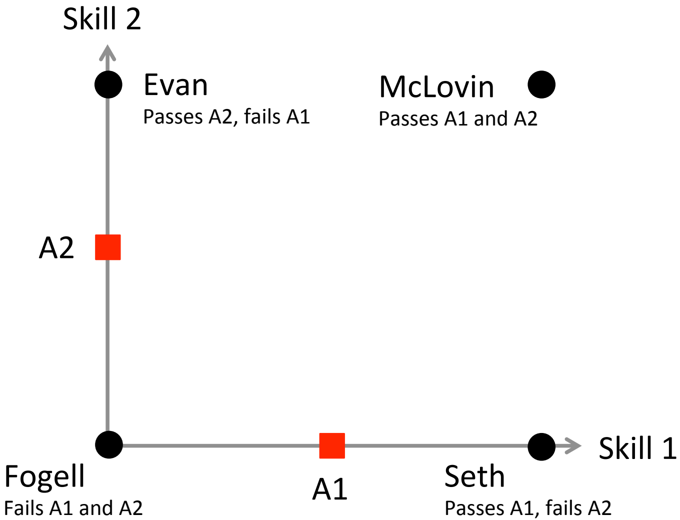




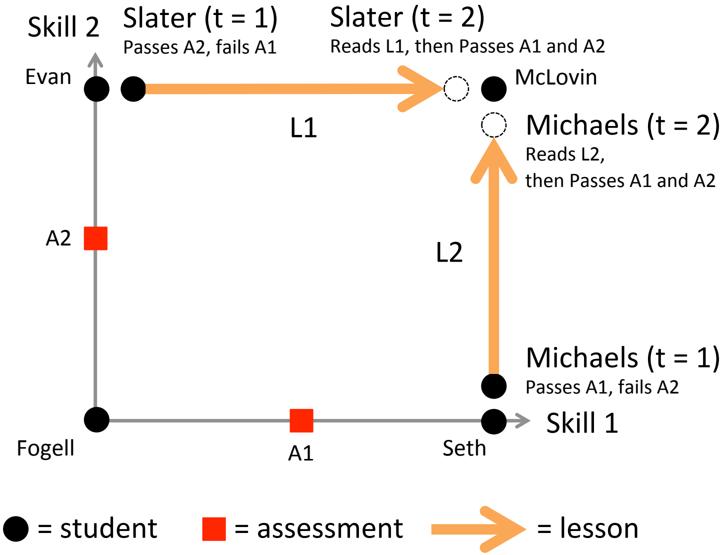
One-Dimensional Embedding



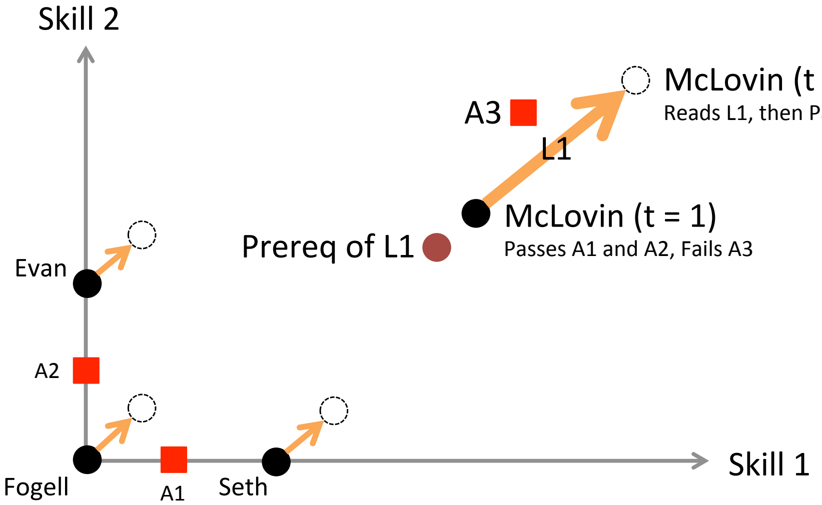
Two-Dimensional Embedding



Embedding with Lessons



Embedding with Lesson Prerequisites



Online Course Data

The two data sets from Knewton collectively contain

- 2,184,352 interaction logs
- 1,939 classrooms
- 6 months
- 7,034 students
- 7,217 lessons
- 7,287 assessments
- Average assessment pass rates of 0.712 and 0.693

Parameter Estimation

- We compute MAP estimates of model parameters Θ by maximizing

$$L(\Theta) = \sum_{\mathcal{A}} \log\left(Pr(R \mid \vec{s}_t, \vec{a}, \gamma_s, \gamma_a)\right) + \sum_{\mathcal{L}} \log\left(Pr(\vec{s}_{t+1} \mid \vec{s}_t, \vec{\ell}, \vec{q})\right) - \beta \cdot \lambda(\Theta)$$

  where $\mathcal{A}$ is the set of assessment interactions, $\mathcal{L}$ is the set of lesson interactions, $\lambda(\Theta)$ is a regularization term that penalizes the L_2 norms of embedding parameters (not bias terms γ), and β is a regularization parameter.
- Solved with L-BFGS-B and random parameter initializations

Assessment Result Prediction

- Hold out assessment interactions at the end of student histories
- Area under ROC Curve (AUC)
- Ten-fold cross-validation

Assessment Result Prediction

	d	s	a	ℓ	q	γ	Training	Validation
1	1	Y	N	N	N	N	0.723	0.686
2	1	N	Y	N	N	N	0.715	0.661
3	2	Y	Y	N	N	N	0.991	0.614
4	2	Y	Y	N	N	Y	0.992	0.726
5	2	Y	Y	Y	N	N	0.897	0.696
6	2	Y	Y	Y	N	Y	0.871	0.748
7	2	Y	Y	Y	Y	N	0.898	0.694
8	2	Y	Y	Y	Y	Y	0.882	0.749

- Joint embedding beats baselines
- Including **lesson embeddings** improves performance significantly
- Including **prerequisite embeddings** has a statistically insignificant effect on performance
- Including **bias terms** improves performance significantly

Assessment Result Prediction

	d	s	a	ℓ	q	γ	Training	Validation
1	1	Y	N	N	N	N	0.723	0.686
2	1	N	Y	N	N	N	0.715	0.661
3	2	Y	Y	N	N	N	0.991	0.614
4	2	Y	Y	N	N	Y	0.992	0.726
5	2	Y	Y	Y	N	N	0.897	0.696
6	2	Y	Y	Y	N	Y	0.871	0.748
7	2	Y	Y	Y	Y	N	0.898	0.694
8	2	Y	Y	Y	Y	Y	0.882	0.749

- Joint embedding beats baselines
- Including **lesson embeddings** improves performance significantly
- Including **prerequisite embeddings** has a statistically insignificant effect on performance
- Including **bias terms** improves performance significantly

Assessment Result Prediction

	d	s	a	ℓ	$\mathbf{q}$	γ	Training	Validation
1	1	Y	N	N	N	N	0.723	0.686
2	1	N	Y	N	N	N	0.715	0.661
3	2	Y	Y	N	N	N	0.991	0.614
4	2	Y	Y	N	N	Y	0.992	0.726
5	2	Y	Y	Y	N	N	0.897	0.696
6	2	Y	Y	Y	N	Y	0.871	0.748
7	2	Y	Y	Y	Y	N	0.898	0.694
8	2	Y	Y	Y	Y	Y	0.882	0.749

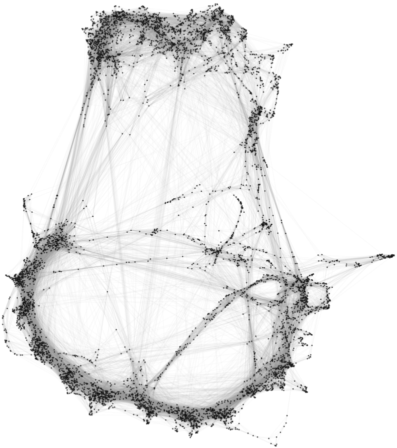
- Joint embedding beats baselines
- Including **lesson embeddings** improves performance significantly
- Including **prerequisite embeddings** has a statistically insignificant effect on performance
- Including **bias terms** improves performance significantly

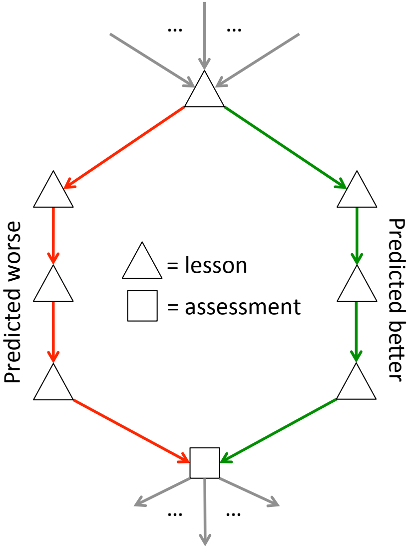
Assessment Result Prediction

	d	s	a	ℓ	q	γ	Training	Validation
1	1	Y	N	N	N	N	0.723	0.686
2	1	N	Y	N	N	N	0.715	0.661
3	2	Y	Y	N	N	N	0.991	0.614
4	2	Y	Y	N	N	Y	0.992	0.726
5	2	Y	Y	Y	N	N	0.897	0.696
6	2	Y	Y	Y	N	Y	0.871	0.748
7	2	Y	Y	Y	Y	N	0.898	0.694
8	2	Y	Y	Y	Y	Y	0.882	0.749

- Joint embedding beats baselines
- Including **lesson embeddings** improves performance significantly
- Including **prerequisite embeddings** has a statistically insignificant effect on performance
- Including **bias terms** improves performance significantly

Lesson Sequence Discrimination





Lesson Sequence Discrimination

- Expected relative gain from taking recommended path

$$\mathbb{E}\left[\frac{\mathbb{E}[R'] - \mathbb{E}[R]}{\mathbb{E}[R]}\right]$$

  where R' is the outcome at the end of the recommended path and R is the outcome at the end of the other path

- Propensity score matching
  - Student features = past outcomes
  - Logistic regression for propensity score estimation
  - Nearest neighbor matching

Lesson Sequence Discrimination

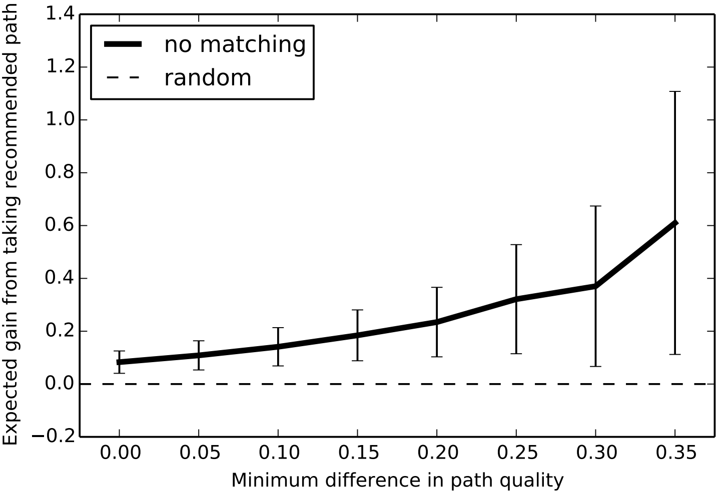
- Expected relative gain from taking recommended path

$$\mathbb{E}\left[\frac{\mathbb{E}[R'] - \mathbb{E}[R]}{\mathbb{E}[R]}\right]$$

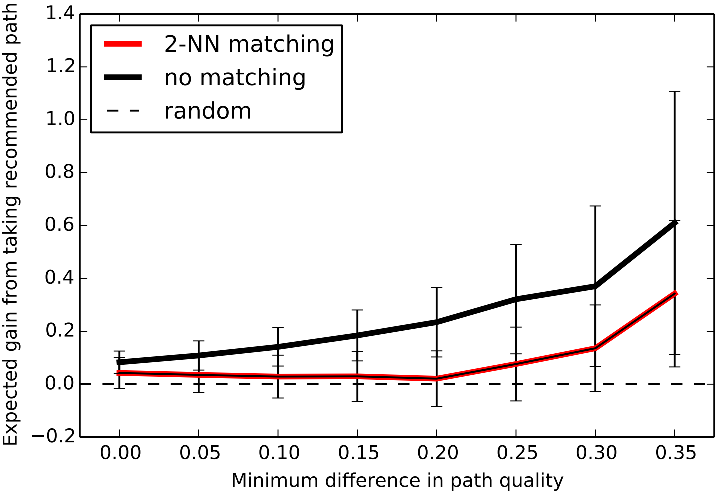
where R' is the outcome at the end of the recommended path and R is the outcome at the end of the other path

- Propensity score matching
  - Student features = past outcomes
  - Logistic regression for propensity score estimation
  - Nearest neighbor matching

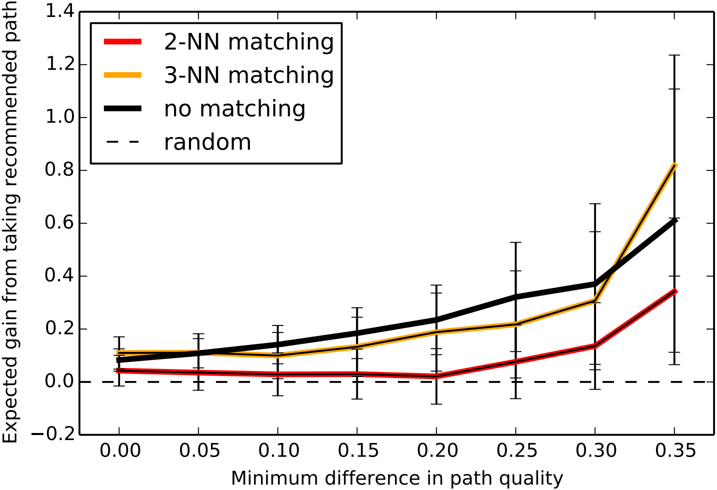
Lesson Sequence Discrimination



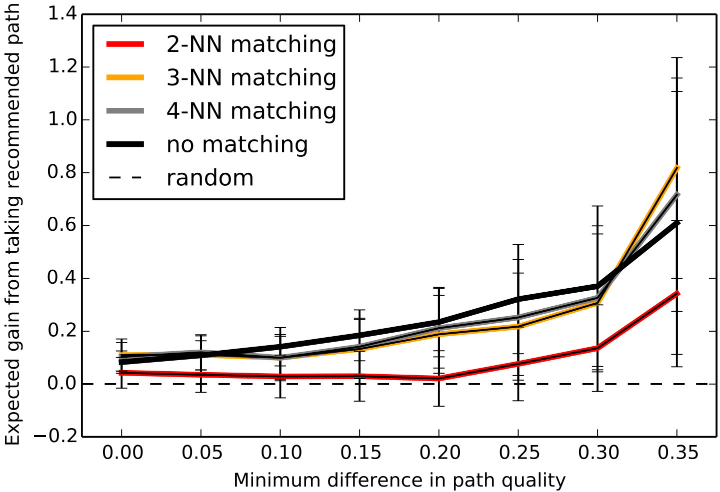
Lesson Sequence Discrimination



Lesson Sequence Discrimination



Lesson Sequence Discrimination



Summary

- Demonstrated the ability of an embedding model to successfully predict assessment results
  - Modeling skill gains from lessons is helpful
  - Modeling prerequisites for lessons is not helpful
- Introduced an offline methodology as a proxy for assessing the ability of a model to recommend personalized lesson sequences
  - Embedding model can distinguish between "good" and "bad" paths in *bubble* scenarios

Ongoing Work

- Cold Start Problem
  - Impose prior distribution on content embeddings based on *content-to-concept map* and *concept dependency graph* made by experts
- Personalized Scheduling
  - Jointly model *assessment results*, *response times*, and *number of attempts*
  - Model the *forgetting effect* and *offline learning* between interactions
  - Handle wall-clock time constraints on recommended sequences
- Content Analytics
  - Small-scale Mechanical Turk experiments for teaching basic programming
  - Measure lesson quality

Contact

- **Email**: sgr45@cornell.edu
- **Paper, Slides, Poster, Software**: http://siddharth.io