Introduction
○○○○○○○○

Logistic Regression
○○○○○○○○

Factorization Machines
○○○○○○○○○○○

Deep Learning
○○○○○○○○○○○

Conclusion
○○○○

# Optimizing Human Learning

Fabrice Popineau     Jill-Jênn Vie     Michal Valko

RIKEN AIP & New York University



June 4, 2019

## Outline

| | |
|---|---|
| 9:30 AM | Welcome and Introduction |
| 9:45 AM | Tutorial: Knowledge Tracing |
| 10:30 AM | Coffee Break |
| 11:00 AM | An Exploration of Disciplinary Literacy in Learners' Short Answers *Jean-Philippe Corbeil, Amal Zouaq, and Michel Gagnon* |
| 11:30 AM | Open Discussion |
| 12:15 AM | End of workshop |

# Optimizing Human Learning

We observe data collected by a platform (ITS, MOOC, etc.)

We can learn a generative model of the world ($\sim$ knowledge tracing)
Then learn a policy to optimize it (e.g. this workshop)

## Challenges

- Representations that evolve over time
  (actions from the teacher can modify the learner)
- Which objective function should be optimized?
- New users & items appear (cold-start)
- Sequential learning requires a measure of uncertainty
- High-stakes applications require interpretability

# Choosing the objective function to optimize

Maximize information $\rightarrow$ learners fail 50% of the time (good for the assessing institution, not for the learning student)

Maximize success rate $\rightarrow$ asking too easy questions

Maximize the growth of the success rate (Clement et al. 2015)

Compromise exploration (items that we don't know)
and exploitation (items that measure well)

Identify a gap from the learner (Teng et al. ICDM 2018)
+ assume that a item brings less learning when it was administered before (Seznec et al. AISTATS 2019, SequeL)

Increasing number of works(hops) about reinforcement learning in education

**Introduction**
00000000

Logistic Regression
00000000

Factorization Machines
00000000000

Deep Learning
00000000000

Conclusion
0000

## Predicting student performance

### Data

A population of students answering questions
- Events: "Student $i$ answered question $j$ correctly/incorrectly"

### Goal

- Learn the difficulty of questions automatically from data
- Measure the knowledge of students
- Potentially optimize their learning

### Assumption

Good model for prediction $\rightarrow$ Good adaptive policy for teaching

# Learning outcomes of this tutorial

- Logistic regression is amazing
  - Unidimensional
  - Takes IRT, PFA as special cases

- Factorization machines are even more amazing
  - Multidimensional
  - Take MIRT as special case

- It makes sense to consider deep neural networks
  - What does deep knowledge tracing model exactly?

## Families of models

- Factorization Machines (Rendle 2012)
    - Multidimensional Item Response Theory
    - Logistic Regression
        - Item Response Theory
        - Performance Factor Analysis
- Recurrent Neural Networks
    - Deep Knowledge Tracing (Piech et al. 2015)

Steffen Rendle (2012). "Factorization Machines with libFM". In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 3.3, 57:1–57:22. DOI: 10.1145/2168752.2168771

Chris Piech et al. (2015). "Deep knowledge tracing". In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 505–513

## Problems

### Weak generalization

Filling the blanks: some students did not attempt all questions

### Strong generalization

Cold-start: some new students are not in the train set

**Introduction**
○○○○○○○●

Logistic Regression
○○○○○○○○

Factorization Machines
○○○○○○○○○○○

Deep Learning
○○○○○○○○○○○

Conclusion
○○○○

## Dummy dataset

- User 1 answered Item 1 correct
- User 1 answered Item 2 incorrect
- User 2 answered Item 1 incorrect
- User 2 answered Item 1 correct
- User 2 answered Item 2 ???

| user | item | correct |
|------|------|---------|
| 1    | 1    | 1       |
| 1    | 2    | 0       |
| 2    | 1    | 0       |
| 2    | 1    | 1       |
| 2    | 2    | 0       |

dummy.csv

## Task 1: Item Response Theory

Learn abilities $\theta_i$ for each user $i$

Learn easiness $e_j$ for each item $j$ such that:

$$Pr(\text{User } i \text{ Item } j \text{ OK}) = \sigma(\theta_i + e_j)$$

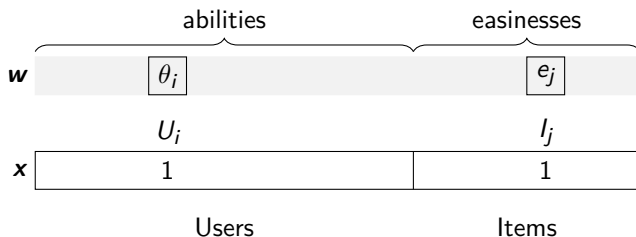$$\text{logit } Pr(\text{User } i \text{ Item } j \text{ OK}) = \theta_i + e_j$$

---

### Logistic regression

Learn $\boldsymbol{w}$ such that $\text{logit } Pr(\boldsymbol{x}) = \langle \boldsymbol{w}, \boldsymbol{x} \rangle$

Usually with L2 regularization: $\|\boldsymbol{w}\|_2^2$ penalty $\leftrightarrow$ Gaussian prior

## Graphically: IRT as logistic regression

Encoding of "User $i$ answered Item $j$":



$$\mathrm{logit}\, Pr(\text{User } i \text{ Item } j \text{ OK}) = \langle \boldsymbol{w}, \boldsymbol{x} \rangle = \theta_i + e_j$$

# Encoding

```
python encode.py --users --items
```

|  | Users |  |  | Items |  |
|---|---|---|---|---|---|
| $U_0$ | $U_1$ | $U_2$ | $I_0$ | $I_1$ | $I_2$ |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 |

```
data/dummy/X-ui.npz
```

Then logistic regression can be run on the sparse features:

```
python lr.py data/dummy/X-ui.npz
```

## Oh, there's a problem

```
python encode.py --users --items
```

```
python lr.py data/dummy/X-ui.npz
```

| | Users | | | Items | | | $\boldsymbol{y}_{\text{pred}}$ | $\boldsymbol{y}$ |
|---|---|---|---|---|---|---|---|---|
| | $U_0$ | $U_1$ | $U_2$ | $I_0$ | $I_1$ | $I_2$ | | |
| User 1 Item 1 OK | 0 | 1 | 0 | 0 | 1 | 0 | 0.575135 | 1 |
| User 1 Item 2 NOK | 0 | 1 | 0 | 0 | 0 | 1 | 0.395036 | 0 |
| User 2 Item 1 NOK | 0 | 0 | 1 | 0 | 1 | 0 | 0.545417 | 0 |
| User 2 Item 1 OK | 0 | 0 | 1 | 0 | 1 | 0 | 0.545417 | 1 |
| User 2 Item 2 NOK | 0 | 0 | 1 | 0 | 0 | 1 | 0.366595 | 0 |

We predict the same thing when there are several attempts.

## Count successes and failures

Keep track of what the student has done before:

| user | item | skill | correct | wins | fails |
|------|------|-------|---------|------|-------|
| 1 | 1 | 1 | 1 | 0 | 0 |
| 1 | 2 | 2 | 0 | 0 | 0 |
| 2 | 1 | 1 | 0 | 0 | 0 |
| 2 | 1 | 1 | 1 | 0 | 1 |
| 2 | 2 | 2 | 0 | 0 | 0 |

data/dummy/data.csv

## Task 2: Performance Factor Analysis

$W_{ik}$: how many successes of user $i$ over skill $k$ ($F_{ik}$: #failures)

Learn $\beta_k$, $\gamma_k$, $\delta_k$ for each skill $k$ such that:

$$\operatorname{logit} Pr(\text{User } i \text{ Item } j \text{ OK}) = \sum_{\text{Skill } k \text{ of Item } j} \beta_k + W_{ik}\gamma_k + F_{ik}\delta_k$$

```
python encode.py --skills --wins --fails
```

| | Skills | | | Wins | | | Fails | |
| $S_0$ | $S_1$ | $S_2$ | $S_0$ | $S_1$ | $S_2$ | $S_0$ | $S_1$ | $S_2$ |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

```
data/dummy/X-swf.npz
```

## Better!

```
python encode.py --skills --wins --fails
```

```
python lr.py data/dummy/X-swf.npz
```

|  | Skills | | | Wins | | | Fails | | | $y_{pred}$ | $y$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
|  | $S_0$ | $S_1$ | $S_2$ | $S_0$ | $S_1$ | $S_2$ | $S_0$ | $S_1$ | $S_2$ | | |
| User 1 Item 1 OK | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.544 | 1 |
| User 1 Item 2 NOK | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0.381 | 0 |
| User 2 Item 1 NOK | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.544 | 0 |
| User 2 Item 1 OK | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0.633 | 1 |
| User 2 Item 2 NOK | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0.381 | 0 |

## Task 3: a new model (but still logistic regression)

```
python encode.py --items --skills --wins --fails

python lr.py data/dummy/X-iswf.npz
```

## Here comes a new challenger

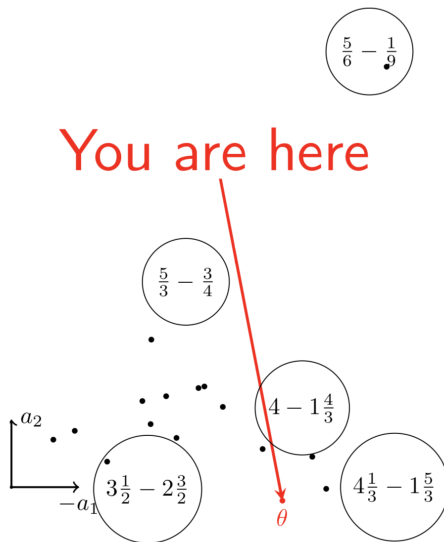How to model side information in, say, recommender systems?

### Logistic Regression

Learn a bias for each feature (each user, item, etc.)

### Factorization Machines
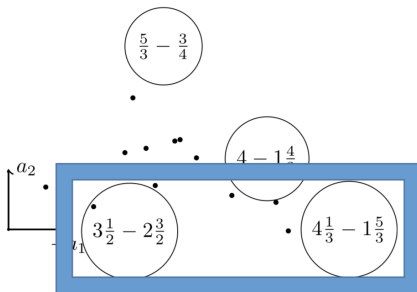
Learn a bias and an embedding for each feature

Introduction
○○○○○○○○

Logistic Regression
○○○○○○○○

**Factorization Machines**
○●○○○○○○○○○

Deep Learning
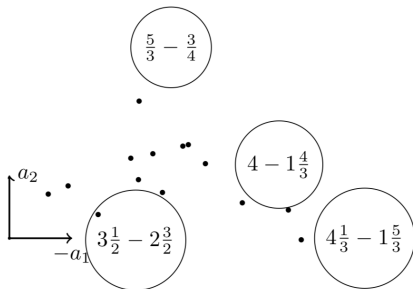○○○○○○○○○○○

Conclusion
○○○○

## What can be done with embeddings?

## Interpreting the components



$$\frac{5}{6} - \frac{1}{9}$$

**Items that
discriminate
only over one dimension**

$$\frac{5}{3} - \frac{3}{4}$$

| $3\frac{1}{2} - 2\frac{3}{2}$ | $4\frac{1}{3} - 2\frac{4}{3}$ | $4\frac{1}{3} - 1\frac{5}{3}$ |
|---|---|---|
| $b = 0.13$ | $b = -0.46$ | $b = -1.99$ |
| $-a_1 = 2.01$ | $-a_1 = 4.65$ | $-a_1 = 5.66$ |
| $a_2 = -0.03$ | $a_2 = -0.02$ | $a_2 = 0.00$ |

$$4 - 1\frac{4}{5}$$

$$a_2$$

$$3\frac{1}{2} - 2\frac{3}{2}$$

$$4\frac{1}{3} - 1\frac{5}{3}$$

## Interpreting the components



**Items that
highly discriminate
over both dimensions**

$$\left(\frac{3}{4} - \frac{3}{8}\right)$$     $$\left(\frac{5}{6} - \frac{1}{9}\right)$$

$b = 1.09$        $b = -0.28$
$-a_1 = 5.54$     $-a_1 = 5.29$
$a_2 = 6.22$      $a_2 = 6.44$

# Graphically: logistic regression
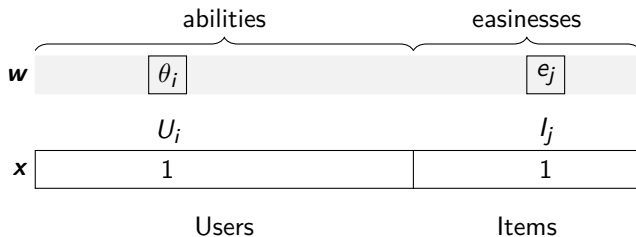
## How to model pairwise interactions with side information?

If you know user $i$ attempted item $j$ on mobile (not desktop)
How to model it?

$y$: score of event "user $i$ solves correctly item $j$"
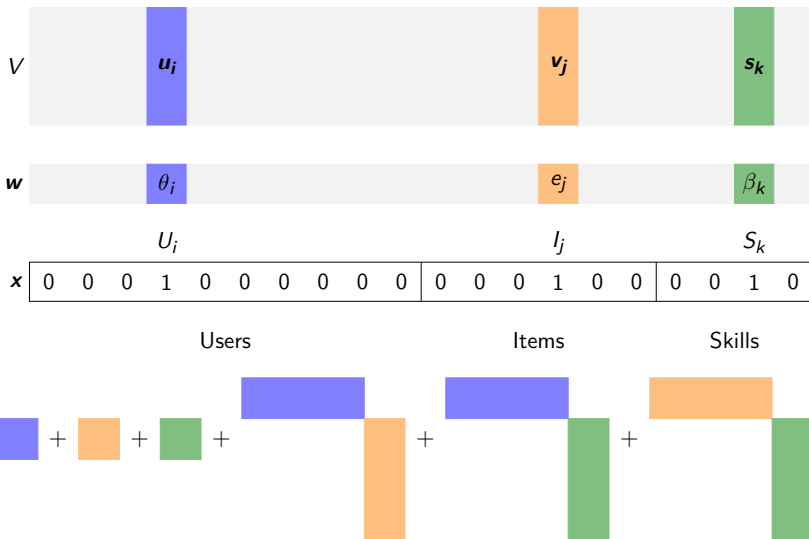
**IRT**

$$y = \theta_i + e_j$$

**Multidimensional IRT (similar to collaborative filtering)**

$$y = \theta_i + e_j + \langle \boldsymbol{v}_{\text{user } i}, \boldsymbol{v}_{\text{item } j} \rangle$$

**With side information**

$$y = \theta_i + e_j + w_{\text{mobile}} + \langle \boldsymbol{v}_{\text{user } i}, \boldsymbol{v}_{\text{item } j} \rangle + \langle \boldsymbol{v}_{\text{user } i}, \boldsymbol{v}_{\text{mobile}} \rangle + \langle \boldsymbol{v}_{\text{item } j}, \boldsymbol{v}_{\text{mobile}} \rangle$$

Introduction
○○○○○○○○

Logistic Regression
○○○○○○○○

**Factorization Machines**
○○○○○○●○○○○

Deep Learning
○○○○○○○○○○○

Conclusion
○○○○

# Graphically: factorization machines

# Formally: factorization machines

Learn bias $w_k$ and embedding $\boldsymbol{v_k}$ for each feature $k$ such that:

$$\operatorname{logit} p(\boldsymbol{x}) = \mu + \underbrace{\sum_{k=1}^{N} w_k x_k}_{\text{logistic regression}} + \underbrace{\sum_{1 \le k < l \le N} x_k x_l \langle \boldsymbol{v_k}, \boldsymbol{v_l} \rangle}_{\text{pairwise interactions}}$$

## Particular cases

- Multidimensional item response theory: $\operatorname{logit} p = \langle \boldsymbol{u_i}, \boldsymbol{v_j} \rangle + e_j$
- SPARFA: $\boldsymbol{v_j} > \boldsymbol{0}$ and $\boldsymbol{v_j}$ sparse
- GenMA: $\boldsymbol{v_j}$ is constrained by the zeroes of a q-matrix $(q_{ij})_{i,j}$

Andrew S Lan, Andrew E Waters, Christoph Studer, and Richard G Baraniuk (2014). "Sparse factor analysis for learning and content analytics". In: *The Journal of Machine Learning Research* 15.1, pp. 1959–2008

Jill-Jênn Vie, Fabrice Popineau, Yolaine Bourda, and Éric Bruillard (2016). "Adaptive Testing Using a General Diagnostic Model". In: *European Conference on Technology Enhanced Learning*. Springer, pp. 331–339

## Assistments 2009 dataset

278608 attempts of 4163 students over 196457 items on 124 skills.

- Download http://jiji.cat/weasel2018/data.csv
- Put it in data/assistments09

python fm.py data/assistments09/X-ui.npz
etc. or make big

| AUC | users + items | skills + w + f | items + skills + w + f |
|-----|---------------|----------------|------------------------|
| LR | 0.734 (IRT) 2s | 0.651 (PFA) 9s | 0.737 23s |
| FM | 0.730 2min9s | 0.652 43s | 0.739 2min30s |

Results obtained with FM $d = 20$

# Benchmarks

| Modèle | Composant | Taille | AUC |
|---|---|---|---|
| Bayesian Knowledge Tracing (Corbett and Anderson 1994) | Modèle de Markov caché | $2N$ | 0.63 |
| Deep Knowledge Tracing (Piech et al. 2015) | Réseau de neurones récurrent | $O(Nd + d^2)$ | 0.75 |
| Item Response Theory (Rasch 1960) (Wilson et al. 2016) | Régression logistique online | $N$ | 0.76 |
| Knowledge Tracing Machines | Factorization Machines | $Nd + N$ | 0.82 |

**AAAI 2019** Jill-Jênn Vie and Hisashi Kashima (2019) "Knowledge Tracing Machines: Factorization Machines for Knowledge Tracing". Proceedings of the 33th AAAI Conference on Artificial Intelligence.

# Impact on learning: modeling forgetting

Optimize scheduling of items in spaced repetition systems ($\sim$ Anki)



Use knowledge tracing machines with extra features: counters of attempts at skill level for different time windows in the past

**EDM 2019** Benoît Choffin, Fabrice Popineau, Yolaine Bourda, and Jill-Jênn Vie (2019) "DAS3H: Modeling Student Learning and Forgetting for Optimally Scheduling Distributed Practice of Skills". Best Paper Nominee..

## Deep Factorization Machines

Learn layers $W^{(\ell)}$ and $b^{(\ell)}$ such that:

$$a^0(\boldsymbol{x}) = (\boldsymbol{v}_{\texttt{user}}, \boldsymbol{v}_{\texttt{item}}, \boldsymbol{v}_{\texttt{skill}}, \ldots)$$

$$a^{(\ell+1)}(\boldsymbol{x}) = \text{ReLU}(W^{(\ell)}a^{(\ell)}(\boldsymbol{x}) + b^{(\ell)}) \quad \ell = 0, \ldots, L-1$$

$$y_{DNN}(\boldsymbol{x}) = \text{ReLU}(W^{(L)}a^{(L)}(\boldsymbol{x}) + b^{(L)})$$

$$\text{logit } p(\boldsymbol{x}) = y_{FM}(\boldsymbol{x}) + y_{DNN}(\boldsymbol{x})$$

Jill-Jênn Vie (2018). "Deep Factorization Machines for Knowledge Tracing". In: *The 13th Workshop on Innovative Use of NLP for Building Educational Applications*. URL: https://arxiv.org/abs/1805.00356
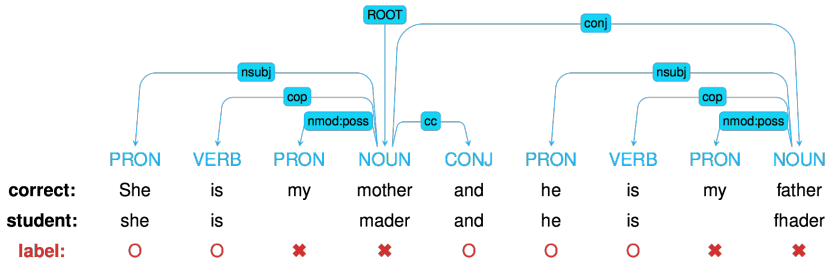
## Comparison

- FM: $y_{FM}$ factorization machine with $\lambda = 0.01$
- Deep: $y_{DNN}$: multilayer perceptron
- DeepFM: $y_{DNN} + y_{FM}$ with shared embedding
- Bayesian FM: $w_k, v_{kf} \sim \mathcal{N}(\mu_f, 1/\lambda_f)$
  $\mu_f \sim \mathcal{N}(0, 1)$, $\lambda_f \sim \Gamma(1, 1)$ (trained using Gibbs sampling)

### Various types of side information

- first: `<discrete>` (user, token, countries, etc.)
- last: `<discrete>` + `<continuous>` (time + days)
- pfa: `<discrete>` + wins + fails

Introduction
○○○○○○○○○

Logistic Regression
○○○○○○○○

Factorization Machines
○○○○○○○○○○○

**Deep Learning**
○○●○○○○○○○○○

Conclusion
○○○○

# Duolingo dataset



```
# user:D2inSf5+  countries:MX  days:1.793  client:web  session:lesson  format:reverse_translate  time:16
8rgJEAPw1001  She      PRON    Case=Nom|Gender=Fem|Number=Sing|Person=3|PronType=Prs|fPOS=PRON++PRP    nsubj      4  0
8rgJEAPw1002  is       VERB    Mood=Ind|Number=Sing|Person=3|Tense=Pres|VerbForm=Fin|fPOS=VERB++VBZ    cop        4  0
8rgJEAPw1003  my       PRON    Number=Sing|Person=1|Poss=Yes|PronType=Prs|fPOS=PRON++PRP$              nmod:poss  4  1
8rgJEAPw1004  mother   NOUN    Degree=Pos|fPOS=ADJ++JJ                                                 ROOT       0  1
8rgJEAPw1005  and      CONJ    fPOS=CONJ++CC                                                           cc         4  0
8rgJEAPw1006  he       PRON    Case=Nom|Gender=Masc|Number=Sing|Person=3|PronType=Prs|fPOS=PRON++PRP   nsubj      9  0
8rgJEAPw1007  is       VERB    Mood=Ind|Number=Sing|Person=3|Tense=Pres|VerbForm=Fin|fPOS=VERB++VBZ    cop        9  0
8rgJEAPw1008  my       PRON    Number=Sing|Person=1|Poss=Yes|PronType=Prs|fPOS=PRON++PRP$              nmod:poss  9  1
8rgJEAPw1009  father   NOUN    Number=Sing|fPOS=NOUN++NN                                               conj       4  1

# user:D2inSf5+  countries:MX  days:2.689  client:web  session:practice  format:reverse_translate  time:6
oMGsnnH/0101  When     ADV     PronType=Int|fPOS=ADV++WRB                                              advmod     4  1
oMGsnnH/0102  can      AUX     VerbForm=Fin|fPOS=AUX++MD                                               aux        4  0
oMGsnnH/0103  I        PRON    Case=Nom|Number=Sing|Person=1|PronType=Prs|fPOS=PRON++PRP               nsubj      4  1
oMGsnnH/0104  help     VERB    VerbForm=Inf|fPOS=VERB++VB                                              ROOT       0  0
```

## Results

| Model | $d$ | epoch | train | first | last | pfa |
|---|---|---|---|---|---|---|
| Bayesian FM | 20 | 500/500 | – | 0.822 | – | – |
| Bayesian FM | 20 | 500/500 | – | – | 0.817 | – |
| DeepFM | 20 | 15/1000 | 0.872 | 0.814 | – | – |
| Bayesian FM | 20 | 100/100 | – | – | 0.813 | – |
| FM | 20 | 20/1000 | 0.874 | 0.811 | – | – |
| Bayesian FM | 20 | 500/500 | – | – | – | 0.806 |
| FM | 20 | 21/1000 | 0.884 | – | – | 0.805 |
| FM | 20 | 37/1000 | 0.885 | – | 0.8 | – |
| DeepFM | 20 | 77/1000 | 0.89 | – | 0.792 | – |
| Deep | 20 | 7/1000 | 0.826 | 0.791 | – | – |
| Deep | 20 | 321/1000 | 0.826 | – | 0.79 | – |
| LR | 0 | 50/50 | – | – | – | 0.789 |
| LR | 0 | 50/50 | – | 0.783 | – | – |
| LR | 0 | 50/50 | – | – | 0.783 | – |

## Duolingo ranking

| Rank | Team | Algo | AUC |
|------|------|------|-----|
| 1 | SanaLabs | RNN + GBDT | .857 |
| 2 | singsound | RNN | .854 |
| 2 | NYU | GBDT | .854 |
| 4 | CECL | LR + L1 (13M feat.) | .843 |
| 5 | TMU | RNN | .839 |
| 7 (off) | JJV | Bayesian FM | .822 |
| 8 (off) | JJV | DeepFM | .814 |
| 10 | JJV | DeepFM | .809 |
| 15 | Duolingo | LR | .771 |

Burr Settles, Chris Brust, Erin Gustafson, Masato Hagiwara, and
Nitin Madnani (2018). "Second language acquisition modeling". In:
*Proceedings of the Thirteenth Workshop on Innovative Use of NLP for
Building Educational Applications*, pp. 56–65. URL:
http://sharedtask.duolingo.com

## What 'bout recurrent neural networks?

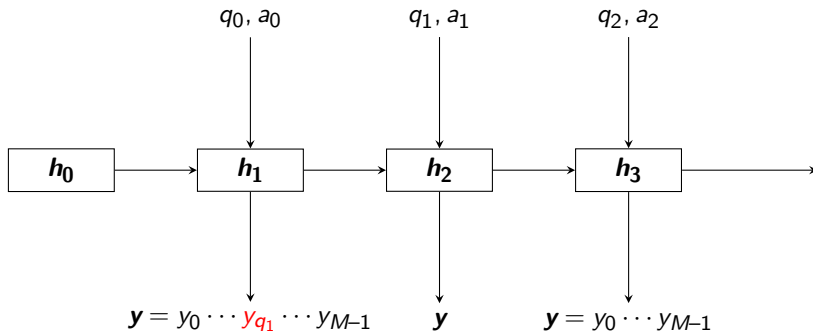Deep Knowledge Tracing: model the problem as sequence prediction

- Each student on skill $q_t$ has performance $a_t$
- How to predict outcomes $\boldsymbol{y}$ on every skill $k$?
- Spoiler: by measuring the evolution of a latent state $\boldsymbol{h_t}$

Chris Piech et al. (2015). "Deep knowledge tracing". In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 505–513

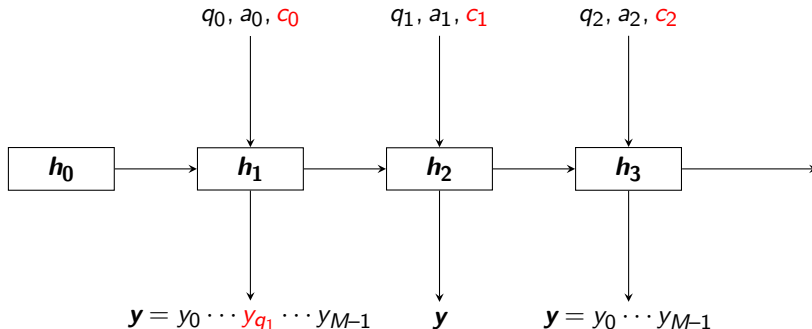### Our approach: encoder-decoder

$$\left\{ \begin{array}{l} \boldsymbol{h_t} = Encoder(\boldsymbol{h_{t-1}}, \boldsymbol{x_t^{in}}) \\ p_t = Decoder(\boldsymbol{h_t}, \boldsymbol{x_t^{out}}) \end{array} \right. \quad t = 1, \ldots, T$$

Introduction
OOOOOOOO

Logistic Regression
OOOOOOOO

Factorization Machines
OOOOOOOOOOO

**Deep Learning**
OOOOOO●OOOO

Conclusion
OOOO

## Graphically: deep knowledge tracing

# Deep knowledge tracing with dynamic student classification



**ICDM 2018** Sein Minn, Yi Yu, Michel Desmarais, Feida Zhu, and Jill-Jênn Vie (2018) "Deep Knowledge Tracing and Dynamic Student Classification for Knowledge Tracing". Proceedings of the 18th IEEE International Conference on Data Mining.

## DKT seen as encoder-decoder



$$y_{q_1} = \sigma(\langle \boldsymbol{h_1}, \boldsymbol{v_{q_1}} \rangle)$$
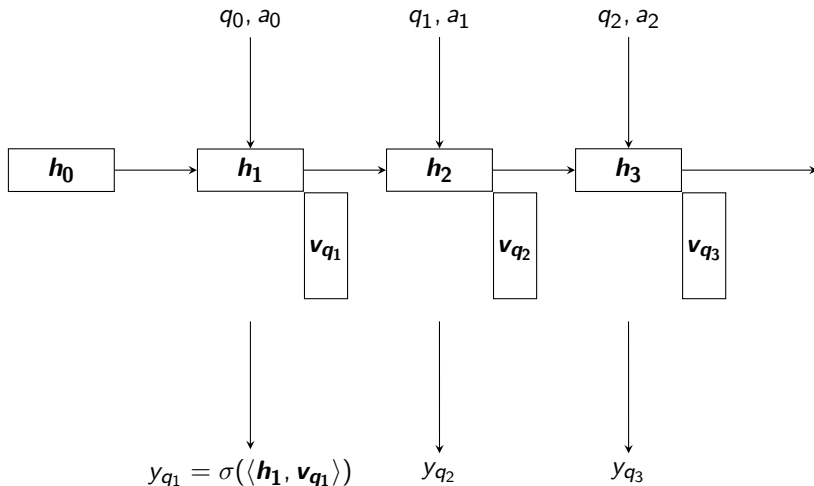
## Results on Fraction dataset

500 middle-school students, 20 Fraction subtraction questions, 8 skills (full matrix)

| Model | Encoder | Decoder | $x_t^{out}$ | ACC | AUC |
| --- | --- | --- | --- | --- | --- |
| **Ours** | GRU $d = 2$ | bias | iswf | **0.880** | **0.944** |
| KTM | counter | bias | iswf | 0.853 | 0.918 |
| PFA | counter | bias | swf | 0.854 | 0.917 |
| Ours | $\varnothing$ | bias | iswf | 0.849 | 0.917 |
| Ours | GRU $d = 50$ | $\varnothing$ | | 0.814 | 0.880 |
| DKT | GRU $d = 2$ | $d = 2$ | s | 0.772 | 0.844 |
| Ours | GRU $d = 2$ | $\varnothing$ | | 0.751 | 0.800 |

## Results on Berkeley dataset

562201 attempts of 1730 students over 234 CS-related items of 29 categories.

| Model | Encoder | Decoder | $x_t^{out}$ | ACC | AUC |
|---|---|---|---|---|---|
| **Ours** | GRU $d = 50$ | bias | iswf | **0.707** | **0.778** |
| **KTM** | counter | bias | iswf | **0.704** | **0.775** |
| Ours | $\varnothing$ | bias | iswf | 0.700 | 0.770 |
| DKT | GRU $d = 50$ | $d = 50$ | s | 0.684 | 0.751 |
| Ours | GRU $d = 100$ | $\varnothing$ | | 0.682 | 0.750 |
| PFA | counter | bias | swf | 0.630 | 0.683 |
| DKT | GRU $d = 2$ | $d = 2$ | s | 0.637 | 0.656 |

Jill-Jênn Vie and Hisashi Kashima (n.d.). "Encode & Decode: Generalizing Deep Knowledge Tracing and Multidimensional Item Response Theory". under review. URL: http://jiji.cat/bigdata/edm2019_submission.pdf

# Take home message

Factorization machines are a strong baseline for knowledge tracing that take many models as special cases

Recurrent neural networks are powerful because they track the evolution of the latent state (try simpler dynamic models?)

Deep factorization machines may require more data/tuning, but neural collaborative filtering offer promising directions

Next step: use this model and optimize human learning

## Any suggestions are welcome!

Feel free to chat:

vie@jill-jenn.net

All code:

github.com/jilljenn/ktm

Do you have any questions?

📄 Lan, Andrew S, Andrew E Waters, Christoph Studer, and
Richard G Baraniuk (2014). "Sparse factor analysis for learning
and content analytics". In: *The Journal of Machine Learning
Research* 15.1, pp. 1959–2008.

📄 Piech, Chris, Jonathan Bassen, Jonathan Huang, Surya Ganguli,
Mehran Sahami, Leonidas J Guibas, and Jascha Sohl-Dickstein
(2015). "Deep knowledge tracing". In: *Advances in Neural
Information Processing Systems (NIPS)*, pp. 505–513.

📄 Rendle, Steffen (2012). "Factorization Machines with libFM". In:
*ACM Transactions on Intelligent Systems and Technology (TIST)*
3.3, 57:1–57:22. DOI: 10.1145/2168752.2168771.

📄 Settles, Burr, Chris Brust, Erin Gustafson, Masato Hagiwara, and
Nitin Madnani (2018). "Second language acquisition modeling".
In: *Proceedings of the Thirteenth Workshop on Innovative Use of
NLP for Building Educational Applications*, pp. 56–65. URL:
http://sharedtask.duolingo.com.

📄 Vie, Jill-Jênn (2018). "Deep Factorization Machines for Knowledge Tracing". In: *The 13th Workshop on Innovative Use of NLP for Building Educational Applications*. URL: https://arxiv.org/abs/1805.00356.

📄 Vie, Jill-Jênn and Hisashi Kashima (n.d.). "Encode & Decode: Generalizing Deep Knowledge Tracing and Multidimensional Item Response Theory". under review. URL: http://jiji.cat/bigdata/edm2019_submission.pdf.

📄 Vie, Jill-Jênn, Fabrice Popineau, Yolaine Bourda, and Éric Bruillard (2016). "Adaptive Testing Using a General Diagnostic Model". In: *European Conference on Technology Enhanced Learning*. Springer, pp. 331–339.